OPTIMIZATION METHODS AND ALGORITHMS FOR CLASSES OF BLACK-BOX AND GREY-BOX PROBLEMS

A Dissertation

by

ISHAN BAJAJ

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	M. M. Faruque Hasan
Committee Members,	Mahmoud El-Halwagi
	Erick Moreno-Centeno
	Joseph Sang-Il Kwon
Head of Department,	M. Nazmul Karim

May 2019

Major Subject: Chemical Engineering

ABSTRACT

There are many optimization problems in physics, chemistry, finance, computer science, engineering and operations research for which the analytical expressions of the objective and/or the constraints are unavailable. These are black-box problems where the derivative information are often not available or too expensive to approximate numerically. When the derivative information is absent, it becomes challenging to optimize and guarantee optimality of the solution. The objective of this Ph.D. work is to propose methods and algorithms to address some of the challenges of blackbox optimization (BBO). A top-down approach is taken by first addressing an easier class of black-box and then the difficulty and complexity of the problems is gradually increased.

In the first part of the dissertation, a class of grey-box problems is considered for which the closed form of the objective and/or constraints are unknown, but it is possible to obtain a global upper bound on the diagonal Hessian elements. This allows the construction of an edge-concave underestimator with vertex polyhedral solution. This lower bounding technique is implemented within a branch-and-bound framework with guaranteed convergence to ϵ-global optimality. The technique is applied for the optimization of problems with embedded system of ordinary differential equations (ODEs). Time dependent bounds on the state variables and the diagonal elements of the Hessian are computed by solving auxiliary set of ODEs that are derived using differential inequalities.

In the second part of the dissertation, general box-constrained black-box problems are addressed for which only simulations can be performed. A novel optimization method, UNIPOPT (Univariate Projection-based Optimization) based on projection onto a univariate space is proposed. A special function is identified in this space that also contains the global minima of the original function. Computational experiments suggest that UNIPOPT often have better space exploration features compared to other approaches.

The third part of the dissertation addresses general black-box problems with constraints of both known and unknown algebraic forms. An efficient two-phase algorithm based on trust-region framework is proposed for problems particularly involving high function evaluation cost. The performance of the approach is illustrated through computational experiments which evaluate its ability to reduce a merit function and find the optima.

DEDICATION

To my parents and wife.

ACKNOWLEDGMENTS

The past four and half years of my PhD have been one of the most exciting, inspiring and satisfying years of my life. This dissertation would not have been possible without the constant support and encouragement from my advisor, Professor Faruque Hasan. I owe him a great debt for spending a lot of his time and energy on the work in this dissertation and on my education and professional development. He has constantly motivated me to realize my full potential in research. Professor Hasan has always been open to new ideas and that enabled me to broaden my skills and expertise. He has always been able to find time to explain and discuss concepts whenever I needed, and for this, I am truly grateful. He has taught me to do research and present the ideas succinctly. Outside research, he has actively encouraged and supported me to take leadership positions even though these are not necessary to complete PhD, but are nonetheless essential in life. Professor Hasan's work ethics and energy are some of the qualities that I hope to ingrain in my own professional life. He has been truly an exceptional mentor with whom I enjoyed working very much and consider as a role model.

I am also grateful for the feedback I have received from the members of my dissertation committee - Professors Mahmoud El-Halwagi, Erick Moreno-Centeno, and Joseph Sang-Il Kwon leading to improved version of this dissertation. Special thanks to Professors El-Halwagi and Kwon to take out time from their schedule and offer career advice. I enjoyed the lectures of Professor Efstratios Pistikopoulos and late Professor Christodoulos Floudas covering several aspects of optimization methods and applications. My journey towards PhD would not have even started if I had not met my first mentor, Professor Mani Bhushan at IIT Bombay who influenced me to pursue this path.

I would like to thank the members of my research group: Shachit Iyer, Emre Demirel, Jianping Li, Priyadarshini Balasubramanian, Spyridon Tsolas, Akhil Arora and Manali Zantye for all the times in and outside the office and teaching me several things. Special thanks to Shachit Iyer, Priyadarshini Balasubramanian and Akhil Arora for the collaborations that led to a more productive

PhD. A special shout out to GERB/JEB tennis club members: Patrick Lathrop, Doga Demirhan, Chen Ling, Abhinav Narasingam, and Sourav Bandyopadhyay for making my stay in College Station memorable. Thanks to Prashanth Siddhamshetty for all the interesting discussions and his support. I am grateful to my roommate Dr. Avinash Parnandi for his kindness during the most difficult times. I am also indebted to members of IIT Alumni of Greater Houston, South Asian Connections, Artie McFerrin Department of Chemical Engineering, Dr. Narendra Kumar, Dr. Nirup Nagabandi, Dr. Sushmitha Kotu, Dr. Prerna Jain, Dr. Monjoo Lee, Dr. Sagar Lonkar, Dr. Pratik Darvekar, Dr. Nandita Kohli, Abhijeet Shinde, Purvali Chaudhari, Pankaj Goel, Smit Shah, Pratik Krishnan, Patrick Lathrop, Katy Mears and Steven White for their generosity. Thanks to my another roommate, Chiranjivi Botre for persuading me to adopt healthier lifestyle.

This work would not have been possible without the unconditional love and unwavering support of my parents - Devraj and Manju. They have ingrained the strong work ethics in me that allowed me to complete the work in this dissertation. I am endlessly grateful to my wife, Sakshi for her patience and sacrifices. She has been there for me in the most difficult of times and makes me a better and happier person. I thank my in-laws for their support and encouragement.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor M. M. Faruque Hasan, Professor Mahmoud El-Halwagi and Professor Joseph Sang-Il Kwon of the Artie McFerrin Department of Chemical Engineering and Professor Erick Moreno-Centeno of the Department of Industrial and Systems Engineering.

The process simulation model depicted in Chapter 5 was provided by Shachit S. Iyer of the Department of Chemical Engineering and was published in 2017 in an article listed in the references. All other work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by Faculty start-up grant from Artie McFerrin Department of Chemical Engineering and U.S. National Science Foundation grant (award number CBET-1606027).

NOMENCLATURE

BBO	Black-box Optimization
BPBC	Black-box Problems with Bound Constraints
GPBC	Grey-box Problems with Bound Constraints
BPUC	Black-box Problems with Unknown Constraints
BPHC	Black-box Problems with Hybrid Constraints
BPKC	Black-box Problems with Known Constraints
GPGC	Grey-box Problems with Grey-box Constraints
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
NAPDE	Nonlinear Algebraic Partial Differential Equation
B&B	Branch and Bound
ECU	Edge-concave Underestimator
UNIPOPT	Univariate Projection-based Optimization
EPIC	Envelope Predictor and Corrector

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION AND LITERATURE REVIEW[1]

Black-box optimization (BBO) refers to a class of problems in which the analytical form of the objective function and/or constraints are not available explicitly in terms of decision variables. There are many important problems in engineering, physics, finance, medicine, and operations research where the problem is posed as BBO. In many cases, the values of the objective function and constraints are only obtained as outputs of deterministic large-scale simulation, evaluation of legacy codes or experimentation. Moreover, the derivative information is either not available or too expensive to evaluate. Automatic differentiation fails when the objective function values are available only as a result of simulating legacy. Due to proprietary restrictions, the closed form of the objective function is unavailable. Applying finite difference to an expensive computer simulation can be prohibitive since it would require at least $n + 1$ evaluations, (n represents the problem dimension) to approximate the gradient at a point. To this end, BBO algorithms strive to find a solution in a computationally efficient manner.

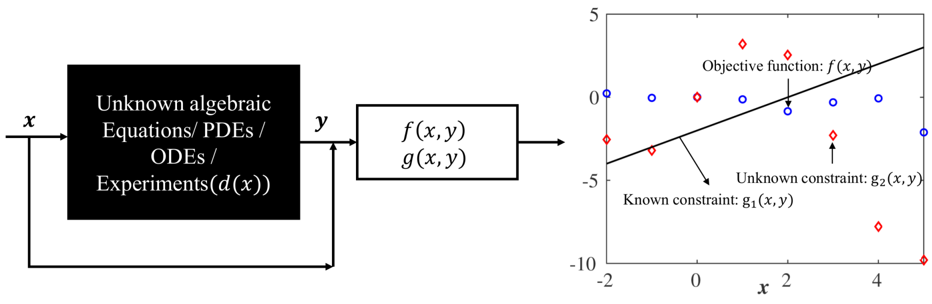
In the most general form, the BBO problems that are addressed in this dissertation can be represented as follows:

$$\begin{aligned} \min_{x} \quad & f(x,y) \\ \textit{s.t.} \quad & g_i(x,y) \leq 0 \quad \forall i \in \mathbb{P} := \{1,\ldots,p\} \\ & y = d(x) \\ & x \in [x^L, x^U] \end{aligned} \tag{1.1}$$

where $x \in \mathbb{R}^n$ are the independent variables, $y \in \mathbb{R}^q$ are the dependent variables, $f(x,y)$ and $g(x,y)$ denote the objective function and the constraints, respectively, and $d(x)$ represents a complex model whose mathematical form may be known or unknown. For instance, y could be ob-

[1]Part of this chapter is reprinted with permission fromI. Bajaj, S. S. Iyer, and M. M. F. Hasan, "A trust region-based two phase algorithm for constrained black-box and grey-box optimization with infeasible initial point" *Computers & Chemical Engineering*, vol. 116, pp. 306-321. Copyright 2018 Elsevier.

tained as a result of solving an ODE/PDE model, using a black-box simulator, or by performing experiments. When the underlying model $d(x)$ is complex and only input-output data are available, then the problem is referred as black-box. If it is possible to extract other information (e.g. derivatives, bounds on diagonal elements of the Hessian, etc.) besides the simulation data, the problem is considered as grey-box. If $\mathbb{P} = \emptyset$, then the respective problems are called black-box problems with bound constraints (BPBC) and grey-box problems with bound constraints (GPBC). Similarly, if $d(x)$ is black-box, $\mathbb{P} \neq \emptyset$ and *all* the constraints are functions of y, then the problems are referred as black-box problems with unknown constraints (BPUC). If $d(x)$ is black-box, $\mathbb{P} \neq \emptyset$ and *some* of the constraints are functions of y, then the problems are referred as black-box problems with hybrid constraints (BPHC). If $d(x)$ is black-box, $\mathbb{P} \neq \emptyset$ and *none* of the constraints are functions of y, then the problems are referred as black-box problems with known constraints (BPKC). Finally, if $d(x)$ is grey-box, the the problems are simply called grey-box problems with grey-box constraints (GPGC). BBO problem is illustrated in Figure 1.1.



x
Unknown algebraic Equations/ PDEs / ODEs / Experiments($d(x)$)
y
$f(x,y)$
$g(x,y)$
Objective function: $f(x,y)$
Unknown constraint: $g_2(x,y)$
Known constraint: $g_1(x,y)$
5
0
-5
-10
-2
0
2
4
x

Figure 1.1: Illustration of the BBO problem. The values of y are obtained by fixing x. Using the values x and y, the values of the black-box objective function (blue circles), the unknown constraints (red diamonds), and known constraints (black solid line) are computed.

In process systems engineering, BBO methods have been applied, among many applications, to natural gas liquefaction [2], heavy hydrocarbons removal [3], membrane reactor for hydrogen and methanol production [4], methanol reactor [5], co-production of ethylene and electricity via methane oxidative coupling [6], material screening and optimization [7, 8], multi-scale optimization [9], carbon capture process [10], pressure swing adsorption [11], oil-field operations [12] and refrigerant circuitry design for heat exchangers [13]. Palmer and Realff [14] developed polynomial and kriging models for a steady state ammonia synthesis flowsheet and subsequently optimized the surrogate model. Lima et. al. [15] applied stochastic algorithm for dynamic optimization of a batch polymerization reactor assuming the reactor model to be black-box. Neural networks have been used as estimator, predictor and controller for a batch reactor [16]. Egea et. al. [17] optimized the wastewater treatment plant using several black-box optimization solvers and compared their results. Caballero and Grossmann [18] developed a Kriging-based algorithm for flowsheet optimization. Boukouvala and Ierapetritou [19] employed Kriging model in combination with black-box feasibility for optimization of tablet manufacturing process. Yang et. al. [20] optimized the reaction selectivity based on CFD-based compartment model using MISO [21]. Dias et. al. [22] applied surrogate-based optimization method to integrate scheduling and control of air separation units. BBO methods have also been applied in the optimization of parameters of numerical code for simulation and optimization [23, 24], optimization of groundwater bioremediation [25], protein structure prediction [26], design of cardiovascular surgeries [27], aircraft routing [28] and dynamic pricing [29].

BBO methods are also growing in importance due to increased use of computationally expensive PDE models. These models account for spatial and temporal variation of the system properties, thereby enabling detailed optimization of design and operating conditions. Although PDE models have higher predictive ability, rigorous optimization with PDE models is challenging. One of the approaches investigated in the literature is to discretize the PDEs and convert into algebraic equations. The resulting optimization model then becomes a large-scale nonlinear programming problem. For instance, Nilchan and Pantelides [30] optimized a periodic adsorption process by applying a sequential quadratic programming algorithm to the discretized equations. Biegler and co-workers [31, 32] applied IPOPT [33] to optimize the discretized model of a adsorption based process. Although promising results were obtained, the level of discretization used in these works were moderate to keep the nonlinear model tractable and by doing so, a certain degree of accuracy was lost. An alternative and promising strategy is to use black-box optimization concepts that involves using simulation data to optimize while maintaining the high level of accuracy of the model. This approach has been successfully used to optimize an integrated carbon capture and conversion process [1] and sorption enhanced reaction processes [34, 35].

Clearly, black-box optimization has numerous practical applications and is one of the open problems in science and engineering [36]. Effective black-box optimization methods are needed in Chemical Engineering now more than ever. This is due to the growing importance of high-fidelity computationally expensive models being used in Chemical Engineering to model several processes such as reactor, furnace, adsorption, etc. From the study performed by Iyer et. al. [1], it is concluded that integrating and intensifying different physiochemical process introduces complexities which are not trivial to understand. Therefore, obtaining the appropriate process parameters that leads to an economical feasible process is a challenging problem. To balance different trade-offs, a systematic optimization approach is needed for which the current solvers are not very effective. To this end, the primary motivation of this Ph.D. study is to develop optimization methods and algorithms for black-box problems where the interactions are complex with multiple variables affecting the process output.

1.1 Theoretical and Algorithmic Advances in Black-box Optimization

The growing number of applications has led to development of several algorithms for solving black-box problems. Intuitively, one can think of two approaches to optimize black-box problems. The first one is a sequential approach where all the simulations are performed at once, followed by developing an accurate surrogate model for the entire domain and thereafter, optimizing the surrogate model. The second approach is iterative with attempts being made to simulate in the promising regions where there is a possibility of finding an optima. The first approach is not adequate when the simulations are expensive and the problem is high dimensional. Therefore, the main interest is in following the second approach involving sampling in a smart manner such that a minima is obtained using few function evaluations. These algorithms can be classified into two major categories, namely direct search and model-based [37]. Direct search algorithms use the function values directly by generating search directions that positively span the search space and then sequentially examining the function values to determine the next step. The search directions can be local [38, 39] or global [40, 41, 42, 43, 44, 45]. Local search techniques generate a set of points to form a simplex or pattern by considering only local search space. Global search is categorized into deterministic and stochastic techniques. One of the stochastic algorithms is the hit-and-run algorithm that generates new candidate points from the current points by randomly generating directions and step sizes [43]. The current point is updated only if an improvement in the objective function is obtained. Simulated annealing [46] generates and accepts a new candidate point as the current point if an increase in the objective function is obtained. Genetic algorithms are based on the idea of natural selection that assures the survival of the fittest [47]. Each point is associated with fitness measure that mutates following probabilistic rules. Particle swarm algorithms [48] are based on generating new set of points from previous points based on certain parameters and randomly generated weights.

Several deterministic approaches also exist for global black-box optimization. The first approach is based on the Lipschitzian-based partitioning scheme that constructs and optimizes a linear underestimator by using Lipschitz constant (e.g., [49]). In this approach, the value of the Lipschitzian constant must be known. Another observation is that the required number of function evaluations typically increases exponentially with the problem dimension. Jones et. al. [40] addressed this issue by dividing the search space into hyperrectangles and performing function evaluations only at the rectangle centers. The method proceeds by iteratively dividing the rectangle that either corresponds to the lowest objective function value or likely to produce substantial reduction in the objective value. In the absence of the Lipschitz constant, the algorithm converges when the function evaluation limit is reached. In contrast, the multilevel coordinate search (MCS) method [41] performs function evaluation at any point within the rectangle, thus avoiding lot of evaluations. Branch-and-bound methods have been also proposed based on estimated upper bound of the Lipschitz constant [50]. In this case, the rate of convergence may depend on the tightness of the estimation. However, the global search methods often require a lot of function evaluations and do not provide theoretical guarantee of convergence to the global minima. Direct methods are often easier to implement and parallelize compared to model-based methods [36].

Model-based methods implicitly approximate derivatives to determine a search direction. Typically, the model-based methods sample the search space, develop and optimize inexpensive surrogate models and then iteratively update the models. Model-based algorithms are also categorized into local [51, 52, 53, 54] and global [55, 56, 57]. Local model-based methods rely on function evaluations and building a fully-linear or fully-quadratic interpolation or regression models in a trust-region. The new point is obtained by solving the trust-region subproblem. The trust-region size is increased, decreased or kept constant and the center is shifted as the iteration progresses. Several algorithms based on trust-region has been developed in the literature [54]. Powell [53] developed BOBYQA based on constructing quadratic models of the objective function and utilized two trust-reigons. Gilmore and Kelley [51] proposed IMFIL based on fully-quadratic approximation of the gradient while Wild et. al. [52] employed radial basis function to approximate the original function. Oeuvray and Bierlaire [58] proposed BOOSTERS that used radial basis function for medical image registration problem. One of the global model-based approach is based on developing global surrogate models. In this case, the algorithm typically performs space-filling sampling, estimates parameters of a global surrogate model, and obtains candidate solutions via surrogate-based optimization. Some of the commonly used surrogate models are kriging [59], radial basis function [60], quadratic response surface, and polynomial approximation [61]. Due to the inaccuracy of the surrogate models, the results obtained from the surrogate-based optimization may not always correspond to the actual solution. Therefore, auxiliary functions such as expected improvement [55] are used that enable exploring the unexplored space. In some cases, the feasible region is divided into smaller subregions with separate surrogate models capturing the local characteristics of the problem [57]. Quadratic models are used in the regions that are more likely to have global minima, while linear models are employed at other evaluated points. Lately, the idea of hybrid approaches is also being explored. For instance, SID-PSM [62] combines pattern search method with quadratic models to improve the search step, while using simplex gradients for the poll step.

Some of the these approaches have been extended to solve constrained black-box problems by incorporating several constraint handling strategies. One of the approaches to handle constraints is to transform the original constrained problem into an unconstrained problem using exact penalty method [63]. A line-search method is applied to optimize the resulting problem. However, the penalty parameter needs to be chosen carefully such that it is smaller than a certain threshold that is not known *a priori*. Liuzzi et. al. [64] defined a merit function and applied a sequential penalty approach to handle constraints. The penalty parameter update rule was connected to the sampling technique to ensure convergence. An alternative approach proposed to handle constraints is by employing augmented lagrangian approach [65]. This method formulates a shifted penalty function that includes black-box constraints in the objective function. This results in a black-box bound constrained problem for which both model-based and direct-search approaches were applied. Other methods have been proposed in the literature that handle constraints directly. Mesh adaptive direct-search (MADS) is a direct-search method that treats constraints using an extreme barrier [66] or a progressive barrier approach [67]. The extreme barrier based method rejects all infeasible points while progressive barrier method balances the trade-offs between constraint violation and objective function. The threshold on constraint violation is reduced iteratively such that the iterates move towards feasible region. Filter methods have been applied along with direct-search methods to solve constrained black-box problems [68].

COBYLA [69] is one of the first model-based approaches for solving constrained black-box problems. In COBYLA, linear models for the objective function and constraints are developed using interpolation points on the vertices of a simplex and the optimization is performed in a trust-region framework. Augustin and Marzouk [70] developed NOWPAC based on a trust-region framework. An offset function is added to the constraints to convexify the local feasible region and facilitate the generation of feasible trial points. However, NOWPAC requires the initial point to be feasible. Arouxét et. al. [71] applied an inexact restoration method with polynomial models and used merit function to measure the progress of the algorithm. Audet et. al. [72] employed quadratic polynomials to approximate the objective function and constraints and progressive barrier approach is applied to handle constraints. Sampaio and Toint [73] extended the trust-funnel method [74] to constrained black-box problems. Each iteration in trust-funnel algorithm is composed of two steps. The first step reduces the infeasibility and the second step aims to improve the optimality. The bound on allowed infeasibility is decreased monotonically with iterations so that infeasible points are also acceptable at certain iterations but the final point is feasible. Radial basis functions are used as surrogate models to approximate the objective function and constraints in CONORBIT [75]. Eason and Biegler [76, 77] proposed trust-region based filter algorithm for black box and glass-box models and applied it to optimize a carbon capture case study. Boukouvala and Floudas [78] proposed ARGONAUT based on constrained sampling and bound tightening techniques. Wang and Ierapetritou [79] developed a radial basis function based methodology to approximate the feasible region of black-box systems. Kieslich et. al. [80] used Smolyak grid to perform global sampling and constructed polynomial models to guide search towards the optimal point.

The model-based methods do not attempt to construct accurate surrogate models throughout the domain. However, applications such as superstructure-based process synthesis (see e.g. [81, 82]) warrants development of globally accurate surrogate models. Cozad et. al. [83] developed ALAMO based on adaptive sampling to construct simpler model. Garud et. al. [84] proposed LEAPS2 to enable selection of the best surrogate model using the idea of knowledge pyramid. Nuchitprasittichai and Cremaschi [85] developed an algorithm to select the optimal sample size to construct artificial neural network model and optimized a process synthesis problem to validate their methodology. Straus and Skogestad [86] used self-optimizing variables theory to efficiently develop surrogate models of low complexity. Tran and Georgakis [87] used a net-elastic regularization method to estimate the parameters of a surrogate model such that overfitting was avoided and insignificant terms were eliminated.

Many algorithms mentioned above are currently available as software packages such as NOMAD [88], ORBIT [52], SNOBFIT [57], BOBYQA [53], DFO [89], NEWUOA [90], PSWARM [91], CMA-ES [44], ASA [92], DAKOTA [93], GLOBAL [94], IMFIL [51], MCS [41], DIRDFN

[95], DFSA [96], and SDPEN [64]. While PSWARM, CMA-ES, ASA, DAKOTA, GLOBAL, MCS, DIRDFN, DFSA, and SDPEN belong to the family of direct search methods, SNOBFIT, BOBYQA, DFO, NEWUOA, and IMFIL are model-based methods. NOMAD is based on a hybrid approach that uses quadratic model in its search step. Rios and Sahinidis [37] have extensively compared the performance of 27 solvers on a set of 502 unconstrained or box-constrained problems.

Two excellent books [36, 97] and several review papers (see e.g. [37, 98, 99, 100]) have been published that summarizes the developments in the field of black-box/derivative-free optimization.

1.2 Key Research Gaps and Challenges

In a recently published paper, Iyer et. al. [1] proposed an intensified carbon capture and conversion process to convert flue gas to syngas. The system was modeled using nonlinear algebraic partial differential equations (NAPDE) to accurately depict the underlying physio-chemical phenomenon. The aim of the process was to economically produce syngas with certain quality while satisfying greenhouse gas emission constraints. The resulting optimization problem was a black-box problem that involved objective function and constraints for which the closed form expressions in terms of the decision variables were unavailable. However, their values were obtained by fixing the decision variables and solving the NAPDE model. When simulations were performed at a large number of random points, even obtaining a feasible point is challenging as illustrated in Figure 1.2. The optimal design and operating conditions were sought that could make the process economically viable and also satisfy the quality and regulatory constraints. Therefore, it is critical to obtain a high quality solution. Secondly, since the model is computationally expensive, it is desired that an optimal solution is obtained within few simulations/function evaluations. Due to complex interactions of the decision variables, relying on heuristics may not be sufficient to study the techno-economic feasibility of a process and systematic optimization approach is essential.

In spite of numerous theoretical and algorithmic advancements in black-box optimization, there are several limitations of the current solvers that need to be addressed. First, in many cases when the model is too complex, it is considered black-box. However, it is actually grey-box and extract-

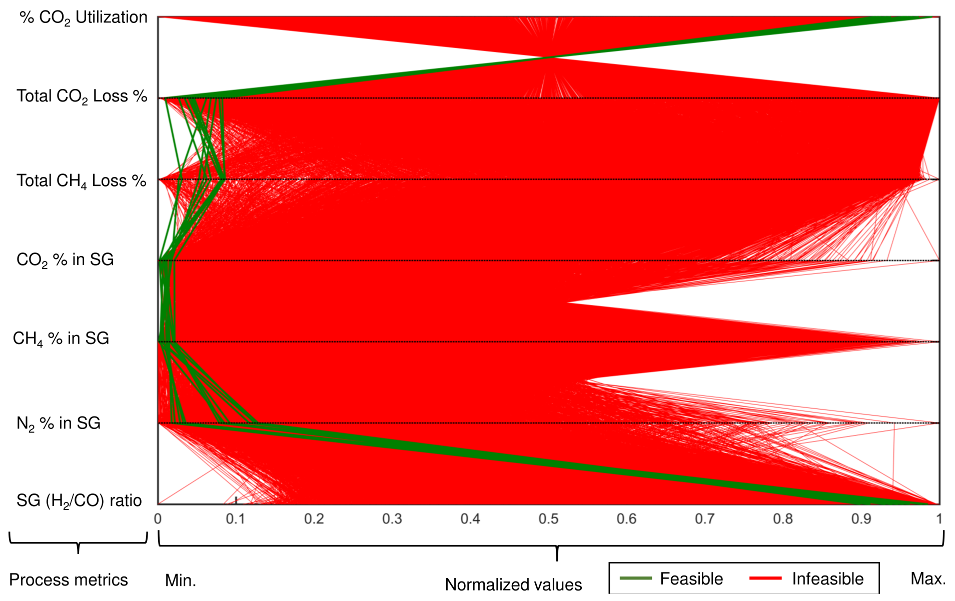


Figure 1.2: Simulation at 10000 points of the process proposed in [1] are performed and adapted from [1]. Each simulation is represented by a line joining different outputs and the decision variables. The red line represents when a simulation is infeasible with respect to the optimization problem while green represents a feasible design points.

ing some information to exploit the model may allow solving a problem to guaranteed global optimality. Therefore, the challenge is to identify a class of grey-box problems for which an ϵ-global optimal solution can be computed. Second, the optimization problems with embedded systems of ODEs are identified as a class of grey-box problems for which the global minima can be guaranteed. Since the analytical form of the objective and constraints in terms of decision variables are unknown, it is challenging to find tight underestimator. Third, the scale of the problems that can be solved by black-box optimization solvers are relatively small. Fourth, finding global solutions for a general non-convex black-box problem is challenging. When the problem is completely black-box, direct methods such as PSWARM, MCS, CMA-ES are more successful in achieving the global minima [37] because they explore the search space by either partitioning the space or randomly sampling the unexplored space. However, they do not even provide theoretical guarantee of convergence to a local minima. On the other hand, except SNOBFIT, model based methods often converge to local minima owing to their inability to explore the search space. Fifth, many of the current solvers for constrained black-box problems assume availability of an initial feasible point which is unreasonable for many practical applications. It is also assumed in many cases that all the constraints are black-box and relaxable. By doing so, many critical model insights are neglected. It is possible in many cases that the simulation would fail if the samples do not satisfy certain constraints. The challenge is to have a convergent method that finds optimal solution even if an initial feasible point is not provided and handles hard constraints.

The key challenges can be summarized as follows:

- Global optimization of black-box problems with only available information on the upper bounds of the diagonal Hessian elements.
- Efficient computation of the diagonal elements of the Hessian for global black-box optimization.
- Projection-based approaches as potential alternatives to optimize black-box problems.
- Efficient algorithms for black-box problems that explore space efficiently to improve the chances of finding global minima and are guaranteed to converge to atleast a local minima.
- Efficient black-box optimization in the presence of constraints and sampling methods that ensure that the hard constraints are always satisfied and the simulations do not fail.

1.3 Research Objectives

In this Ph.D. work, novel methods to solve constrained black-box and grey-box problems are proposed. A focus has been on global optimization, whenever possible. The proposed methods have the been applied to solve both literature problems and practical engineering problems. The primary objective of this work are:

1. Develop a guaranteed underestimator for the global optimization of problems of problems where only the upper bounds on the diagonal Hessian elements are known.
2. Develop a deterministic algorithm that incorporates the underestimator and efficiently solve the grey-box problems to ϵ-global optimality.

3. Develop a projection-based algorithm based on identifying and optimizing a univariate function for box-constrained black-box problems.

4. Develop an algorithm that have the ability to handle black-box problems with hard/unrelaxable constraints, and

5. Provide rigorous proofs and identify conditions of convergence of the proposed algorithms

1.4 Outline of the Dissertation

This dissertation includes six chapters. Optimization of grey-box problems with bound constraints (GPBC) and grey-box problems with grey-box constraints (GPGC) problems are addressed in Chapter 2 and Chapter 3, respectively. A projection based methodology for solving black-box problems with bound constraints (BPBC) is given in Chapter 4. The algorithm given in Chapter 5 is capable of solving BPBC, black-box problems with hybrid constraints (BPHC) and and black-box problems with unknown constraints (BPUC).

In Chapter 2, a class of grey-box problems is defined for which the maximum of the upper bounds of the diagonal elements of the Hessian are available. Using this information, a novel underestimator is constructed and incorporated within a branch-and-bound framework. The proposed methodology is also compared to other available solvers on several nonconvex problems and results indicate that the proposed method is computationally advantageous.

It is inferred that optimization problems with integral terms in the objective function and constraints with embedded system of nonlinear ODEs are a special case of grey-box problems defined in Chapter 2. In Chapter 3, the details of the deterministic global optimization method to compute ϵ-global optimal solution of dynamic optimization problems are provided. The effectiveness of the method is illustrated through several case studies.

In Chapter 4, bound constrained black-box problems are addressed. An algorithm based on univariate projection is developed. The problem is completely black-box in the sense that only simulation data are needed. The algorithm is shown to converge to the local minima of the original black-box problem when certain assumptions are satisfied. The algorithm is thoroughly compared to existing model-based approaches using an extensive set of test problems. The proposed methodology offers an advantage in terms of finding the global minima for more number of problems compared to other approaches.

In Chapter 5, a trust-region based two-phase algorithm is developed for optimization of constrained black-box problems. This algorithm is also guaranteed to converge to the local minima of the original problem. The two-phase methodology is compared to other solvers on a extensive set of problems from standard test libraries. It is shown that this algorithm is computationally advantageous to other solvers.

Finally, several conclusions and recommendations for future research are provided in Chapter 6.

2. GLOBAL OPTIMIZATION OF GREY-BOX PROBLEMS WITH BOUNDED HESSIAN

2.1 Introduction

A special form of the original problem in Eq. (1.1) is considered and given by:

$$\min_{x \in S} \quad f(x) \tag{P1}$$

where, $x \in S = \{x^L \leq x \leq x^U\} \subseteq \mathbb{R}^n$, and $f(x) : S \mapsto \mathbb{R}$ is a twice differentiable grey-box function with unknown algebraic form. However, the values of f can be obtained by performing deterministic simulation, evaluation of legacy codes, or experiments for fixed x. Furthermore, a global upper bound of the diagonal Hessian elements of f i.e., $\max_i \left[\frac{\partial^2 f}{\partial x_i^2}\right]^U$ is known.

It is assumed that computing derivatives at every point is not possible or computationally expensive but a global upper bound on the elements of the Hessian can be quickly estimated based on physical interpretation and model analysis or even intuition. For instance, consider the following optimization problem:

$$\begin{aligned} \min_{k_1, k_2} \quad & y_1(t_f) \\ \textit{s.t.} \quad & \dot{y}_1 = k_1 y_1^2 - k_2 y_2, \quad \forall t \in (t_0, t_f] \\ & y_1(t_0) = y_0 \\ & k_1, k_2 \in [0, 1] \end{aligned} \tag{2.1}$$

where, the time variation of y_2 could be given by computationally expensive proprietary code. However, assume that it is known through physical insights that y_2 is independent of y_1 and the decision variables - k_1 and k_2. In the above problem, the objective function is grey-box and computing derivatives will be computationally expensive. However, it is possible to compute the upper bound of the elements of the Hessian.

Furthermore, consider another example involving estimating kinetic parameters of the following reaction:

$$A \xrightarrow{k} B$$

Typically, the parameter estimation problem involves solving an optimization problem such that the error between the experimental data and that predicted by model is minimized. Consider the following optimization problem:

$$\begin{aligned} \min_{k} \quad & \sum_{i=1}^{p} (y(t_i) - y^{exp}(t_i))^2 \\ s.t. \quad & \dot{y} = -\frac{k\alpha y}{1+y}, \quad \forall t \in (0,1] \\ & y(t_0) = y_0 \\ & k \in [0,10] \end{aligned} \tag{2.2}$$

where, y^{exp} denote the composition of A measures at different time instants, α is the activity coefficient. Due to proprietary reasons, the catalyst manufacturer may not provide the equations governing the deactivation of catalyst but may instead provide a blackbox simulator or the equations are given by computationally expensive set of PDEs [101] or it is determined using experiments. However, since it is known through system knowledge that $\alpha \in [0,1]$, it is possible to obtain the bounds on the elements of the Hessian.

Except for the cases with known Lipschitz constants, most global methods require dense sampling to guarantee the optimality for black-box problems. In this chapter, the main interest is in optimizing a class of grey-box problems for which a valid lower bound can be generated via minimum information, such as a bound on the diagonal Hessian elements. This would allow to use a branch-and-bound [102] scheme in a deterministic fashion. At each node, a general underestimator would provide a lower bound, and a local optimizer would generate an upper bound. These lower and upper bounds would be iteratively updated until their difference converges to an ϵ tolerance.

In the absence of any algebraic functional forms for the objective function, it is not possible to either use termwise convexification [103] or construct factorable reformulations [104]. To this end, a general underestimator is desired such that (i) it does not require the explicit algebraic form of the original function, and (ii) its minima can be obtained using finite number of evaluations. Therefore, the problem P1 can be considered with no exploitable mathematical structure. To this end, the vertex polyhedral property of the recently proposed edge-concave underestimator (ECU) [105] of general $\mathcal{C}^2$-continuous functions is exploited that provides an avenue for the global optimization of a certain class of grey-box problems. The edge-concave underestimator has a special property that its minima lies at one of the corner points. Construction of an ECU requires that the upper bound of the diagonal elements of the Hessian of the original function is known. Once this information is known, valid lower bound can be obtained by simply evaluating the underestimator at the corner points.

In this chapter, a method for deterministic global optimization of a class of grey-box problems ir proposed. Valid lower bounds are obtained by constructing ECU by assuming that the upper bound on the diagonal elements of the Hessian of the original function is known. The method is implemented in a branch-and-bound framework. The efficiency of the algorithm is illustrated using a set of black-box problems.

The remaining of the chapter is organized as follows. In Section 2.2, the edge-concave underestimator based lower bounding of grey-box functions with known Hessian bounds is discussed. In Section 2.3, a branch-and-bound algorithm towards solving these classes of grey-box problems to ϵ-global optimality is presented. Computational results and conclusions are provided in Section 2.4 and 2.5, respectively.

2.2 Lower Bounding Method

Establishing valid lower bounds is central in any deterministic global optimization based on a branch-and-bound framework. The following result enables one to provide such a lower bound on $f(x)$ based on finite input-output simulation or sampling of $f(x)$. The mathematical form of the edge-concave underestimator is given as follows:

$$L(x) = f(x) - \sum_{i=1}^{n} \theta_i (x - x_i^m)^2 \tag{2.3}$$

where,

$$\begin{aligned}\theta_i &= \max\left\{0, \nabla^2_{ii} f\right\} \\ x_i^m &= \frac{x_i^l + x_i^u}{2}\end{aligned} \tag{2.4}$$

The parameter θ_i requires upper bounds of each of the diagonal elements of the Hessian. However, the result presented next enables computation of a valid lower bound by just knowing the maximum of θ_i.

Theorem 2.1. *Let $f(x)$: $\mathbb{R}^n \mapsto \mathbb{R}$ be a $\mathcal{C}^2$-continuous function. Then LB is a lower bound of $f(x)$, i.e., $LB \leq f(x)$ over S, if*

$$LB = \underset{x^v \in V}{min}\left[f(x^v) - \Theta \sum_{i=1}^{n} (x_i^v - x_i^m)^2\right] \tag{2.5}$$

where,

$$\Theta = \underset{i}{max}\left[max\left\{0, \frac{1}{2}\left[\nabla^2_{ii} f\right]^U\right\}\right], \tag{2.6}$$

where, V denotes the set of 2^n vetrices in S, and $[\nabla^2_{ii} f]^U$ denotes the known upper bound of the diagonal elements of the Hessian of f.

Proof. Hasan [105] showed that $L(x)$ in Eq. (2.3) is an underestimator of $f(x)$ over $x \in S$. By definition, $\Theta = ||\theta||_\infty \geq \theta_i$. Therfore, $LB \leq L(x^v)$ for any $x^v \in V$. Moreover, since $\nabla^2_{ii} L \leq 0$, $L(x)$ is an edge-concave function. Therefore, $L(x)$ admits a vertex polyhedral convex envelope (Theorem 3.2, Tardella [106]). This means that the convex envelope of an edge-concave function on a polyhedron S coincides with the convex envelope of its restrictions to the vertices of S. As a consequence, $\min_{x^v \in V} L(x^v) \leq L(x)$. This leads to $LB \leq \min_{x \in X_v} L(x) \leq L(x) \leq f(x)$. $\square$

REMARK 1. The fact that the minimum of $L(x)$ is located at one of the vertices $x^v \in V$ is very useful in the context of black-box optimization, since a valid lower bound can be obtained in finite number of evaluations. This suggests that edge-concave underestimators have the ability to relax a twice differentiable nonconvex function with no special structure. Furthermore, only Θ needs to be known to construct the global underestimator. As long as Θ is equal or greater than the value obtained from Eq. (2.6), the LB value obtained by Eq. (2.5) is a valid lower bound of $f(x)$.

Property 2.1. . *The maximum separation distance between $f(x)$ and LB is bounded and given by*

$$d_{sep}^{max} = \frac{\Theta}{4}\sum_{i=1}^{n}(x_i^u - x_i^l)^2 \tag{2.7}$$

2.2.1 Illustrative Example 2.1

To illustrate ECU, lets consider the following 1-dimensional function:

$$f(x) = 20 - 20exp(-0.02\left|\frac{x}{30}\right|) + e - exp\Big(\frac{cos(2\pi x)}{30}\Big), \quad x \in [1,3] \tag{2.8}$$

Here, the global minimum of $f(x)$ is 1.757. The exact upper bound of the Hessian of function listed in Eq. (2.8) is 1.36. The original function and its edge-concave underestimator (Eq. (2.3)) are shown in Figure 2.1 for different Θ values. Note that the underestimators, $L(x)$ are obtained with $\theta_i = \Theta$. Even if the actual form of $f(x)$ is unknown as given by Eq. (2.8) but only the value of Θ is given, then based on only two simulations of $f(x)$ at the vertices ($x = 1$, and $x = 3$), a valid lower bound is obtained. For instance, for $\Theta = 1.36$, lower bound is obtained as follows:

$$LB = \min\Big[\{f(1) - 1.36(1-2)^2\},\ \{f(3) - 1.36(3-2)^2\}\Big] \tag{2.9}$$

This results in the lower bound value to be 0.396. It is apparent from Figure 2.1 that for $\Theta > 1.36$, the underestimators constructed are valid but becomes loose with increasing Θ. The lower bound obtained for $\Theta = 2.5$, and $\Theta = 4$ are -0.742 and -2.24, respectively.

2.3 A Branch-and-Bound Algorithm

Based on *Property* 2.1 (Eq. (2.7)), the maximum separation distance between the black-box function and its edge-concave underestimator is proportional to $\sum_{i=1}^{n}(x_i^u - x_i^l)^2$. As the size of the domain becomes smaller, the maximum separation distance decreases. Therefore, a method can be

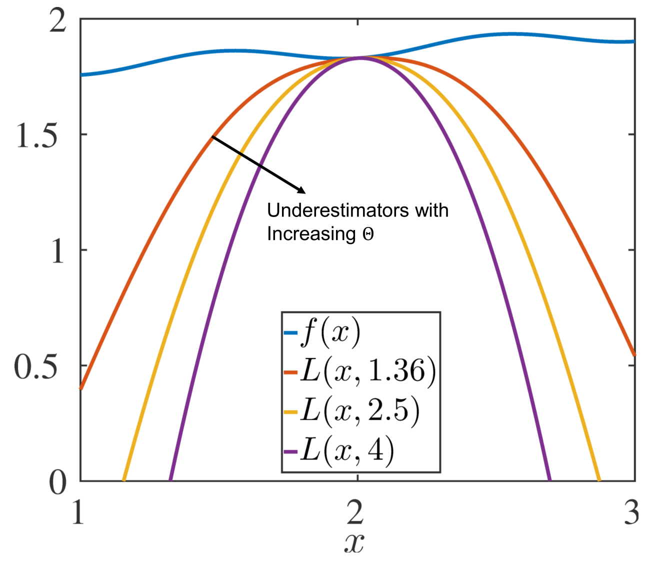


Figure 2.1: Illustration of the function and its ECU.

deployed that refines the bounds systematically so that they collapse to ϵ-optimality. To this end, a spatial branch-and-bound scheme is adopted that repeatedly partitions a rectangle into subrectangles. At each node, a lower bound using Eq. (2.5) is computed over the subdomain defined by the partitioning. The global lower bound is updated by taking the minimum of these lower bounds amongst all subrectangles. This way the algorithm generates a nondecreasing sequence of global lower bound as the iteration proceeds. The upper bound is a nonincreasing sequence of points generated by a local data-driven optimizer for different subrectangles. Any feasible point can also serve as a valid upper bound. However, a good upper bound obtained initially can expedite the convergence of the branch-and-bound algorithm. In this work, BOBYQA [53] is employed as the local solver to compute the upper bounds.

The following steps are involved in the proposed algorithm:

Step 1. Initialize a tolerance ϵ on the difference between lower and upper bound, the iteration counter $i = 0$, and the current region $M^0 \equiv [x^L, x^U]$. Obtain lower bound LB and upper bound UB. Let $\hat{x}^0$ denote the point at which UB is obtained.

Step 2. If $UB - LB \leq \epsilon$, or $\frac{UB-LB}{|LB|} \leq \epsilon$, then ϵ-global minima has been obtained. Report the global solution:

$$f^{opt} \leftarrow UB, \quad x^{opt} \leftarrow \hat{x}^i$$

Otherwise, go to the next step.

Step 3. Partition the current rectangle, M^i into two rectangles, M_L^i and M_R^i by dividing M^i along the longest edge.

$$M_L^i = \begin{bmatrix} x_1^{l,i} & x_1^{u,i} \\ \vdots & \vdots \\ x_j^{l,i} & \frac{(x_j^{l,i}+x_j^{u,i})}{2} \\ \vdots & \vdots \\ x_n^{l,i} & x_n^{u,i} \end{bmatrix}, \quad M_R^i = \begin{bmatrix} x_1^{l,i} & x_1^{u,i} \\ \vdots & \vdots \\ \frac{(x_j^{l,i}+x_j^{u,i})}{2} & x_j^{u,i} \\ \vdots & \vdots \\ x_n^{l,i} & x_n^{u,i} \end{bmatrix} \tag{2.10}$$

where j represents the variable with the longest edge of M^i.

Step 4. Use Eq. (2.5) to compute lower bounds for both M_L^i and M_R^i represented by LB_L^i and LB_R^i, respectively. Store the solutions and mark the rectangles as “unexplored" if they are less than the upper bound, UB. Update the overall lower bound by finding the minimum of the stored lower bounds:

$$LB = \min_{\mathbb{I}} \left[\min\{LB_L^{\mathbb{I}}, LB_R^{\mathbb{I}}\} \right]$$

where $\mathbb{I} = 0, \ldots, i$. Increase the iteration counter as $i = i + 1$.

Step 5. Let i^m denote the iteration at which the minima of the lower bound amongst “unexplored" rectangles is obtained. Update the current rectangle as follows:
If $LB = LB_L^{i^m}$, then

```
    $M^i = M_L^{im}$
else if $LB = LB_R^{im}$
    $M^i = M_R^{im}$
end if
```

The selected rectangle is removed from the "unexplored" category and the solution is deleted i.e., it is no longer considered when updating the lower bound, LB.

Step 6. Use the local optimizer to find the optima within M^i and denote the solution by $\tilde{f}^i$ and $\tilde{x}^i$. Update the upper bound as follows:

$$UB = \min\{UB, \tilde{f}^i\}$$

If $UB = \tilde{f}^i$, then the best point is updated, $\hat{x}^i = \tilde{x}^i$. Go to Step 2.

Note that in order to reduce function evaluations in Step 6, the local optimizer need not be run at each iteration. This will not affect the ability of the algorithm to find global solution. In our implementation, the local optimizer is applied at every iteration while allowing a maximum of 100 function evaluations.

2.3.1 Illustrative Example 2.2

The algorithm is illustrated on Problem 11a listed in Table 2.1. Although the objective function has an explicit expression, no derivative information is used. The only information that are used are the variable bounds and a valid upper bound on the diagonal elements of the Hessian. Although, the tightest upper bound of Θ (Θ^g) is 0.1, it is further relaxed and the value of Θ used is 5.06. The nodes at which the lower bound is evaluated are counted when determining the number of explored nodes.

The number of nodes explored for this illustrative example to converge to ϵ-global minima are 29 and the steps are illustrated in Figure 2.2. Figure 2.2 (a) illustrates the branch-and-bound

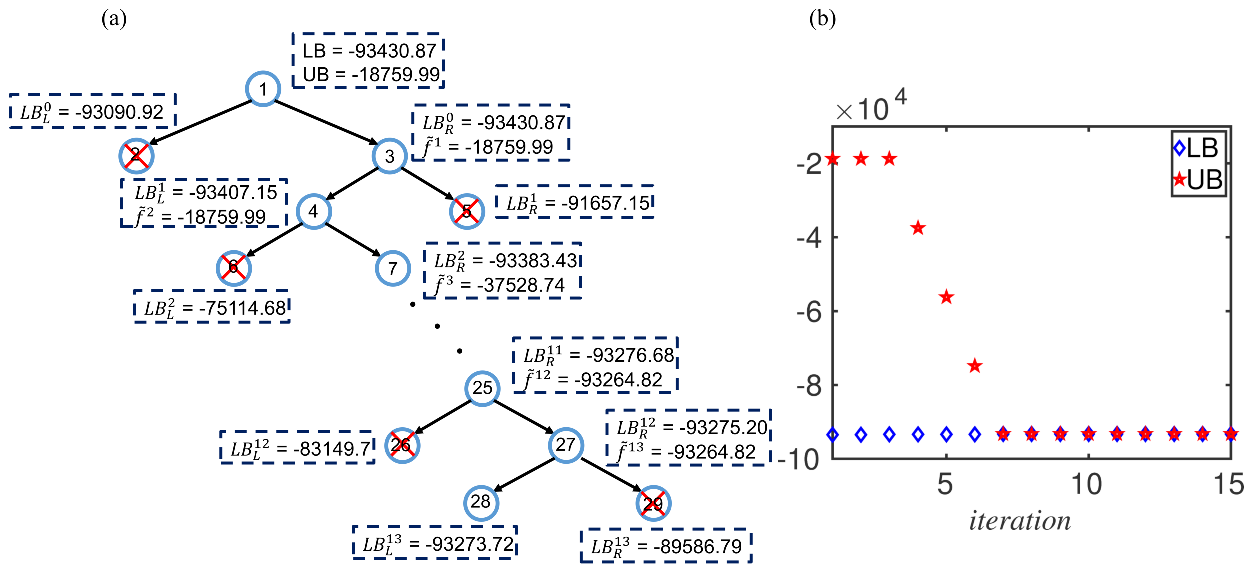


Figure 2.2: (a) Illustration of branch-and-bound algorithm on a 5-D black-box problem (Problem 11a in Table 2.1). (b) The progress of the algorithm with iterations.

tree explored and corresponding lower and upper bounds. Figure 2.2 (b) shows the nondecreasing sequence of lower bound and nonincreasing sequence of upper bounds obtained with each iteration. Note that the global minima of the problem is -93264.82. At root node, the local optimizer gives the value of the upper bound that is far from the global minima. The lower bound is obtained by minimizing the ECU. The initial rectangle is then divided into two subrectangles and lower bound is calculated for both of them. Depth-first search is employed to explore the nodes. Lower bound at node 3 is less than the lower bound at node 2. Therefore, node 3 is further explored by dividing the rectangle into two smaller subrectangles. The upper bound corresponding to the global minima is obtained at node 15. This enables pruning many nodes of the branch-and-bound tree, thereby expediting the convergence of the algorithm. The branch-and-bound requires 29 nodes and 908 evaluations to converge to ϵ-global optimum.

2.4 Computational Studies

The branch-and-bound algorithm is implemented in MATLAB R2016a and the problems are solved using a 64-bit Intel Xeon E5-2670 2.5 Ghz processor running Linux. The fortran subroutines of BOBYQA is called within MATLAB to obtain upper bound. The algorithm is implemented on a test suite of 24 problems [107, 105] to illustrate its effectiveness. The dimensions of the problems vary from 2 to 6. The details of the problem set and results are given in Table 2.1. The first column gives the problem number, the objective function is provided in column 2, the variable bounds and problem dimension is given in column 3 and 4 respectively. The global minima (f^{opt}) is provided in column 5. The parameter required to construct the edge-concave underestimator (Θ) in Eq. (2.5) for each of the problem is given in column 6. Column 7 shows the maximum separation distance between the black-box function and at the root node. Column 8 and 9 presents the number of nodes explored (N^{nodes}) and function evaluations (N^{evals}) needed for the algorithm to converge. The algorithm converges when either $UB - LB \leq \epsilon$; or $\frac{UB-LB}{|LB|} \leq \epsilon$. The value of the parameter ϵ used is 10^{-4}.

As expected, the algorithm converges to ϵ-global optimum for all the problems. One of the major factor affecting the performance is the quality of the underestimator constructed. For instance, problem 7, 9, 13, 14, 15 are edge-concave and the underestimator is the same as that of the original function. As a result, the global minima is obtained by exploring only few nodes and relatively less evaluations are needed. From the results of problem 10a, 10b and 10c; it can be concluded that with increasing dimension, the number of nodes and evaluations needed to converge increases. Note that the parameter Θ also plays a critical role in the convergence rate of the algorithm. Specifically, for problems 16 and 20, even though they are 2 and 3 dimensional problems, it requires a lot of function evaluations and nodes to obtain the global minima. The parameter Θ affects the tightness of the relaxation and for these two problems, the value of Θ is high leading to loose relaxation. Therefore, relatively more nodes and function evaluations are required to converge to the global optima.

Another important factor that influences the required number of nodes is the maximum separation distance (d_{sep}^{max}) between the edge-concave underestimator and the original function. In general, the number of nodes required to converge increase with increasing maximum separation distance. For some problems with $d_{sep}^{max} = 0$, the global minima is obtained at the root node while

Table 2.1: Test functions and results.

#	Objective function $f(x)$	$[x^L, x^U]^n$	n	f^{opt}	Θ^g	d_{sep}^{max}	N^{nodes}	N^{evals}
1	$-x_1x_2 + x_2x_3x_4 - x_1x_2x_3x_4 - x_1x_2x_3x_4x_5 + 0.01x_1^2 - 0.2x_5 - 50x_2^3 - x_1x_3^4x_4$	[0,1]	5	-53.19	0.01	0.0125	9	229
2	$10x_1^2 - 50x_2^3 + x_1^2x_3^{1.4}x_4 - 2x_5^{1.5} - 5x_1^{1.5}x_2x_3^{0.9}x_4^{2.7}x_5^{2.95} + 15x_1x_2x_3^{0.9} - x_1^{2.1}x_2^{2.3}x_3^{2.9}x_4^{4.5}x_5^2$	[1,2]	5	-17145.96	8.43	10.53	17	321
3	$-x_1x_2 + x_2x_3x_4 - x_1x_2x_3x_4 - x_1x_2x_3x_4x_5 + 0.01x_1^{2.5} - 0.2x_5^{0.5} - 50x_2^3 - x_1x_3^{4.4}x_4$	[1,2]	5	-528.67	0.027	0.033	1	56
4	$-x_1x_2 + x_2x_3x_4 - x_1x_2x_3x_4 - x_1x_2x_3x_4x_5 + 10x_1^{2.5} - 50x_2^3 - x_1x_3^{4.4}x_4$	[0,5]	5	-39085.58	41.95	1.31E+3	45	850
5	$x_2^{2.5} - 5x_2^3 + 5x_1x_2 - 10x_2x_3x_4 + 2x_1x_2x_3x_4 - x_1x_2x_3x_4x_5$	[0,1]	5	-14	0.0586	0.073	71	890
6	$-10x_1x_2 + 10x_1x_3 - 10x_1x_4 + 10x_1x_5 - 10x_2x_3 + 10x_2x_4 - 10x_2x_5 + 10x_3x_4 - 10x_3x_5 + 10x_4x_5 - 10x_1x_2x_3 + 10x_1x_2x_4 - 10x_1x_2x_5 + 10x_2x_3x_4 - 10x_2x_3x_5 + 10x_3x_4x_5 - 10x_1x_2x_3x_4 + 10x_1x_2x_3x_5 - 10x_1x_2x_4x_5 + 10x_2x_3x_4x_5 - 10x_1x_2x_3x_4x_5 + 0.01x_1^2 + 0.01x_2^2 + 0.01x_3^2 + 0.01x_4^2 + 0.01x_5^2$	[1,5]	5	-31248.75	0.01	0.2	1	69
7	$-x_1x_2 + x_2x_3x_4 - x_1x_2x_3x_4 + x_1x_2x_3x_4x_5$	[0,1]	5	-1	0	0	5	132
8	$2x_1x_2 + 2x_1x_3 + 2x_2x_3 - 0.2x_1x_2x_3 + 0.01x_1^2 + 0.01x_2^2 + 0.01x_3^2$	[-10,10]	3	-397	0.01	3	43	243
9	$-x_1x_2 + x_2x_3x_4$	[0,1]	4	-1	0	0	5	85
10a	$-\frac{1}{2}\sum_{i=1}^{n}(x_i^4 - 16x_i^2 + 5x_i)$	[-5,2]	3	-300	8	294	85	154
10b			4	-400	8	392	169	388
10c			5	-500	8	490	579	917
11	$x_2^{2.5} - 5x_2^3 + 5x_1x_2 - 10x_2x_3x_4 + 2x_1x_2x_3x_4 - x_1x_2x_3x_4x_5x_6$	[0,1]	6	-14	0.059	0.092	295	3010
12	$x_1x_2 + x_1x_2x_3$	[-1,1]	3	-2	0	0	9	72
13	$x_1x_2 - x_2x_3 - x_3x_4 + x_1x_2x_3 - x_1 + x_4$	[0,1]	4	-1	0	0	1	55

Table 2.1 continued.

#	Objective function $f(x)$	$[x^L, x^U]^n$	n	f^{opt}	Θ^g	d_{sep}^{max}	N^{nodes}	N^{evals}
14a	$x_1x_2 + x_1x_3 + x_1x_4 + x_1x_5 + x_1x_6 + x_2x_3 + x_2x_4 + x_2x_5 + x_2x_6 + x_3x_4 + x_3x_5 + x_3x_6 + x_4x_5 + x_4x_6 + x_5x_6 + x_1x_2x_3 - x_1x_2x_4 + x_1x_2x_5 + x_1x_2x_6 + x_2x_3x_4 + x_2x_3x_5 + x_2x_3x_6 + x_3x_4x_5 + x_3x_4x_6 + x_4x_5x_6 + x_1x_2x_3x_4 + x_1x_2x_3x_5 + x_1x_2x_3x_6 + x_1x_2x_4x_5 + x_1x_2x_4x_6 + x_1x_2x_5x_6 + x_1x_2x_3x_4x_6 + x_1x_2x_3x_5x_6 + x_1x_2x_3x_4x_5x_6 - (x_1 + 1)^{0.2} - x_2^2 - x_3^2 - x_4^2 - x_5^{2.1} + 0.1x_6^2 - 0.1(x_1+1)^{1.3} - 0.1x_2^3 - 0.1x_3^3 - 0.1(x_4+10)^{0.3} - 0.1x_5^3 - 0.1x_6^{2.1} - 0.01x_1^{4.1} - 0.01x_2^4 - 0.01x_3^4 - 0.01x_4^4 - 0.01x_5^4 - 0.01x_6^4 - x_1^5 - x_2^5 - x_3^5 - x_4^5 - x_5^5 - x_6^5 - x_1^6 - x_2^6 - x_3^6 - x_4^6 - x_5^6 - x_6^6$	[0,5]	6	-93264.83	0.1	3.75	15	484
14b		[0,1]	6	-7.73	0.1	0.15	49	1005
14	$x_1x_2 - x_2x_3 - x_3x_4 + x_1x_2x_3 - x_1 + x_4$	[0,1]	4	-1	0	0	1	55
15a	$x_1x_2 + x_2x_3x_4 + x_2x_4 - x_1x_3x_4x_5 + x_2x_3x_5 - x_1x_5$	[-1,1]	5	-4	0	0	1	66
15b		[1,3]	5	-180	0	0	1	72
16	$\sum_{i=1}^{n-1}(4x_i^2 - 2.1x_i^4 + \frac{x_i^6}{3} + x_ix_{i+1} - 4x_1^2 + 4x_{i+1}^4)$	[-2,2]	3	-13.37	125.6	1507.2	15145	11865
17	$\sum_{i=1}^{n}(-0.1cos(5\pi x_i) + x_i^2)$	[-1,1]	3	-0.3	13.34	40.02	1913	3002
18	$100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((1 - x_2)^2 + (1 - x_4)^2) + 19.8(2 - x_2 - x_4)$	[0,1]	4	0	601	601	1571	6616
19	$10\prod_{i=1}^{n}(x_i + 1) + 0.01\sum_{i=1}^{n} x_i^2$	[1,5]	5	320.05	0.01	0.2	17	173
20	$(1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2))(30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2))$	[-2,2]	2	3	8.86E+5	7.02E+6	1.76E+5	37815

for other problems, the branch-and-bound tree needs to be further explored. This occurs because the local optimizer is able to find an upper bound that is equal to global minima at the root node. Therefore, finding a good upper bound quickly is also critical to the convergence rate of the algorithm.

The performance of the proposed approach is compared with PSWARM (Particle Swarm optimizer) [48] and CMA-ES (Covariance Matrix Adaptation Evolution Strategy) [44]. Unlike the proposed deterministic algorithm, both PSWARM and CMA-ES are stochastic global methods. PSWARM performs global search step based on the particle swarm algorithm and local poll step based on coordinate search. CMA-ES is an evolutionary algorithm based on sampling from a normal distribution and updating the mean and covariance matrix such that the new samples are performed in the areas likely to have lower objective function value. In Table 2.1, the value of the parameter Θ is obtained by globally optimization of the diagonal elements of the Hessian matrix. Although this approach results in tight Θ value, it is computationally expensive and difficult to apply to the type of problems in (2.1) and (2.2). A practical approach for these problems and relatively computationally inexpensive alternative is using interval based methods. Therefore, the performance of the two methods to estimate Θ, i.e., global optimization and interval method is also compared. The interval calculations are performed using C-XSC [108].

The comparison of the aforementioned approaches is provided as data profile and shown in Figure 2.3. The performance metric used is the number of function evaluations since it is assumed that the simulations are computationally expensive. CMA-ES and PSWARM require an initial guess and for all the problems, both of them are provided the middle point as the initial guess. A time limit of 24 hours is provided for each of the solvers and default parameters are used for both the solvers. The two variants of the proposed approach (represented as EC-Θ^g and EC-Θ^{int}) are successful in solving more number of problems to global optimality using less function evaluations compared to other approaches. As expected, using global optimization to estimate Θ leads to a tighter relaxation compared to that obtained by interval method. Therefore, EC-Θ^g performs better than EC-Θ^{int}.

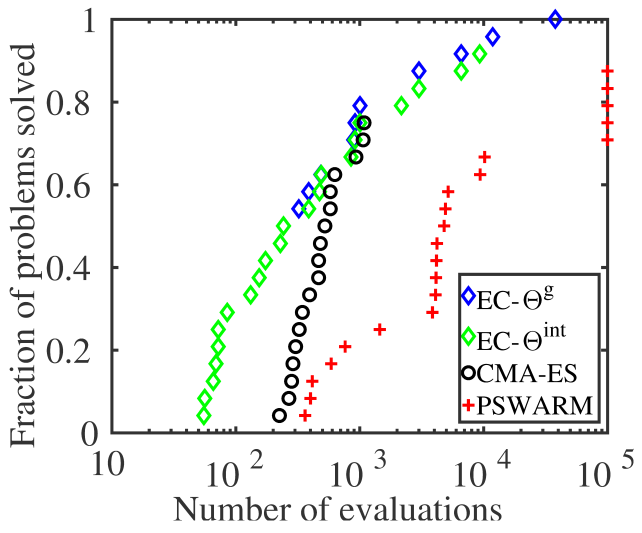


Figure 2.3: Data profile comparing the ability of different methods to solve a problem to global optimality. EC-Θ^g and EC-Θ^{int} represents the proposed method with Θ calculated by globally optimizing the diagonal elements of the Hessian matrix and interval method, respectively.

2.5 Conclusions

A class of grey-box problems is identified for which a valid underestimator can be constructed using maximum of the upper bounds of the diagonal elements of the Hessian of the original grey-box function. A strategy is proposed that enables finding valid lower bound for the grey-box problem. Since the algebraic form of the objective function is unknown, a general underestimator that does not depend on the mathematical structure is suitable. The methodology depends on the assumption that the upper bound on the diagonal elements of the Hessian of the original function (Θ) is known. This enables construction of a valid edge-concave underestimator of the grey-box problem. The lower bound is thereby obtained by enumerating the vertices of the polyhedral space and choosing the minimum. A local solver BOBYQA is applied to obtain valid upper bounds. To converge to ϵ-global minima, a branch-and-bound framework is utilized. The effectiveness of the method is shown on a set of 24 numerical problems from literature. The global minima is obtained in finite number of evaluations without performing dense sampling. The influence of the factors such as the tightness of the underestimator, problem dimension on the convergence rate of the algorithm has also been discussed.

3. GLOBAL OPTIMIZATION OF GREY-BOX PROBLEMS WITH EMBEDDED ODES

3.1 Introduction

In Chapter 2, a broad class of grey-box problems are defined for which global optimization is possible if the maximum of an upper bound on the diagonal elements of the Hessian is either available or it can be estimated. In this chapter, the focus is on identifying the problems that belong to this class of grey-box problem. It is recognized that for optimization problems with embedded system of ordinary differential equations, it is possible to estimate bounds of the Hessian elements using the model insights. Specifically, in this chapter, the following grey-box optimization problem is addressed:

$$\begin{aligned}
\min_{x} \quad & f_a(x) + f_d(y(x,t_f)) + \int_{t_0}^{t_f} \psi(t, y(x,t))dt \\
\textit{s.t.} \quad & g_{aj}(x) + g_{dj}(y(x,t_f)) + \int_{t_0}^{t_f} \xi_j(t, y(x,t))dt \le 0, \quad j \in \mathcal{J} \\
& \dot{y}_k = \phi_k(t, y, x), \quad \forall t \in (t_0, t_f], \;\; k \in \mathcal{K} \\
& y_k(x, t_0) = y_{0k}(x), \quad k \in \mathcal{K} \\
& x \in [x^L, x^U]
\end{aligned} \tag{P2}$$

where $t \in (t_0, t_f] \subseteq \mathbb{R}$ is time, $x \in [x^L, x^U] \subseteq \mathbb{R}^n$ are optimization variables, $y \in \mathbb{R}^p$ are state variables, and $\dot{y} \in \mathbb{R}^p$ denote derivatives of the state variables with respect to time. $\mathcal{J}$ and $\mathcal{K}$ are set of inequalities and state variables, respectively. Let $m = |\mathcal{J}|$ and $p = |\mathcal{K}|$. The functions $f_a(x)$ and g_{aj} are twice continuously differentiable with respect to $x \in [x^L, x^U]$ such that $f_a : \mathbb{R}^n \mapsto \mathbb{R}$ and $g_{aj} : \mathbb{R}^n \mapsto \mathbb{R}$, $j \in \mathcal{J}$. Similarly, the functions $f_d(y(x,t_f))$ and $g_{dj}(y(x,t_f))$ are twice continuously differentiable functions with respect to y for all $x \in [x^L, x^U]$ at $t = t_f$ such that $f_d : \mathbb{R}^p \mapsto \mathbb{R}$ and $g_{dj} : \mathbb{R}^p \mapsto \mathbb{R}$, $j \in \mathcal{J}$. The functions $\psi(t, y(x,t))$, $\xi_j(t, y(x,t))$ and $\phi_k(t, y, x)$ are Lebsegue integrable, twice continuously differentiable function with respect to y and $x \in [x^L, x^U]$ at all $t \in [t_0, t_f]$ such that $\psi : (t_0, t_f] \times \mathbb{R}^p \mapsto \mathbb{R}$, $\xi_j : (t_0, t_f] \times \mathbb{R}^p \mapsto \mathbb{R}$, $j \in \mathcal{J}$, and $\phi_k : (t_0, t_f] \times \mathbb{R}^p \times \mathbb{R}^n \mapsto \mathbb{R}$, $k \in \mathcal{K}$. Finally, y_{0k} is twice continuously differentiable function with respect to x such that $y_{0k} : \mathbb{R}^n \mapsto \mathbb{R}, k \in \mathcal{K}$.

P2 is considered grey-box because the objective function and constraints can not be expressed analytically in terms of decision variables but besides simulation data, other information (e.g. derivative) can be extracted based on model analysis. Solving P2 is of primary importance in many engineering areas. One application involves estimating kinetic parameters using time series data. In this case, a cost function measuring the discrepancy of the model with respect to the available experimental data is minimized while subjected to the dynamics of the system [109, 110, 111, 112, 113]. Another class of problems in this domain seeks time varying optimal control inputs that optimize certain performance metric under transient conditions [114, 115, 116, 117, 118, 15]. It has been recognized that even simple instances of P2 are often nonconvex and exhibit multiple local solutions [119].

The primary motivation in this chapter is to provide a deterministic global optimization method for P2. To this end, branch-and-bound is a deterministic method that guarantees convergence to ϵ-global optima as long as valid relaxations are generated [102]. While convex envelopes provide tightest lower bounds, they are available for only a handful of functions and it is often difficult to derive convex envelopes of dynamic systems with nonconvexity. Therefore, underestimators that have the ability to relax general nonconvex terms are more promising alternatives. The global optimization algorithm for solving P2, presented in this chapter, belongs to this category of deterministic and sequential approach based on a spatial branch-and-bound (B&B) framework.

One of the earliest deterministic algorithms for global optimization of problems with embedded ODEs was proposed by Esposito and Floudas [120, 121]. αBB [122] underestimator was generated to find a valid lower bound. It is constructed by adding a quadratic term to a general nonconvex function such that the convexity of the quadratic term overpowers the nonconvexity of the original function. A key challenge is estimating α parameters for the quadratic term that ensures convexity of the underestimator. To estimate the α parameters, Esposito and Floudas relied on sampling methods. Since the method relies on sampling, at times, it may lead to generation of invalid underestimator. Papamichail and Adjiman [123, 124] overcame this drawback by defining second order sensitivity equations and using differential inequalities to obtain bounds on the Hessian of the state variables. Chachuat and Latifi [125] used adjoint method instead of sensitivity equations for higher dimensional problems to obtain second order derivatives of the objective function.

Singer and Barton [126, 127, 128] developed a global optimization method by utilizing McCormick relaxations and illustrated its effectiveness on several case studies. Their method avoided calculating bounds on second-order sensitivities and required bounds on state variables only. Efforts have been made [129, 130, 131, 132, 133, 134, 135, 136, 137] to obtain tighter state bounds of nonlinear ODEs and further improve the performance of McCormick relaxation-based global optimization. Furthermore, Lin and Stadtherr [138, 139] estimated state bounds by utilizing a combination of interval analysis and Taylor models, and incorporated domain reduction technique. The method was also extended to handle problems with inequality path constraints [140].

The tightness of the underestimator is critical to the rate of convergence of B&B algorithm. Depending on the domain of the problem under consideration, one underestimator may be tighter than the other, as illustrated by Papamichail and Adjiman [123]. Therefore, there is a need to further explore other relaxation schemes. Recently, edge-concave underestimator [105] has been proposed as an alternative method to relax a general twice continuously differentiable function. Numerical examples suggest that edge-concave underestimator (ECU) can perform better than αBB in many instances. In this chapter, a method is proposed for global optimization of P2 that uses ECU to find valid lower bounds of P2. The details of constructing the underestimator will be discussed in the subsequent sections. The performance of the algorithm will be demonstrated by solving several benchmark problems. A hybrid approach is also discussed that combines different underestimators to further improve the performance of the global optimization algorithm.

The rest of the chapter is organized as follows. In Section 3.2, the details of constructing edge-concave underestimator for different terms and finding lower bound are provided. Section 3.3 gives an overview of the spatial branch and bound algorithm. Implementation details and the case studies are discussed in Section 3.4. Finally, concluding remarks are given in Section 3.5.

3.2 Constructing a Valid Relaxation

Computing lower bounds through relaxation of the original problem is key to convergence of a sBB algorithm. The mathematical form of ECU is given by Eq. (2.3). Some theoretical results are also presented that allows formulating a valid relaxation of P2 using edge-concave underestimator.

3.2.1 Properties of Edge-concave Underestimator

The following properties make an ECU attractive for global optimization applications.

Property 3.1. *[141] An edge-concave function defined on a polytope $\mathcal{B} = \{x \in \mathbb{R}^n : x^L \leq x \leq x^U\}$ attains its minima at a vertex $x = x^v$.*

Property 3.2. *Let $L_1(x)$, $L_2(x)$, ..., $L_S(x)$ be edge-concave functions such that $L_s : \mathbb{R}^n \mapsto \mathbb{R},\ \forall s = 1, \ldots, S$, then the function L_T defined as:*

$$L_T(x) = \sum_{s=1}^{S} L_s(x) \tag{3.1}$$

is also edge-concave.

Proof. Considering the diagonal elements of the Hessian of Eq. (3.1),

$$\frac{\partial^2 L_T}{\partial x_i^2} = \sum_{s=1}^{S} \frac{\partial^2 L_s}{\partial x_i^2}, \quad \forall i = 1, \ldots, n$$

By hypothesis, since each of $L_s,\ \forall s = 1, \ldots, S$ are edge-concave, the following holds by definition of an edge-concave function,

$$\frac{\partial^2 L_s}{\partial x_i^2} \leq 0, \quad \forall i = 1, \ldots, n;\ \forall\ s = 1, \ldots, S$$

This leads to $\frac{\partial^2 L_T}{\partial x_i^2} \leq 0,\ \forall i = 1, \ldots, n$, which implies that L_T is edge-concave. □

Property 3.3. *A function defined by the sum of edge-concave functions ($L_T(x) = \sum_{s=1}^{S} L_s(x)$) on a polytope $\mathcal{B} = \{x \in \mathbb{R}^n : x^L \leq x \leq x^U\}$ attains its minima at a vertex $x = x^v$.*

Proof. Proof follows trivially based on Property 3.1 and 3.2. □

Although ECU given in Eq. (2.3) can be directly used to relax a nonconvex problem, the resulting relaxed problem may be nonconvex and difficult to solve. To avoid this issue, it is advantageous to consider the form of ECU that results in formulation of a simpler and preferably a linear relaxed problem. Property 3.4 allows generation of tight linear relaxation of P2 by constructing linear facets of the convex envelope of the nonlinear ECU.

Property 3.4. *[141] An edge-concave function $L(x)$ has a vertex polyhedral convex envelope on a polytope $\mathcal{B} = \{x \in \mathbb{R}^n : x^L \leq x \leq x^U\}$. The linear facets of the convex envelope of $L(x)$ on $\mathcal{B}$ at a point x is given as follows [142]:*

$$\begin{aligned} CE(L,\mathcal{B})\big|_x = \underset{\lambda_v}{min} \quad & \sum_{v=1}^{2^n} \lambda_v L(x^v) \\ \textit{s.t.} \quad & \sum_{v=1}^{2^n} \lambda_v x^v = x \\ & \sum_{v=1}^{2^n} \lambda_v = 1 \\ & \lambda_v \geq 0, \quad \forall v \in 1,\dots,2^n \end{aligned} \tag{3.2}$$

where x^v denotes a vertex of the polytope $\mathcal{B}$.

3.2.2 Edge-concave Relaxation of P2

P2 is composed of two types of terms, namely algebraic and dynamic terms. The algebraic terms are dependent on only optimization variables, while the dynamic terms are composed of the state variables. The algebraic nonconvex terms ($f_a(x), g_{aj}(x)$) are relaxed by constructing ECUs for all. The ECU for $f_a(x)$ is as follows:

$$L_{f_a}(x) = f_a(x) - \sum_{i=1}^{n} \theta_i^{f_a}(x_i - x_i^M)^2 \tag{3.3}$$

where,

$$\theta_i^{f_a} = max\left\{0, \frac{1}{2}\left[\frac{\partial^2 f_a}{\partial x_i^2}\right]^U\right\} \tag{3.4}$$

Similarly, ECU for g_{aj} is given by:

$$L_{g_{aj}}(x) = g_{aj}(x) - \sum_{i=1}^{n} \theta_i^{g_{aj}}(x_i - x_i^M)^2 \tag{3.5}$$

where,

$$\theta_i^{g_{aj}} = max\left\{0, \frac{1}{2}\left[\frac{\partial^2 g_{aj}}{\partial x_i^2}\right]^U\right\} \tag{3.6}$$

ECU for $f_d(y(x,t_f))$ is defined as:

$$L_{f_d}(x) = f_d(y(x,t_f)) - \sum_{i=1}^{n} \theta_i^{f_d}(x_i - x_i^M)^2 \tag{3.7}$$

where,

$$\theta_i^{f_d} = max\left\{0, \frac{1}{2}\left[\frac{\partial^2 f_d}{\partial y^2}\left(\frac{\partial y}{\partial x_i}\right)^2\Big|_{t=t_f} + \frac{\partial f_d}{\partial y}\frac{\partial^2 y}{\partial x_i^2}\Big|_{t=t_f}\right]^U\right\} \tag{3.8}$$

Analogously, ECU for $g_{dj}(y(x,t_f))$ is as follows:

$$L_{g_{dj}}(x) = g_{dj}(y(x,t_f)) - \sum_{i=1}^{n} \theta_i^{g_{dj}}(x_i - x_i^M)^2 \tag{3.9}$$

where,

$$\theta_i^{g_{dj}} = max\left\{0, \frac{1}{2}\left[\frac{\partial^2 g_{dj}}{\partial y^2}\left(\frac{\partial y}{\partial x_i}\right)^2\Big|_{t=t_f} + \frac{\partial g_{dj}}{\partial y}\frac{\partial y^2}{\partial x_i^2}\Big|_{t=t_f}\right]^U\right\} \tag{3.10}$$

Finally, the edge-concave relaxation for the integral term is needed. The results are described that will allow the construction of the edge-concave relaxation of the integral by generating edge-concave relaxation of the integrand. Next, the integral monotonicity of the Lebesgue integrable function is stated.

Lemma 3.1. *[143] Let* $t \in (t_0, t_f]$, $x \in \mathbb{R}^n$ *and* $\zeta_1(t,x), \zeta_2(t,x)$ *are Lebesgue integrable function* *such that $\zeta_1, \zeta_2 : (t_0, t_f] \times \mathbb{R}^n \mapsto \mathbb{R}$. If the following condition holds:*

$$\zeta_1(t,x) \leq \zeta_2(t,x), \quad \forall\ t \in (t_0, t_f], x \in \mathbb{R}^n$$

then their corresponding integrals defined as $Z_1(x) = \int_{t_0}^{t_f} \zeta_1(t,x)dt$, $Z_2(x) = \int_{t_0}^{t_f} \zeta_2(t,x)dt$ are related as follows,

$$Z_1(x) \leq Z_2(x), \quad \forall x \in \mathbb{R}^n$$

To generate an edge-concave relaxation of the integral that is valid in the decision variable space, it is sufficient to construct edge-concave relaxation of the integrand in decision variable space $\mathbb{R}^n$ for each fixed time $t \in (t_0, t_f]$. This is formalized through Theorem 3.1.

Theorem 3.1. *Let $t \in (t_0, t_f]$, $x \in \mathbb{R}^n$ and $\zeta(t,x)$ be a Lebesgue integrable function such that $\zeta : (t_0, t_f] \times \mathbb{R}^n \mapsto \mathbb{R}$. If $\zeta(t,x)$ is edge-concave on decision variable space and for each fixed $t \in (t_0, t_f]$, then*

$$Z(x) = \int_{t_0}^{t_f} \zeta(t,x)dt \tag{3.11}$$

is edge-concave on $\mathbb{R}^n$.

Proof. Differentiating Eq. (3.11) twice with respect to the decision variable x_i,

$$\frac{\partial^2 Z}{\partial x_i^2} = \frac{\partial^2}{\partial x_i^2} \int_{t_0}^{t_f} \zeta(t,x)dt, \quad \forall i = 1, \dots, n \tag{3.12}$$

Using Leibniz's integral rule [144],

$$\frac{\partial^2 Z}{\partial x_i^2} = \int_{t_0}^{t_f} \frac{\partial^2 \zeta(t,x)}{\partial x_i^2} dt, \quad \forall i = 1, \dots, n$$

Since $\zeta(t,x)$ is edge-concave on $\mathbb{R}^n$ for each $t \in (t_0, t_f]$, the following should hold by definition of edge-concave function,

$$\frac{\partial^2 \zeta(t,x)}{\partial x_i^2} \leq 0, \quad \forall\ t \in (t_0, t_f], x \in \mathbb{R}^n, i = 1, \dots, n \tag{3.13}$$

Using Eq. (3.13) and Lemma 3.1 in Eq. (3.12) completes the proof. $\square$

The edge-concave underestimator for the integral term, $\Psi(x) = \int_{t_0}^{t_f} \psi(t, y(x,t))dt$ is explicitly written as follows:

$$L_\Psi(x) = \int_{t_0}^{t_f} \left[\psi(t, y(x,t)) - \sum_{i=1}^{n} \theta_i^{\psi}(x_i - x_i^M)^2\right] dt \tag{3.14}$$

where,

$$\theta_i^{\psi} = max\left\{0, \frac{1}{2}\left[\frac{\partial^2 \psi}{\partial y^2}\left(\frac{\partial y}{\partial x_i}\right)^2 + \frac{\partial \psi}{\partial y}\frac{\partial^2 y}{\partial x_i^2}\right]^U\right\} \tag{3.15}$$

Correspondingly, ECU for the integral term $\Xi_j(x) = \int_{t_0}^{t_f} \xi_j(t, y(x,t))dt$ is defined as:

$$L_{\Xi_j}(x) = \int_{t_0}^{t_f} \left[\xi_j(t, y(x,t)) - \sum_{i=1}^{n} \theta_i^{\xi_j}(x_i - x_i^M)^2\right] dt \tag{3.16}$$

where,

$$\theta_i^{\xi_j} = max\left\{0, \frac{1}{2}\left[\frac{\partial^2 \xi_j}{\partial y^2}\left(\frac{\partial y}{\partial x_i}\right)^2 + \frac{\partial \xi_j}{\partial y}\frac{\partial^2 y}{\partial x_i^2}\right]^U\right\} \tag{3.17}$$

Constructing valid edge-concave underestimators rely on reliably estimating the parameters defined in Eq. (3.4), (3.6), (3.8), (3.10), (3.15), and (3.17). Next, the methods to estimate these parameters are discussed.

3.2.3 Estimating $\theta_i^{f_a}$ and $\theta_i^{g_{aj}}$

By construction, ECUs L_{f_a} and $L_{g_{aj}}$ for the algebraic terms $f_a(x)$ and $g_{aj}(x)$ ensure that

$$L_{f_a}(x) \leq f_a(x), \text{ and } L_{g_{aj}}(x) \leq g_{aj}(x), \quad \forall x \in [x^L, x^U]$$

Here, the details of the construction of ECU for the term $f_a(x)$ are only discussed. The ECU for $g_{aj}, \forall j \in \mathcal{J}$ is constructed following similar arguments. Unlike αBB underestimator, ECU requires only diagonal elements of the Hessian of the function for which the underestimator is being constructed. In particular, the upper bound of the diagonal elements of the Hessian are needed. The maximum separation distance [105] between $f_a(x)$ and $L_{f_a}(x)$ considered in $x \in$

$[x^L, x^U]$ is equal to $\frac{1}{4}\sum_{i=1}^{n}\theta_i^{f_a}(x_i^U - x_i^L)^2$. Therefore, smaller values of the parameters $\theta_i^{f_a}$ and $\theta_i^{g_{aj}}$ results in tighter underestimator.

In certain cases, all the diagonal elements of a nonconvex function may be convex. In such cases, each of the elements can be locally optimized. However, for an arbitrary function, finding the smallest possible $\theta_i^{f_a}$ parameters require solving n nonconvex problems. Alternatively, interval methods [145] allows to compute lower and upper bound of a function given the range of variables. The bounds provided by interval methods may be loose, but they are obtained in computationally inexpensive manner.

3.2.3.1 *Illustrative Example 3.1*

Consider the following polynomial in two variables:

$$f_a(x) = x_1^5 - x_1 x_2^2, \quad \forall x_1, x_2 \in [0, 1]$$

The Hessian of the above function is

$$H_{f_a} = \begin{bmatrix} 20x_1^3 & -2x_2 \\ -2x_2 & -2x_1 \end{bmatrix}$$

The corresponding interval matrix is

$$[H_{f_a}] = \begin{bmatrix} [0, 20] & [-2, 0] \\ [-2, 0] & [-2, 0] \end{bmatrix}$$

Finally, the parameters $\theta_1^{f_a} = 10$ and $\theta_2^{f_a} = 0$. Since the diagonal elements of the Hessian are monotonic, the bounds given by the interval method are globally optimal. The corresponding expression for the ECU is given as follows:

$$L_{f_a}(x) = x_1^5 - x_1 x_2^2 - 10\left(x_1 - \frac{1}{2}\right)^2$$

3.2.4 Estimating $\theta_i^{f_d}$ and $\theta_i^{g_{dj}}$

Computing $\theta_i^{f_d}$ and $\theta_i^{g_{dj}}$ are more challenging than computing $\theta_i^{f_a}$ and $\theta_i^{g_a}$. The difficulty arises because the Hessian is implicitly dependent on the decision variables and requires evaluating bounds on second-order derivatives of state variables with respect to the decision variables. Lets consider the original set of ODEs (for brevity, the index k is excluded):

$$\begin{aligned} \dot{y} &= \phi(t, y, x) \\ y(x, t_0) &= y_0(x) \end{aligned} \tag{3.18}$$

A set of first-order sensitivity equations are obtained by differentiation Eq. (3.18):

$$\begin{aligned} \frac{\partial \dot{y}(x,t)}{\partial x_i} &= \frac{\partial \phi}{\partial y}\frac{\partial y}{\partial x_i} + \frac{\partial \phi}{\partial x_i} \\ \frac{\partial y(x,t_0)}{\partial x_i} &= \frac{\partial y_0}{\partial x_i} \end{aligned} \tag{3.19}$$

The above set of ODEs are differentiated again *w.r.t.* x to obtain second-order sensitivity equations:

$$\begin{aligned} \frac{\partial^2 \dot{y}(x,t)}{\partial x_i^2} &= \frac{\partial^2 \phi}{\partial y^2}\left[\frac{\partial y}{\partial x_i}\right]^2 + \frac{\partial \phi}{\partial y}\frac{\partial^2 y}{\partial x_i^2} + \frac{\partial^2 \phi}{\partial x_i^2} \\ \frac{\partial^2 y(x,t_0)}{\partial x_i^2} &= \frac{\partial^2 y_0}{\partial x_i^2} \end{aligned} \tag{3.20}$$

A key issue that needs to be addressed in order to solve sensitivity equations is the continuity and differentiability of $y(x,t)$. The solution of ODE system, $y(x,t)$, is in fact twice continuously differentiable with respect to the decision variables x, based on the assumptions on ϕ and the following results.

Assume that the function ϕ and its partial derivative with respect to y ($\frac{\partial \phi}{\partial y}$) is continuous over domain Π defined by the space of variables t, x, and y.

Theorem 3.2. *[146] If (t_0, x_0 and y_0) is a point in Π, positive real numbers β_1 and β_2 exists such that for*

$$|x - x_0| < \beta_1$$

the solution of Eq. (3.18) *defined as*

$$y = \chi(x, t)$$

that satisfies the initial condition $\chi(x, t_0) = y_0$ *is defined on the interval* $|t - t_0| < \beta_2$ *and is a continuous function of the variables t and* x.

Theorem 3.3. *[146] Let* $\frac{\partial \phi}{\partial x}$ *exist and be continuous in* Π. *If (*t_0, x_0, p_0*) is a point in* Π, *then there exists positive real numbers* β_1' *and* β_2' *such that for*

$$|x - x_0| < \beta_1' \text{ and } |t - t_0| < \beta_2'$$

the solution of Eq. (3.18)*,* $\chi(x, t)$ *that satisfies the initial condition* $\chi(x, t_0) = y_0$*, has continuous partial derivative with respect to* x *i.e.,* $\frac{\partial \chi(t,x)}{\partial x}$ *is continuous.*

Corollary 3.1. *[146] Let all the partial derivatives of* ϕ *with respect to the variables* x *and* y *exist up to the* k*-th order inclusive and are continuous, then the solution of Eq.* (3.18)*,* $\chi(x, t)$ *also have partial derivatives with respect to the parameters* x *up to the* k*-th order inclusive that are continuous.*

From Eq. (3.7), it is apparent that the bounds on the state variables and their first-order and second-order derivatives with respect to decision variables are needed. The B&B algorithm considered here is in x space and, therefore, the bounds on x are refined automatically. However, since y is not an optimization variable, the bounds of y need to be derived separately. It is theoretically possible to ensure convergence to ϵ global optimality even if $\theta_i^{f_d}$ computed at the root node is used in the subsequent nodes. This is due to the fact that as $||x^u - x^l|| \to 0$, by construction of the underestimator, $L_{f_d} \to f_d$. However, the convergence of the algorithm may be slow.

The solution of the set of ODEs given in Eqs. (3.18) - (3.20) are dependent on the realization of a particular value of the decision variables. Essentially, the right hand side of the ODE is a set of twice continuously differentiable functions. Therefore, the lower and upper bounds on the state variables ($y \in [y^l, y^u]$) depends on the bounds of the decision variables ($x \in [x^l, x^u]$). Next, the results based on differential inequalities [147] are presented that provides a practical approach to compute bounds on the state variables that are valid for each $t \in [t_0, t_f]$. The bounds on first-order and second-order derivatives of the state variables can be computed following same reasoning.

The bounds on the state variables for system of ODE given in Eq. (3.18) are now derived. Let $y(t, x)$ denote its solution for each $t \in [t_0, t_f]$, $x \in \mathcal{B} = [x^l, x^u] \subseteq \mathbb{R}^n$. The lower and upper bounds on the state variables for each $t \in [t_0, t_f]$ are denoted as $y^l(t)$ and $y^u(t)$, respectively, such that $y^l(t), y^u(t) \in Y(t)$.

Theorem 3.4. *[123] Let ϕ be continuous such that it satisfies a uniqueness condition on $(t_0, t_f] \times \mathbb{R}^p \times B$. If,*

$$\dot{y}_k^l = \omega_k^l(t, y^l, y^u, x^l, x^u) \leq \underset{\substack{z \in Y(t), x \in B, \\ z_k = y_k^l(t)}}{\textit{inf}} \phi_k(t, z, x), \quad \forall k \in \mathcal{K}$$

$$\dot{y}_k^u = \omega_k^u(t, y^l, y^u, x^l, x^u) \geq \underset{\substack{z \in Y(t), x \in B, \\ z_k = y_k^u(t)}}{\textit{sup}} \phi_k(t, z, x), \quad \forall k \in \mathcal{K}$$

and,

$$y_k^l(t_0) = \omega_k^l(t_0, x_l, x_u) \leq \underset{x \in B}{\textit{inf}}\ y_{k0}(x), \quad \forall k \in \mathcal{K}$$

$$y_k^u(t_0) = \omega_k^u(t_0, x_l, x_u) \geq \underset{x \in B}{\textit{sup}}\ y_{k0}(x), \quad \forall k \in \mathcal{K}$$

where $z \in [y^l(t), y^u(t)]$, then the following holds true:

$$y_k^l(t) \leq y(t, x) \leq y_k^u(t), \quad \forall t \in [t_0, t_f], x \in B, k \in \mathcal{K}$$

Theorem 3.4 provides a practical way to compute state bounds for each $t \in [t_0, t_f]$. It requires deriving lower and upper bounds of the right side of Eq. (3.18) that are valid for both x and y. Interval method is a realistic approach that can compute these bounds in an inexpensive manner. In the differential equation corresponding to the lower bound of $k\text{-}th$ state variable, the corresponding state variable is fixed to the lower bound. However, interval arithmetic is applied for all other state variables. Same strategy is employed for computing upper bound of the $k\text{-}th$ state variable.

It is worth mentioning that there are alternative approaches [130, 131, 129, 132, 134, 135] to interval method that may yield tighter bounds. However, interval method is used for deriving state bounds.

3.2.4.1 Illustrative Example 3.2

Consider the following function:

$$f_d = -y(1,x)^2$$

with the governing ODE given by:

$$\begin{aligned} &\dot{y} = -y + x^3, \quad \forall t \in (0,1] \\ &y(0,x) = 1 \end{aligned} \tag{3.21}$$

where, $x \in [-1,1]$. Differentiating f_d twice *w.r.t* x yields the following expression of θ^{f_d}:

$$\theta^{f_d} = max\left\{0, \frac{1}{2}\left[-2\left(\frac{\partial y(1,x)}{\partial x}\right)^2 - 2y(1,x)\frac{\partial^2 y(1,x)}{\partial x^2}\right]^U\right\}$$

Further simplifying the above expression yields:

$$\theta^{f_d} = max\left\{0, -\text{inf}\left(\frac{\partial y(1,x)}{\partial x}\right)^2 - \text{inf}\left(y(1,x)\frac{\partial^2 y(1,x)}{\partial x^2}\right)\right\}$$

Let $y_1 = \frac{\partial y}{\partial x}$, $\dot{y}_1 = \frac{\partial}{\partial t}\left(\frac{\partial y}{\partial x}\right)$, $y_2 = \frac{\partial y_1}{\partial x}$ and $\dot{y}_2 = \frac{\partial}{\partial t}\left(\frac{\partial y_1}{\partial x}\right)$. These are used to define the first and second-order sensitivity equations as follows:

$$\begin{aligned} &\dot{y}_1 = -y_1 + 3x^2, \quad \forall t \in (0,1] \\ &\dot{y}_2 = -y_2 + 6x, \quad \forall t \in (0,1] \\ &y_1(0,x) = 0 \\ &y_2(0,x) = 0 \end{aligned} \tag{3.22}$$

Based on Theorem 3.4, the following set of ODE is solved to compute bounds on the state variables of ODE system given by Eq. (3.22):

$$\begin{aligned}
\dot{y}^l &= -y^l - 1, \quad \forall t \in (0,1] \\
\dot{y}_1^l &= -y_1^l, \quad \forall t \in (0,1] \\
\dot{y}_2^l &= -y_2^l - 6, \quad \forall t \in (0,1] \\
\dot{y}^u &= -y^u + 1, \quad \forall t \in (0,1] \\
\dot{y}_1^u &= -y_1^u + 3, \quad \forall t \in (0,1] \\
\dot{y}_2^u &= -y_2^u + 6, \quad \forall t \in (0,1] \\
y^l(0) = 1,\ y_1^l(0) &= 0,\ y_2^l(0) = 0,\ y^u(0) = 1,\ y_1^u(0) = 0,\ y_2^u(0) = 0
\end{aligned} \tag{3.23}$$

Finally, θ^{f_d} is given by

$$\theta^{f_d} = max\left\{0, -y_1^c(1) - min[y^l(1)y_2^l(1), y^l(1)y_2^u(1), y^u(1)y_2^l(1), y^u(1)y_2^u(1)]\right\}, \tag{3.24}$$

where,

$$y_1^c(1) = \begin{cases} 0 & \text{if } y_1^l(1) \le 0 \text{ and } y_1^u(1) \ge 0 \\ min\{(y_1^l(1))^2, (y_1^u(1))^2\} & \text{otherwise} \end{cases}$$

The bounds on the state variable (y), its first-order derivative (y_1) and second-order derivative (y_2) at $t = 1$ are obtained by solving the system of ODEs in Eq. (3.23). Once the desired bounds are computed, Eq. (3.24) yields $\theta^{f_d} = 3.79$.

3.2.5 Estimating θ_i^{ψ} and $\theta_i^{\xi_j}$

Computing θ_i^{ψ} and $\theta_i^{\xi_j}$ enables construction of ECUs for the integral terms $\int_{t_0}^{t_f} \psi(t, y(x,t))dt$ and $\int_{t_0}^{t_f} \xi(t, y(x,t))dt$, respectively. To keep the discussion simple, the focus here is on computing θ_i^{ψ} and same arguments are applicable to computing $\theta_i^{\xi_j}$. From the definition of θ_i^{ψ} in Eq. (3.15), the bounds on the state variables and their first-and second-order derivatives are needed for each $t \in [t_0, t_f]$. Theorem 3.4 is employed to compute the bounds. Note that Theorem 3.4 is also used to determine $\theta_i^{f_d}$, as discussed previously. However, a key difference is that in determining $\theta_i^{f_d}$, only the value of the bounds at $t = t_f$ is of interest, whereas θ_i^{ψ} depends on the bounds for each $t \in [t_0, t_f]$.

3.2.5.1 Illustrative Example 3.3

Consider the following function:

$$\psi(t, y(x,t)) = -y^2$$

The system of ODEs considered is the same as that given in Eq. (3.21). The parameter θ^{ψ}, corresponding to the edge-concave underestimator $L_{\Psi}(x)$, is given as follows:

$$\theta^{\psi}(t) = max\Bigg\{0, -y_1^c(t) - min[y^l(t)y_2^l(t), y^l(t)y_2^u(t), y^u(t)y_2^l(t), y^u(t)y_2^u(t)]\Bigg\}, \quad \forall t \in [0,1] \tag{3.25}$$

where,

$$y_1^c(t) = \begin{cases} 0 & \text{if } \ y_1^l(t) \leq 0 \ \text{ and } \ y_1^u(t) \geq 0 \\ min\{(y_1^l(t))^2, (y_1^u(t))^2\} & \text{otherwise} \end{cases}$$

The required bounds on the state variables are obtained by solving Eq. (3.23), which are subsequently used in Eq. (3.25) to compute $\theta^{\psi}(t)$. Figure 3.1 shows the resulting θ^{ψ} for each $t \in [0,1]$.

3.2.6 Relaxed Problem Formulations

In the previous sections, the methodologies to develop edge-concave underestimators corresponding to different classes of nonlinear terms involved in P2 are discussed. In this section, the formulation to compute lower bound for P2 is presented. The formulation of relaxed problem for a special form of P2 with no inequality constraints (i.e., $\mathcal{J}$ is empty) is first examined and then the relaxed problem for P2 in its original form will be formulated.

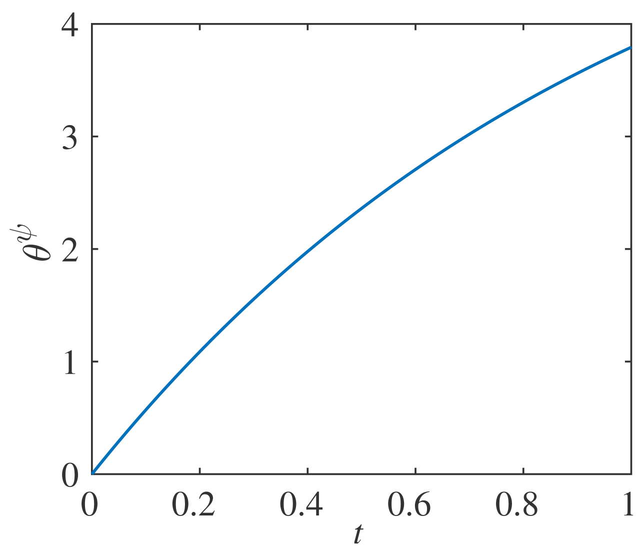


Figure 3.1: Illustrating θ^{ψ} as a function of t for Example 3.

3.2.6.1 Relaxation of a Special Form of P2

A special form of P2 with no inequality constraints is as follows:

$$\begin{aligned} \min_{x} \quad & f_a(x) + f_d(y(x,t_f)) + \int_{t_0}^{t_f} \psi(t, y(x,t))dt \\ \textit{s.t.} \quad & \dot{y}_k = \phi_k(t,y,x), \quad \forall t \in (t_0, t_f], \;\; k \in \mathcal{K} \\ & y_k(x,t_0) = y_{0k}(x), \quad k \in \mathcal{K} \\ & x \in [x^L, x^U] \end{aligned} \tag{P3}$$

Replacing each of the nonlinear terms $f_a(x)$, $f_d(y(x,t_f))$, and $\int_{t_0}^{t_f} \psi(t, y(x,t))dt$ in the objective function with the edge-concave underestimators $L_{f_a}(x)$, $L_{f_d}(x)$, and $L_{\Psi}(x)$, respectively, yields the following relaxation:

$$\begin{aligned} L_{EC} = \min_{x} \quad & L_{f_a}(x) + L_{f_d}(x) + L_{\Psi}(x) \\ \textit{s.t.} \quad & L_{f_a}(x) = f_a(x) - \sum_{i=1}^{n} \theta_i^{f_a}(x_i - x_i^M)^2 \\ & L_{f_d}(x) = f_d(y(x,t_f)) - \sum_{i=1}^{n} \theta_i^{f_d}(x_i - x_i^M)^2 \\ & L_{\Psi}(x) = \int_{t_0}^{t_f} \left[\psi(t, y(x,t)) - \sum_{i=1}^{n} \theta_i^{\psi}(x_i - x_i^M)^2 \right] dt \end{aligned} \tag{R3}$$

where $y(x,t)$ is given by solving following ODEs:

$$\begin{aligned} \dot{y}_k &= \phi_k(t, y, x), \quad \forall t \in (t_0, t_f], \;\; k \in \mathcal{K} \\ y_k(x, t_0) &= y_{0k}(x), \quad k \in \mathcal{K} \end{aligned}$$

At every iteration of the B&B algorithm, R3 needs to be solved to yield lower bounds. The ECUs $L_{f_d}(x)$ and $L_{\Psi}(x)$ depends on the state variables and therefore, the original system of ODEs need to be solved. R3 can be solved in a sequential manner, i.e., the system of ODEs is solved at x determined by the optimization algorithm.

Property 3.3 allows to solve R3 by simply evaluating the sum of the ECUs at the vertices of the polytope and choosing the minimum value. (R3) can be rewritten as follows:

$$\begin{aligned} L_{EC} = \min_{x \in \mathcal{V}} \quad & L_{f_a}(x) + L_{f_d}(x) + L_{\Psi}(x) \\ \textit{s.t.} \quad & L_{f_a}(x) = f_a(x) - \sum_{i=1}^{n} \theta_i^{f_a}(x_i - x_i^M)^2 \\ & L_{f_d}(x) = f_d(y(x,t_f)) - \sum_{i=1}^{n} \theta_i^{f_d}(x_i - x_i^M)^2 \\ & L_{\Psi}(x) = \int_{t_0}^{t_f} \left[\psi(t, y(x,t)) - \sum_{i=1}^{n} \theta_i^{\psi}(x_i - x_i^M)^2 \right] dt \end{aligned} \tag{R3}$$

where $y(x,t)$ is given by solving following ODEs:

$$\dot{y}_k = \phi_k(t,y,x), \quad \forall t \in (t_0, t_f], \;\; k \in \mathcal{K}$$
$$y_k(x,t_0) = y_{0k}(x), \quad k \in \mathcal{K}$$

where $\mathcal{V}$ denotes the set of vertices of the polytope $\mathcal{B} = \{x \in \mathbb{R}^n : x^L \leq x \leq x^U\}$. Besides generating tight underestimator, it is also desirable for the fast convergence of B&B algorithm that the computational cost at each node be kept at minimum. Solving ODE is the most computationally demanding step and, therefore, it is advantageous that function calls be kept at minimum. In this respect, edge-concave based relaxation may offer additional computational benefits. This is due to the fact that other underestimators, such as αBB, rely on solvers that may take many evaluations for even simple problems to find the lower bound.

3.2.6.2 A Hybrid Method to Compute Lower Bound

Solving auxiliary set of ODEs based on Theorem 3.4 to compute $\theta_i^{f_d}$ and θ_i^{ψ} yields bounds on state variables (y), first-and second-order derivatives of the state variables *w.r.t* the decision variables. An alternative lower bound can be obtained by interval method using the bounds on state variables $y \in [y^l, y^u]$ and decision variables $x \in [x^l, x^u]$ only and is given by:

$$L_{INT} = \inf\Big\{ f_a([x^l,x^u]) + f_d([y^l([x^l,x^u],t_f), y^u([x^l,x^u],t_f)]) + \int_{t_0}^{t_f} \psi(t,[y^l([x^l,x^u],t), y^u([x^l,x^u],t)])dt \Big\} \tag{3.26}$$

It is possible that there are subregions of the original feasible space where $L_{INT} \geq L_{EC}$. Therefore, it is advantageous to use a hybrid approach by choosing the tighter amongst the two:

$$L_{HYB-EC} = max\{L_{INT}, L_{EC}\} \tag{3.27}$$

where L_{HYB-EC} is a valid lower bound of P3.

3.2.6.3 Relaxation of P2

Similar to P3, the nonconvex terms $f_a(x)$, $f_d(y(x,t_f))$, and $\int_{t_0}^{t_f}\psi(t,y(x,t))dt$ are replaced by $L_{f_a}(x)$, $L_{f_d}(x)$, and $L_\Psi(x)$, respectively. All nonconvex terms in the $j\text{-}th$ constraint such as $g_{aj}(x)$, $g_{dj}(y(x,t_f))$ and $\int_{t_0}^{t_f}\xi_j(t,y(x,t))dt$ are also replaced by their edge-concave underestimators $L_{g_{aj}}(x)$, $L_{g_{dj}}(x)$, and $L_{\Xi_j}(x)$, respectively. This results in the formulation of the following nonlinear relaxed problem:

$$\begin{aligned} \min_x \quad & L_{f_a}(x) + L_{f_d}(x) + L_\Psi(x) \\ \textit{s.t.} \quad & L_{g_{aj}}(x) + L_{g_{dj}}(x) + L_{\Xi_j}(x) \le 0, \quad \forall j \in \mathcal{J} \\ & x \in [x^L, x^U] \end{aligned} \tag{R2}$$

Since L_{f_d}, L_Ψ, $L_{g_{dj}}$ and L_{Ξ_j} are dependent on the state variables, these are obtained by solving the following ODEs:

$$\dot{y}_k = \phi_k(t,y,x), \quad \forall t \in (t_0,t_f],\;\; k \in \mathcal{K}$$
$$y_k(x,t_0) = y_{0k}(x), \quad \forall t \in (t_0,t_f],\;\; k \in \mathcal{K}$$

Although R2 includes only algebraic terms, it is still nonlinear and nonconvex. Therefore, it is beneficial to further relax it and formulate an easier problem to compute the lower bound of P2. This can be done by considering the linear facets of the vertex polyhedral convex envelopes of the edge-concave underestimators [105]. Meyer and Floudas [142] proposed an algorithm based on geometric arguments to construct the following unique facets of the convex envelope of $L(x)$ on polytope $\mathcal{B} = \{x \in \mathbb{R}^n : x^L \le x \le x^U\}$:

$$CE[L(x),\mathcal{B}] \equiv \left\{\gamma \ge \sum_{i=1}^{n} \eta_{i,l}x_i + \eta_{0,l}, \quad l \in \mathcal{L}\right\} \tag{3.28}$$

Here $\mathcal{L}$ denotes set of unique facets that determines the convex envelope, whereas $\eta_{i,l}$ and $\eta_{0,l}$ are parameters defining the linear facet. Although using Eq. (3.28) may result in the formulation of a relaxed problem, it is only applicable to problems of upto 3 dimensions. Therefore, the linear facets defined in Eq. (3.2) are used to formulate the linear relaxed problem. Lets define $F_0(x) = L_{f_a}(x) + L_{f_d}(x) + L_{\Psi}(x)$ and $F_j(x) = L_{g_{aj}}(x) + L_{g_{dj}}(x) + L_{\Xi_j}(x), \forall j \in \mathcal{J}$ yielding the following linear relaxation of P2:

$$
\begin{aligned}
L_{EC} = \min_{x,\lambda_{j',v}} \quad & \upsilon_0 \\
\textit{s.t.} \quad & \upsilon_j \leq 0, \quad \forall j \in \mathcal{J} \\
& \upsilon_{j'} \geq \sum_{v=1}^{2^n} \lambda_{j',v} F_{j'}(x^v), \quad j' = 0, \ldots, m \\
& \sum_{v=1}^{2^n} \lambda_{j',v} x^v = x, \quad j' = 0, \ldots, m \\
& \sum_{v=1}^{2^n} \lambda_{j',v} = 1, \quad j' = 0, \ldots, m \\
& \lambda_{j',v} \geq 0, \quad j' = 0, \ldots, m; \;\; v = 1, \ldots, 2^n \\
& x \in [x^L, x^U]
\end{aligned}
\tag{R2-L}
$$

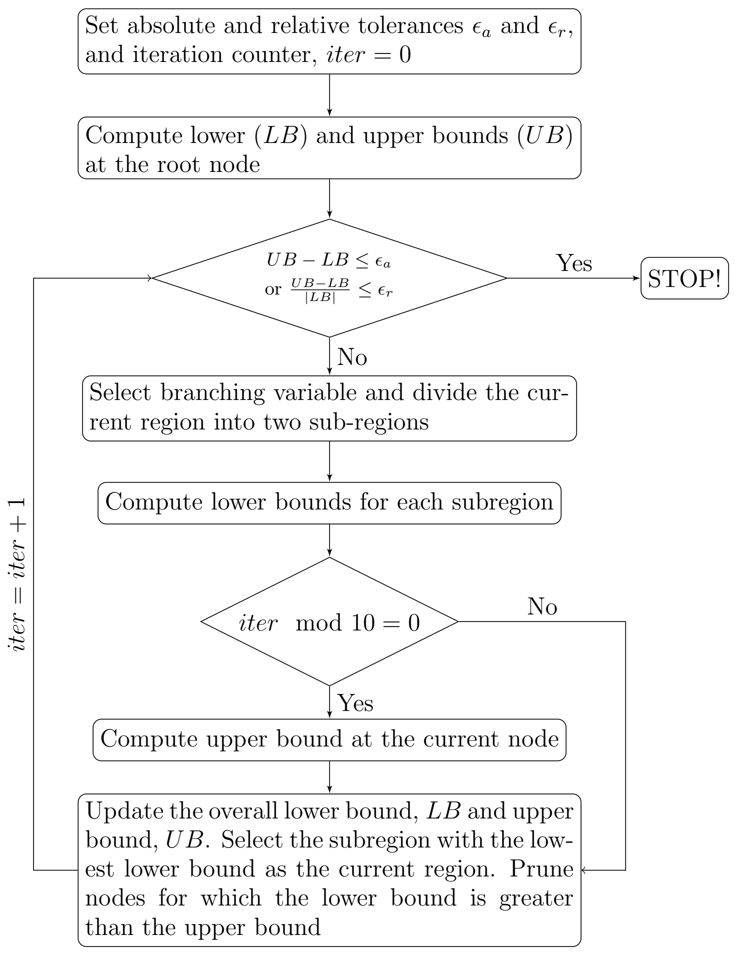
where each of the functions $F_{j'}$ are evaluated at $x = x^v$ by solving the following set of ODEs:

$$
\begin{aligned}
&\dot{y}_k = \phi_k(t, y, x^v), \quad \forall t \in (t_0, t_f]; k \in \mathcal{K} \\
&y_k(x^v, t_0) = y_{0k}(x^v), \quad \forall k \in \mathcal{K}
\end{aligned}
$$

It is possible to formulate a similar linear relaxed problem corresponding to R3. However, this would be unwarranted since the solution of R3 lies at a vertex and generating convex envelopes will add to the computational cost.

3.3 Overall Algorithm

A spatial branch-and-bound (sBB) algorithm is used for deterministic global optimization of P2. A brief overview of the algorithm is provided in Figure 3.2. The algorithm functions by dividing the space into subregions and eliminating those regions that can not contain the optimum. At any node of the B&B tree, subregions with the lower bound that are greater than the overall



Set absolute and relative tolerances ϵ_a and ϵ_r, and iteration counter, $iter = 0$
Compute lower (LB) and upper bounds (UB) at the root node
$UB - LB \leq \epsilon_a$ or $\frac{UB-LB}{|LB|} \leq \epsilon_r$
Yes
STOP!
No
Select branching variable and divide the current region into two sub-regions
Compute lower bounds for each subregion
$iter \mod 10 = 0$
No
Yes
Compute upper bound at the current node
Update the overall lower bound, LB and upper bound, UB. Select the subregion with the lowest lower bound as the current region. Prune nodes for which the lower bound is greater than the upper bound
$iter = iter + 1$

Figure 3.2: An overview of the spatial branch-and-bound algorithm for solving P2 and P3.

upper bound are eliminated. The sub-regions where the relaxed problem is infeasible are also eliminated. The algorithm consists of three main steps: (1) computing lower bounds using valid underestimators, (2) computing upper bound, and (3) selecting branching variable to divide the current region into subregions. The algorithm converges to global optimum within ϵ tolerance [102] if the bounding and selection operations are consistent and bound improving, respectively. The procedure to construct valid relaxations based on edge-concave underestimators (R3 or R2-L) was discussed in Section 3.2. Next, the branching operation, computation of lower and upper bounds, and the convergence criteria are briefly discussed.

3.3.1 Branching Rule

In the branching operation, the current region is further partitioned into two subregions. At each node, the branching variable x_b corresponding to the longest edge of the current region is selected:

$$x_b = \arg\max_i |x_i^u - x_i^l|$$

In case there are more than one variable corresponding to the largest range, the variable with the lowest index is selected for branching. Thereafter, the two subregions are obtained by bisecting the branching variable. There are other branching and range reduction strategies [148, 149] that can accelerate the convergence of the sBB algorithm. However, the branching rule discussed above is used to keep the implementation simple.

3.3.2 Computing Lower Bound: Solving R3 and R2-L

First, the technique used to solve R3 to obtain valid lower bounds is discussed. As mentioned before, Property 3.3 allows finding lower bound by computing objective function at the vertices of the polytope $\mathcal{B}$ and taking the minimum. The focus here is on describing the approach used to evaluating L_Ψ at $x = x^v$. The two terms comprising L_Ψ are separated as follows:

$$L_\Psi(x) = L_\Psi^1(x) - L_\Psi^2(x)$$

The terms L_Ψ^1 and L_Ψ^2 are defined as follows:

$$\begin{aligned} L_\Psi^1(x) &= \int_{t_0}^{t_f} \Big[\psi(t, y(x,t))\Big] dt \\ L_\Psi^2(x) &= \int_{t_0}^{t_f} \Big[\sum_{i=1}^{n} \theta_i^\psi (x_i - x_i^M)^2\Big] dt \\ &= \sum_{i=1}^{n} (x_i - x_i^M)^2 \Theta_i^\psi(t_f) \end{aligned}$$

where, $\Theta_i^{\psi} = \int_{t_0}^{t_f} \theta_i^{\psi} dt$. The details of the method used to compute Θ_i^{ψ} is given in Section 3.2.5. Since Θ_i^{Ψ} depends on the bounds of the optimization variables, it is calculated before computing lower bound. These parameters computed at the parent node of the branch and bound tree is also valid for the child nodes. Since computing Θ^{Ψ} and $\theta_i^{f_d}$ requires solving larger system of ODEs, it is possible to compute them only at the root note or parent node without affecting the theoretical convergence of the algorithm. However, this may lead to construction of loose relaxations and consequently, exploring more nodes of the sBB tree. To avoid this, θ parameters are updated at each node of the sBB tree to generate tight underestimators. L_{Ψ}^1 is computed at $x = x^v$ by solving the original system of ODEs. Finally, the value of L_{Ψ} at $x = x^v$ is:

$$L_{\Psi}(x^v) = \Psi(x^v) - \Theta^{\psi}(t_f)\sum_{i=1}^{n}(x_i^v - x_i^M)^2$$

where $\Psi(x^v) = \int_{t_0}^{t_f} \psi(t, y(x^v, t))dt$. L_{f_d} is evaluated at $x = x^v$ analogously by computing $\theta_i^{f_d}$ first and then solving the original set of ODEs to obtain the value of state variables at $t = t_f$. To compute lower bounds of P2, the linear relaxed problem R2-L is solved. It involves computing each of the ECUs at all the vertices of the polytope of $\mathcal{B}$ using the same technique as discussed previously. This results in a linear relaxed problem that is solved using CPLEX.

3.3.3 Computing Upper Bound

Any feasible point or a locally optimal solution of P2 will serve as a valid upper bound. Although finding a feasible point may need less simulations than locally optimizing P2, finding a good upper bound can help eliminate many subregions and accelerate the convergence of the branch and bound algorithm. To balance this trade-off, local optimization is performed after every 10 iterations using `fmincon`. A sequential approach is used, where the simulation is performed at a point x that is determined by the local optimization. The first-order derivative information is provided by solving the sensitivity equations along with the original ODEs.

3.3.4 Convergence Criteria

The maximum separation distance between a function and its ECU on a polytope $\mathcal{B} = \{x \in \mathbb{R}^n : x^l \leq x \leq x^u\}$ is given by [105]:

$$d_{max} = \frac{1}{4}\sum_{i=1}^{n} \theta_i (x_i^u - x_i^l)^2 \tag{3.29}$$

The sBB algorithm operates by dividing the current region into smaller subregions. As the size of the region becomes smaller, i.e., $||x^u - x^l|| \to 0$, by construction, $LB \to UB$. From Eq. (3.29), it is clear that unless the original function is edge-concave ($\theta_i = 0, i = 1, \ldots, n$), the separation distance can not be reduced to exactly zero. Therefore, to converge in a finite number of iterations, it is essential to provide reasonable tolerance (ϵ) of the gap between lower and upper bound. The sBB algorithm is terminated when the absolute or relative gap between the lower and upper bound is within user defined tolerances ϵ_a or ϵ_r, respectively. Consequently, the final result obtained on the convergence of the algorithm will be ϵ-global optimal.

3.4 Case Studies

For all case studies, the sBB algorithm has been implemented as follows. The value of the tolerances ϵ_a and ϵ_r are both set to 10^{-4}. The lower and upper bounds are obtained using a sequential approach that requires numerically solving the ODEs. The sensitivity equations are also solved along with the original ODEs to locally optimize P2. Constructing a valid underestimator also requires solving auxiliary ODE system based on Theorem 3.4. The ODE solver `ode45` in MATLAB R2016a is used for numerically solving all the system of ODEs with absolute and relative tolerances set to 10^{-8}. The function `trapz` is used to compute the required integrals.

The lower bound is obtained by simply evaluating the value of the objective at all the corner points if the problem to be solved in P3. If the problem under consideration is P2, then the corresponding relaxed problem R2-L is linear and the lower bound is obtained using CPLEX. The upper bound is computed after every 10 iterations using `fmincon` with default optimization tolerances. The algorithm used by `fmincon` is set to sequential quadratic programming (SQP). SQP methodology is an iterative procedure that approximates the original nonlinear problem at a point using quadratic programming (QP) subproblem. The QP subproblem is iteratively solved to obtain new points until the optimality and constraint satisfaction tolerances are met.

Interval calculations are needed to estimate θ parameters to construct edge-concave underestimators. The calculations are explicitly performed for simple functions. Otherwise, the open source C++ library C-XSC [108] is used to compute interval bounds of a function. Note that alternate modules can be used to perform various tasks. For instance, instead of `fmincon`, derivative-free optimization approaches [150, 151] can also be employed for locally optimizing P2. Similarly, interval arithmetic package INTLAB [152] may be used instead of C-XSC. Nevertheless, the purpose of the chapter is to demonstrate the utility of edge-concave underestimator to compute tight lower bounds.

The performance of the algorithm is examined on a variety of problems from literature. All the case studies are solved on a 64-bit Intel Xeon E5-2670 2.5 GHz processor running Linux. The first case study is a modified version of a problem reported in Singer [126], while the second case study is an optimal control problem taken from Papamichail and Adjiman [123]. The third example is a parameter estimation problem of a first-order irreversible series of reactions reported by Tjoa and Biegler [153]. The fourth case study is taken from Singer [126], while the fifth case study is a modified problem from Singer [126] to include nonlinear constraints.

3.4.1 Case Study 3.1

The optimization problem is given as follows:

$$
\begin{aligned}
\min_{x} \quad & \int_0^2 [4y_1^2 - 2.1y_1^4 - \frac{y_1^6}{3} + y_1 y_2 - 4y_2^2 - 4y_2^4] dt \\
\textit{s.t.} \quad & \dot{y}_1 = y_1 + 5y_2 + 0.1x_2, \quad \forall t \in (0, 2] \\
& \dot{y}_2 = -5y_1 - 2y_2 - 0.3x_2, \quad \forall t \in (0, 2] \\
& y_1(x, 0) = \frac{x_1}{4} \\
& y_2(x, 0) = 0 \\
& x_1 \in [0, 5], \;\; x_2 \in [-7, 5.5]
\end{aligned}
\tag{3.30}
$$

Let $\psi(y(x,t)) = 4y_1^2 - 2.1y_1^4 - \frac{y_1^6}{3} + y_1 y_2 - 4y_2^2 - 4y_2^4$. The corresponding edge-concave underestimator is:

$$
L_\Psi = \int_0^2 [\psi(y(x,t)) - \sum_{i=1}^{n} \theta_i^\psi (x_i - x_i^M)^2] dt \tag{3.31}
$$

where $y(x, t)$ is obtained by solving the ODEs. For illustration, the objective function along with the corresponding edge-concave underestimator is shown in Figure 3.3. It can be observed that the proposed underestimator provides a tight relaxation of the original objective function. The values of the parameters Θ_1^Ψ and Θ_2^Ψ are 0.072 and 0.0243, respectively. This indicates that the original problem is not edge-concave.

The edge-concave underestimator is compared with three convex relaxations. The first one is the αBB underestimator:

$$
L_\Psi^{\alpha BB}(x) = \int_0^2 [\psi(y(x,t)) + \sum_{i=1}^{n} \alpha_i^\psi (x_i - x_i^L)(x_i - x_i^U)] dt \tag{3.32}
$$

where α_i^ψ are positive parameters that ensure the convexity of $L_\Psi^\alpha(x), \forall x \in [x^L, x^U]$. While writing the αBB underestimator in Eq. (3.32), the result is used that if the integrand is convex for each t, then the integral is also convex [126]. The parameters α_i^ψ are calculated using Gerschgorin method as follows:

$$\alpha_i^{\psi} = \max\left\{0, -\frac{1}{2}\left(\left[\frac{\partial^2\psi}{\partial x_i^2}\right]^L - \sum_{i\neq j}\max\left\{\left|\left[\frac{\partial^2\psi}{\partial x_i \partial x_j}\right]^L\right|, \left|\left[\frac{\partial^2\psi}{\partial x_i \partial x_j}\right]^U\right|\right\}\right)\right\}$$

The values of the parameters $\int_0^2 \alpha_1^{\psi} dt$ and $\int_0^2 \alpha_2^{\psi} dt$ at the root node are 2×10^{13} and 1.73×10^{13}, respectively.

The second convex relaxation can be derived by exploiting the special structure of the objective function. The integrand is composed of one convex term ($4y_1^2$), four concave functions ($-2y_1^4$, $-\frac{y_1^6}{3}$, $-4y_2^2$, $-4y_2^4$) and one bilinear term ($y_1 y_2$). Since the sum of convex functions is convex, replacing each of the nonconvex terms by its convex relaxation will yield a convex underestimator. Each concave function is replaced by its convex envelope, while bilinear term is underestimated using McCormick relaxation. Finally, the following underestimator is obtained:

$$L_{\Psi}^{SS}(x) = \int_0^2 [4y_1^2 + l_1(t) + l_2(t) + l_3(t) + l_4(t) + l_5(t)]dt \tag{3.33}$$

where, $l_1(t)$, $l_2(t)$, $l_3(t)$, $l_4(t)$ and $l_5(t)$ denote the convex relaxation of $-2.1y_1^4$, $-\frac{y_1^6}{3}$, y_1y_2, $-4y_2^2$ and $-4y_4^4$, respectively. The expression of the convex relaxations are given as follows:

$$\begin{aligned}
l_1(t) &= -2.1[y_1^L]^4 - 2.1([y_1^U]^2 + [y_1^L]^2)(y_1^U + y_1^L)(y_1 - y_1^L) \\
l_2(t) &= -\frac{[y_1^L]^6}{3} - \frac{([y_1^U]^3 + [y_1^L]^3)([y_1^U]^2 + [y_1^L]^2 + y_1^L y_1^U)(y_1 - y_1^L)}{3} \\
l_3(t) &= \max\{y_1^L y_2 + y_1 y_2^L - y_1^L y_2^L, y_1^U y_2 + y_1 y_2^U - y_1^U y_2^U\} \\
l_4(t) &= 4y_2^L y_2^U - 4y_2(y_2^U + y_2^L) \\
l_5(t) &= -4[y_2^L]^4 - 4([y_2^U]^2 + [y_2^L]^2)(y_2^U + y_2^L)(y_2 - y_2^L)
\end{aligned}$$

The third underestimator based on interval arithmetic and Lemma 3.1 can also be derived:

$$\begin{gathered}
L_{INT} = \int_0^2 \inf\{4[y_1^L, y_1^U]^2 - 2.1[y_1^L, y_1^U]^4 - \frac{[y_1^L, y_1^U]^6}{3} + \\
[y_1^L, y_1^U][y_2^L, y_2^U] - 4[y_2^L, y_2^U]^2 - 4[y_2^L, y_2^U]^4\}dt
\end{gathered} \tag{3.34}$$

Table 3.1: Summary of the results obtained by different underestimators for case study 3.1.

Underestimators	LB_{root}	N_T	F_L	F_U
Edge-concave	-11.615	21	24	10
αBB	-8.017×10^{14}	10,000*	3.26×10^5	2132
Special Structure	-4.53×10^{21}	10,000*	2×10^4	2159
Interval	-4.53×10^{21}	10,000*	-	2138

* An optimal solution was not found.

The results of different underestimators are shown in Table 3.1. The list of underestimators used for relaxing the problem is provided in column 1. The lower bounds obtained by different underestimators at the root node (denoted as LB_{root}) are presented are listed in Column 2 and the total number of nodes (N_T) required to converge to ϵ-global optimality are shown in column 3. The total number of function evaluations required to obtain lower (F_L) and upper bound (F_U) are shown in column 4 and column 5, respectively. While computing lower bound for edge-concave underestimator requires solving original ODEs, for special structure and αBB based underestimators, both the original ODEs and first-order sensitivity equations need to be solved. The lower bound for interval based underestimator is constant and, therefore, there is no need to solve ODEs to compute the lower bound. Hence, function evaluations by αBB and special structure based underestimator are more expensive than those of edge-concave underestimator.

The global minima of the problem is -10.209. The local solver `fmincon` is able to find it at the root node as an upper bound. However, different underestimators required different number of nodes to tighten the lower bound. The edge-concave underestimator is found to be the tightest amongst all other relaxation schemes. This results in finding the global minima using only 21 nodes, while other methods could not converge to the global solution within 10,000 nodes. The best lower bounds obtained after exploring 10,000 nodes by αBB, special structure and interval based methods are -36912, -4.76×10^{11}, and -4.85×10^{11}, respectively.

3.4.2 Case Study 3.2

The second example is an optimal control problem from Papamichail and Adjiman [123]:

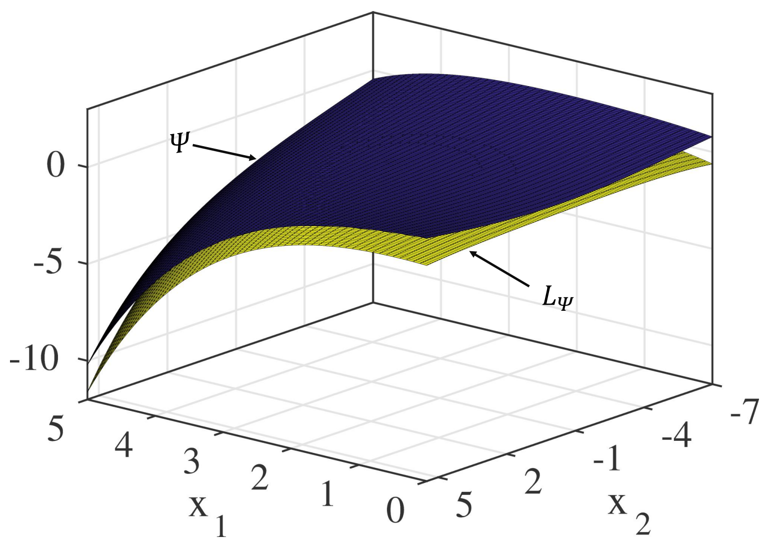


Figure 3.3: The objective function of case study 3.1 is shown along with the corresponding edge-concave underestimator.

$$
\begin{aligned}
\min_{x} \quad & -y(1)^2 \\
\textit{s.t.} \quad & \dot{y} = -y^2 + x, \quad \forall t \in (0,1] \\
& y(x,0) = 9 \\
& x \in [-5,5]
\end{aligned}
\tag{3.35}
$$

The global minima of the problem is -8.2326 and corresponds to $x = -5$. The upper bound obtained by `fmincon` at the root node corresponds to the global minima. Table 3.2 shows the results given by different underestimators. For this example, edge-concave underestimator is tighter than αBB at the root node. However, the nodes required for both the methods are the same because as the bounds are updated, αBB becomes tighter. Although αBB needs the same number of nodes to converge, the number of function calls are almost four times more than that those needed by edge-concave.

As mentioned before, the auxiliary system of ODEs solved to compute θ also yields the lower and upper bounds on the state variables. This can be used to inexpensively compute valid lower bounds by interval method as shown in Eq. (3.26). Furthermore, it would be advantageous to use a hybrid approach given in Eq. (3.27) to choose the tighter lower bound amongst the lower bounds given by ECU and interval-based method. When the hybrid approach is used, the problem is solved to global optimality at the root node. Similarly, the hybrid technique for αBB can also be used. The lower bound obtained by choosing the tighter lower bound amongst αBB and interval-based method is given as follows:

$$L_{HYB-\alpha\text{BB}} = max\{L_{INT}, L_{\alpha\text{BB}}\} \tag{3.36}$$

where $L_{\alpha\text{BB}}$ is the lower bound obtained by generating αBB relaxation. It is observed that when the hybrid of αBB and interval-based relaxation is used, the problem is solved at the root node.

Table 3.2: Summary of the results obtained by different underestimators for case study 3.2.

Underestimators	LB_{root}	N_T	F_L	F_U
Edge-concave	-66.20	9	5	5
αBB	-152.05	9	67	5
Hybrid of Edge-concave and Interval (Eq. (3.27))	-8.2326	1	2	5
Hybrid of αBB and Interval (Eq. (3.36))	-8.2326	1	10	5

3.4.3 Case Study 3.3

The third example is a parameter estimation problem of a first-order irreversible series of reactions [153]:

$$A \xrightarrow{x_1} B \xrightarrow{x_2} C$$

The optimization problem involves estimating kinetic parameters of the reactions (x_1 and x_2) such that the errors between experimental observations and model prediction are minimized. The experimental data are shown in Table 3.3. The problem formulation is given as follows:

$$\begin{aligned}
\min_{x} \quad & \sum_{i=1}^{10}\sum_{j=1}^{2}(y_j(t_i) - y_j^{exp}(t_i))^2 \\
s.t. \quad & \dot{y}_1 = -x_1 y_1, \quad \forall t \in (0,1] \\
& \dot{y}_2 = x_1 y_1 - x_2 y_2, \quad \forall t \in (0,1] \\
& y_1(x,0) = 1 \\
& y_2(x,0) = 0 \\
& x_1 \in [0,10], \;\; x_2 \in [0,10]
\end{aligned} \tag{3.37}$$

The sBB algorithm is applied with different relaxation schemes and the results are summarized in

Table 3.3: Experimental data for case study 3.3.

t_i	y_1^{exp}	y_2^{exp}	t_i	y_1^{exp}	y_2^{exp}
0.1	0.606	0.373	0.6	0.05	0.624
0.2	0.368	0.564	0.7	0.03	0.583
0.3	0.223	0.647	0.8	0.018	0.539
0.4	0.135	0.669	0.9	0.011	0.494
0.5	0.082	0.656	10	0.007	0.451

Table 3.4. All the relaxation schemes converge to the global solution of 1.18×10^{-6} at $x_1 = 5.0035$ and $x_2 = 1$. In this case study, the edge-concave relaxation is found to be weaker compared to αBB at the root node and takes more number of nodes to converge. Although the nodes required by the edge-concave underestimator is more, the number of function calls needed by αBB to compute lower bound is order of magnitude more than that taken by edge-concave relaxation. When the edge-concave and interval-based relaxations are combined as described in Case Study 3.2, the problem is solved at the root node. The same number of nodes are explored when αBB is used along with the interval-based relaxation. However, almost six times more number of function calls (F_L = 23) are needed when computing the lower bound. Furthermore, Papamichail and Adjiman [123] required 35 iterations to converge to global minima with relative tolerance of 10^{-3} using

the same principle but different problem formulation. Their formulation involved introducing new variables corresponding to each state variable at a fixed time, and defining equality constraints incorporating the relationship between new and state variables. This equality constraint is then relaxed by lower and upper bounding inequalities resulting in a convex relaxed problem. Singer and Barton [128] also employed McCormick relaxation for this example and converged to the global minima at the root node.

Table 3.4: Summary of the results for case study 3.3.

Underestimators	LB_{root}	N_T	F_L	F_U
Edge-concave	-1.71×10^5	3713	2093	775
αBB	-5.55×10^4	1331	$23,011$	339
Hybrid of Edge-concave and Interval (Eq. (3.27))	0	1	4	21
Hybrid of αBB and Interval (Eq. (3.36))	0	1	23	21

3.4.4 Case Study 3.4

This involves a one dimensional problem with integral objective function and linear ODE:

$$\begin{aligned} \min_x \quad & \int_o^1 -y^2 dt \\ \textit{s.t.} \quad & \dot{y} = -2y + x, \quad \forall t \in (0,1] \\ & y(x,0) = 1 \\ & x \in [-4,4] \end{aligned} \tag{3.38}$$

The results are reported in Table 3.5. The globally optimum value of decision variable is $x = 4$ and the corresponding objective function value is -2.516. The problem is edge-concave which is identified by computing θ. Therefore, there is no need to locally optimize Eq. (3.38) since the minima of an edge-concave function lie at a vertex.

3.4.5 Case Study 3.5

This case study involves two decision variables with integral objective function, nonlinear constraints and a system of ODEs. A version of this problem was originally reported by Singer [126]

Table 3.5: Summary of the results obtained by different underestimators for case study 3.4.

Underestimators	LB_{root}	N_T	F_L	F_U
Edge-concave	-2.516	1	2	0
αBB	-2.516	1	6	5
Hybrid of Edge-concave and Interval (Eq. (3.27))	-2.516	1	2	0
Hybrid of αBB and Interval (Eq. (3.36))	-2.516	1	6	5

but did not involve nonlinear constraints ($x_1x_2 \leq 4$). The optimization formulation is as follows:

$$\begin{aligned} \min_{x} \quad & \int_0^1 [cos(y_1)sin(y_2) + \frac{y_1}{y_2^2+1}]dt \\ s.t. \quad & x_1x_2 \leq 4 \\ & \dot{y}_1 = ty_1 + x_1, \quad \forall t \in (0,1] \\ & \dot{y}_2 = y_1 - y_2 + p_1 - p_2, \quad \forall t \in (0,1] \\ & y_1(x,0) = 0, \;\; y_2(x,0) = 0 \\ & x_1 \in [-5,6], \;\; x_2 \in [-5,6] \end{aligned} \tag{3.39}$$

This problem is of the form P2 and, therefore, the lower bound can not be determined by simply evaluating the objective function at the vertices. The lower bound is computed by solving the linear relaxed problem R2-L. For the objective function, the values of the desired parameters $\int_0^1 \theta_1^{\psi} dt$ and $\int_0^1 \theta_2^{\psi} dt$ at the root node are 1.01×10^3 and 265.68, respectively. The values of the parameters $\theta_i^{g_{aj}}$ is 0 which indicates that the bilinear term in the constraint is edge-concave. The values of the parameters required to relax the objective function using αBB underestimators, $\int_0^1 \alpha_1^{\psi} dt$ and $\int_0^1 \alpha_2^{\psi} dt$ at the root node are 1.31×10^3 and 323.84, respectively. The value of α for bilinear term is 0.5. This results in a nonlinear convex problem which is solved using `fmincon`. The sBB algorithm converges to yield a global solution of -0.94966 at $x_1 = -1.7921$ and $x_2 = -2.232$. The results for the case study are provided in Table 3.6. The edge-concave underestimator is tighter than αBB at the root node but αBB converges in slightly less number of nodes. However, the number of function calls (solving ODEs) required by αBB is an order of magnitude more than those needed by ECU based method.

Table 3.6: Summary of the results obtained by different underestimators for case study 3.5.

Underestimators	LB_{root}	N_T	F_L	F_U
Edge-concave	-3.87×10^4	291	199	105
αBB	-4.40×10^4	289	3013	117

3.5 Conclusions

In this chapter, a branch-and-bound-based deterministic global optimization algorithm is presented to globally solve black-box problems with embedded ordinary differential equations and the objective function and the constraints comprises of the algebraic and integral terms. Instead of gradient-based approach, a data-driven approach is considered. To generate lower bounds, an edge-concave underestimator is constructed for algebraic and integral terms. It requires computing the upper bound on the diagonal elements of the Hessian of both algebraic and integral terms. Obtaining the underestimator is straightforward for algebraic terms involving only decision variables. Since the Hessian of an algebraic term involves decision variables explicitly, interval method is directly applied to obtain the upper bound. However, it is challenging to obtain the upper bound on diagonal elements of the Hessian for the algebraic and integral terms involving state variables. The difficulty is overcome by defining differential inequalities and solving auxiliary set of ODEs. The algorithm is applied to several case studies for ϵ-global optimality. The edge-concave relaxation offers several advantages over other relaxations. Firstly, for small to medium scale problems, lower bound can be computed in small number of evaluations. On the other hand, αBB relaxation relies on external nonlinear solvers to compute lower bound and the solver may need a lot of evaluations if the initial point is not too close. Secondly, computing parameters to construct edge-concave relaxations require upper bounds on only the diagonal elements of the Hessian, while generating αBB relaxation need bounds on all the Hessian elements. Thirdly, unlike McCormick relaxation, edge-concave relaxation is smooth, which is advantageous when computing lower bounds. It is also illustrated through the case studies that using interval method based relaxation along with the edge-concave is often beneficial in reducing the required number of nodes. Finally, it is not trivial to determine *a-priori* which underestimator will provide tightest relaxation. This difficulty is exacerbated further due to inability to express the objective function and constraints explicitly in terms of decision variables. Nevertheless, edge-concave relaxation offers advantages over other approaches and presents an alternative approach for global optimization of nonconvex problems with embedded system of ODEs.

4. OPTIMIZATION OF BOX-CONSTRAINED BLACK-BOX PROBLEMS: UNIPOPT FRAMEWORK

4.1 Introduction

In the previous chapters, global optimization of grey-box problems with embedded ODEs was considered for which the bounds on the diagonal elements of the Hessian can be estimated. However, due to several reasons, computing these bounds may be tedious or computing finite may not be even possible. Furthermore, the underlying model may be black-box for which only simulation data are available. In this chapter, the focus is now turned on solving general black-box problems for which first and second-order derivatives information are not available and only simulation can be performed. The form of the optimization problem addressed in this chapter is the same as that in Chapter 2 and it is represented as following:

$$\min_{x \in S} \quad f(x) \tag{P1}$$

where $x \in S = \{x^L \leq x \leq x^U\} \subseteq \mathbb{R}^n$, and $f(x) : S \mapsto \mathbb{R}$ is a continuous black-box function whose values are obtained as outputs of deterministic simulation, evaluation of legacy codes or experimentation.

The current state-of-the-art solvers outlined in Chapter 1 can solve only relatively small dimensional problems. Besides, solving nonconvex black-box problems to global optimality is a challenge. In the absence of Jacobian and Hessian information, global blackbox optimization methods rely on exploring the global space to improve the chances of finding the global minima. Although numerical results indicate that they are successful in finding the global minima, there is no guarantee of convergence to even local minima. In contrast to the global methods, there are local methods (see e.g. [54]) that guarantee convergence to a local minima but lack strategies to explore the global space.

To this end, UNIPOPT (UNIvariate Projection-based OPTimization) is presented in this chapter, that explores the global space and is also guaranteed to converge to a local minima when certain conditions are satisfied. UNIPOPT is a BBO framework based on projection of samples onto a univariate space defined by a linear summation of the decision variables. The projection results in a multivalued function also known as point-to-set map or set-valued map or multifunction [154] since there are multiple function values corresponding to the variable. It is observed that a function exists on this map such that its minima is also the minima of the original n-dimensional problem. This function is called the *lower envelope*. The points on this lower envelope are identified by applying a combination of an approximation of the sensitivity theorem (prediction step) and optimizing a subproblem using a trust-region method (correction step). Once the lower envelope is identified, it is then minimized (strictly speaking, the analytical form of the lower envelope is not identified but rather, it is evaluated at discrete points). Specifically, in this chapter, the following is described:

(i) a univariate function whose minima is the same as the minima of $f(x)$ and proposing an algorithm to evaluate this function at discrete points,

(ii) the UNIPOPT framework that is based on evaluating the univariate function while exploring any arbitrary n-dimensional space,

(iii) EPIC (Envelope PredIctor and Corrector) for obtaining the samples on the lower envelope,

(iv) key theoretical results related to the bounds on the solution given by prediction step is derived and the convergence of the algorithm is proven, and

(v) a comparison of the performance of UNIPOPT with BOBYQA, ORBIT, SNOBFIT, IMFIL on 161 convex nonsmooth and 232 nonconvex smooth problems.

The remaining chapter is structured as follows. Section 4.2 presents the main idea of projection. The optimization algorithm is presented in Section 4.3 and the convergence results are provided in Section 4.4. The numerical results and conclusions are given in Section 4.5 and Section 4.6 respectively.

4.2 Projection on Univariate Space

Before presenting the main idea of projection, the notations that are used in the chapter are briefly discussed. A subscript is used to represent an element of sequence of vectors (s_k) when a vector $s \in \mathbb{R}^n$ is given. The i^{th} component of a vector s is represented by (s_i). The subscript k is used as the subscript to represent sequence of vector to avoid ambiguity in representing component of a vector and sequence of vectors. The superscript is used (s^j) to denote the j^{th} element of a finite set of vectors. The norm $||.||$ is the L^1-norm unless otherwise specified. Next, the definitions are provided that are relevant for the discussions presented in this section.

Definition 4.1. ***Point-to-set Map****: A point-to-set map (also known as multivalued function, multi-function, set-valued map, multi-valued map, set-valued function) is a map (Ψ) from a set X into a set Y that associates a subset of Y with each element of X.*

Definition 4.2. ***Lower Envelope****: Lower envelope is a function obtained by associating the minimum of the elements of set Y with each element of X.*

An auxiliary variable added to P1 is equivalent to projecting the original function in the τ space that results in the following problem:

$$\begin{aligned} \min_{x,\tau} \quad & f(x) \\ \textit{s.t.} \quad & \mathbf{1}^T x = \tau \\ & x \in [x^L, x^U], \quad \tau \in [\tau^L, \tau^U] \end{aligned} \tag{P4}$$

where $\tau^L = \mathbf{1}^T x^L$ and $\tau^L = \mathbf{1}^T x^U$.

Remark 4.1. *Adding an auxiliary variable that is linearly dependent on the decision variables does not modify the optimization problem and therefore, the problems P4 and P1 are equivalent problems, i.e., they have the same feasible, local and global optima sets.*

Without loss of generality, the above problem can be rewritten by scaling τ as follows:

$$\begin{aligned} \min_{x,t} \quad & f(x) \\ \text{s.t.} \quad & \mathbf{1}^T x = \tau^L + t(\tau^U - \tau^L) \\ & x \in [x^L, x^U], \quad t \in [0,1] \end{aligned} \tag{P4$'$}$$

While there are many choices of defining the auxiliary variable, the one shown above is a simple choice. Note that the objective function is independent of the auxiliary variable (t) and it is just a mathematical artifact. On writing the KKT conditions for P4$'$, the Lagrange multiplier corresponding to the equality constraint becomes zero. Therefore, the KKT conditions for P4$'$ and P1 are the same, which implies that they have the same local optima. Furthermore, P4$'$ can be rewritten as:

$$\begin{aligned} \min_{t} \quad & G(t) \\ \text{s.t.} \quad & G(t) = \min_{x} \quad f(x) \\ & \qquad \text{s.t.} \quad \mathbf{1}^T x = \tau^L + t(\tau^U - \tau^L) \\ & \qquad \qquad x \in [x^L, x^U], \quad t \in [0,1] \end{aligned} \tag{P4$''$}$$

where, $G(t)$ is the lower envelope of the point-to-set map from t to f. Expressing problem as shown in P4$''$ enables us to decompose the problem into two steps. The lower problem is equivalent to identifying the samples on the lower envelope. Once the lower envelope is known, it can then be minimized. Solving the lower level problem includes optimizing a linearly constrained problem for different fixed values of t, such as t^p, to obtain $G(t^p)$, where $t^p \in \mathbb{Z} = \{t^1, \dots, t^P\}$:

$$\begin{aligned} G(t^p) = \min_{x} \quad & f(x) \\ \text{s.t.} \quad & g_{L,i}(x) := x_i^L - x_i \leq 0 && \forall i \in \{1, \dots, n\} \\ & g_{U,i}(x) := x_i - x_i^U \leq 0 && \forall i \in \{1, \dots, n\} \\ & h_1(x, t^p) := \mathbf{1}^T x = \tau^L + t^p(\tau^U - \tau^L) \end{aligned} \tag{P5}$$

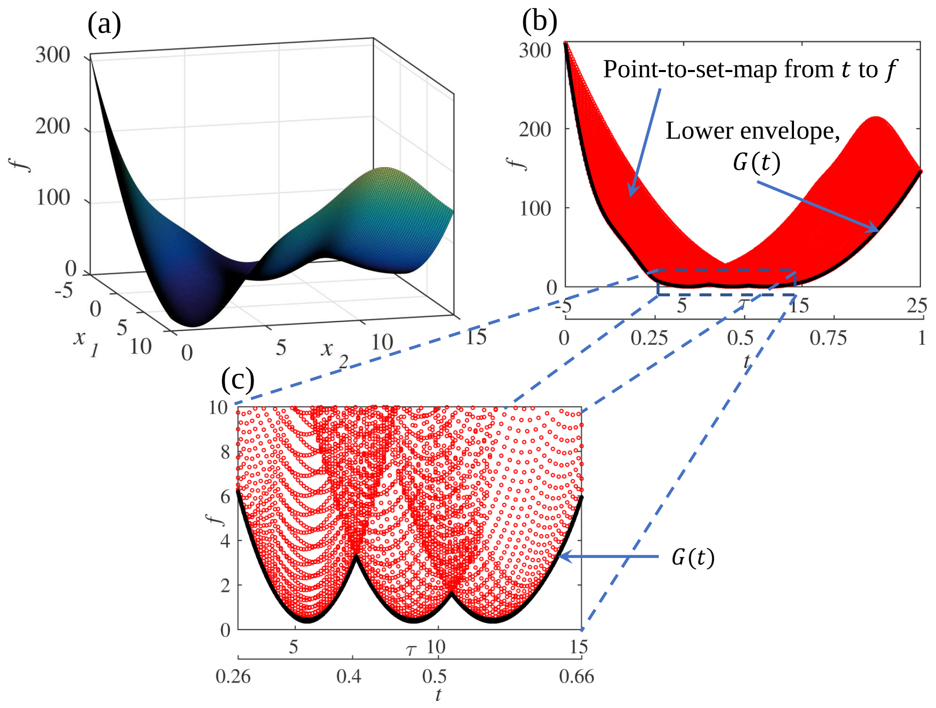


Figure 4.1: Branin function and its projected samples: (a) Branin function, (b) Point-to-set map and the lower envelope of the Branin function, and (c) Global minima of Branin function located on the lower envelope.

Let us denote $\bar{x}^p$ to be the solution of P5 at $t = t^p$. This $(G(t))$ is the optimal value function that also arises in developing algorithms for decomposable mathematical programming problems [155], interim steps of techniques for solution of canonical constrained optimization problem [156], and chemical equilibrium studies [157]. Solving P5 will yield a point on the lower envelope of the point-to-set map from t to f. Its properties such as continuity, differentiability have been studied previously [158, 159, 160] in a different context than shown here and it is briefly mentioned in Section 4.2.2.

4.2.1 Illustrative Example 4.1

To illustrate the idea of univariate projection, let us consider the Branin function (Figure 4.1(a)) given by:

$$f(x_1, x_2) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})cos(x_1) + 10 \\ x_1 \in [-5, 10], \quad x_2 \in [0, 15] \tag{4.1}$$

The Branin function has 3 global minima at $f^* = 0.397887$ and is achieved at x^* =$(-\pi, 12.275)$, $(\pi, 2.275)$, and $(9.424, 2.475)$. The point-to-set map from t to f is shown by red region in Figure 4.1 (b) and the lower envelope is shown in black. The red region in Figure 4.1 (b) contains all the samples shown in Figure 4.1 (a). The lower envelope represents the minimum value of the Branin function at different fixed values of τ. Figure 4.1 (c) provides a closer look at the lower envelope and shows that the three minima in Figure 4.1 (a) are conserved after projection and are located on the lower envelope.

Essentially, for any arbitrary n-dimension problem, there exists a univariate function $(G(t))$ minimizing which is equivalent to solving the original problem P1. In other words, it is possible to map important features of an optimization problem using an auxiliary univariate function. However, determining this map/transformation is a challenging problem and will be addressed in this chapter. Although, P5 is difficult to solve, but a good initial guess can facilitate obtaining the solution quickly. The strategy to provide initial guess and solve P5 is provided in later sections. Another novel aspect of this method is that unlike many local-model based methods which only explore a region locally around a given point, solving P1 using a two step strategy allows for implicitly exploring the n-dimensional space. It may not be always possible to solve a problem in less number of evaluations compared to other local methods but it is expected that the proposed methodology will converge to the global optima in many instances.

The geometrical interpretation of the transformation here is as follows. The linear equation $(\mathbf{1}^T x = \tau^L + t^p(\tau^U - \tau^L))$ represents an n-dimensional hyperplane for a fixed value t^p. All the points in n-dimensional space projected to 1-dimension will result in a point-to set map (Ψ) from set t to function value set, f where for each input, at least one output exists. Correspondingly, there will be numerous points where the plane intersects the function. But a one-to-one or many-to-one map i.e. a single function value is desired corresponding to t^p and the minimum of these function values is taken. Similarly, if the minima of the points on the plane are taken, there will exist a plane corresponding to the optimal value t^* which intersects the function at the optimum x^*.

4.2.2 Properties of $G(t)$

It is observed that P5 is a parametric problem in t, and therefore, the results about convexity, concavity and differentiability from sensitivity literature can be utilized to explore the properties of $G(t)$. These properties are critical to derive the conditions that will influence the convergence of the algorithm.

Property 4.1. *[154] If f(x) is continuous, then G(t) is continuous.*

Property 4.1 implies that a lower envelope will always exist for point-to-set map from t into f when f is continuous in the original space.

Property 4.2. *[161] If f(x) is convex (concave), then G(t) is convex (concave).*

Property 4.3. *(Corollary 3.4.4 in Fiacco [162]) Assume that KKT conditions, second order sufficient condition (SOSC), LICQ and strict complementarity slackness (SCS) hold true at $\bar{x}^p$ with associated Lagrangian multipliers $\bar{\nu}$ and $\bar{\lambda}$, then in the neighborhood of t^p, the optimal function G(t) is twice continuously differentiable.*

Note that even if the original function is differentiable, lower envelope of the point-to-set map from t to f can be non-differentiable. This also explains the nonsmooth behavior observed in Figure 4.1(c) for Branin function.

Property 4.4. *$G(t^L) = f(x^L)$, and $G(t^U) = f(x^U)$.*

4.3 UNIPOPT Framework

Now that a method to obtain G at discrete values of t is provided, the following is further proposed: 1) sampling scheme in the univariate (t) space, and 2) optimization of the lower envelope. Therefore, an algorithm is needed to obtain the minima of the original function using the route of projection onto a univariate space. In this section, the outline of UNIPOPT is provided that integrates the sampling scheme in t-space to evaluate $G(t)$ and then $G(t)$ is optimized such that the final solution of UNIPOPT corresponds to the solution of P1.

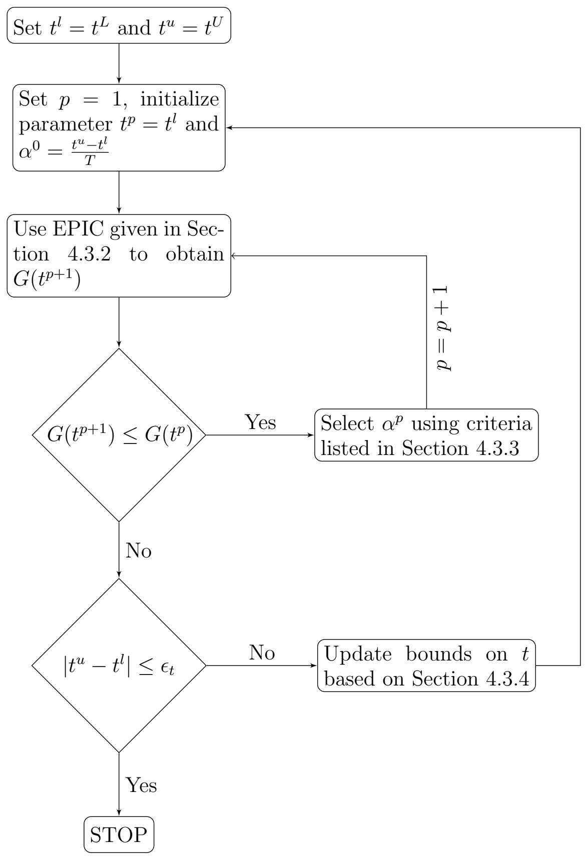

Set $t^l = t^L$ and $t^u = t^U$
Set $p = 1$, initialize parameter $t^p = t^l$ and $\alpha^0 = \frac{t^u - t^l}{T}$
Use EPIC given in Section 4.3.2 to obtain $G(t^{p+1})$
$G(t^{p+1}) \leq G(t^p)$
Yes
Select α^p using criteria listed in Section 4.3.3
$p = p + 1$
No
$|t^u - t^l| \leq \epsilon_t$
No
Update bounds on t based on Section 4.3.4
Yes
STOP

Figure 4.2: Schematic of the UNIPOPT Algorithm Framework.

4.3.1 Outline and Implementation of the UNIPOPT Algorithm

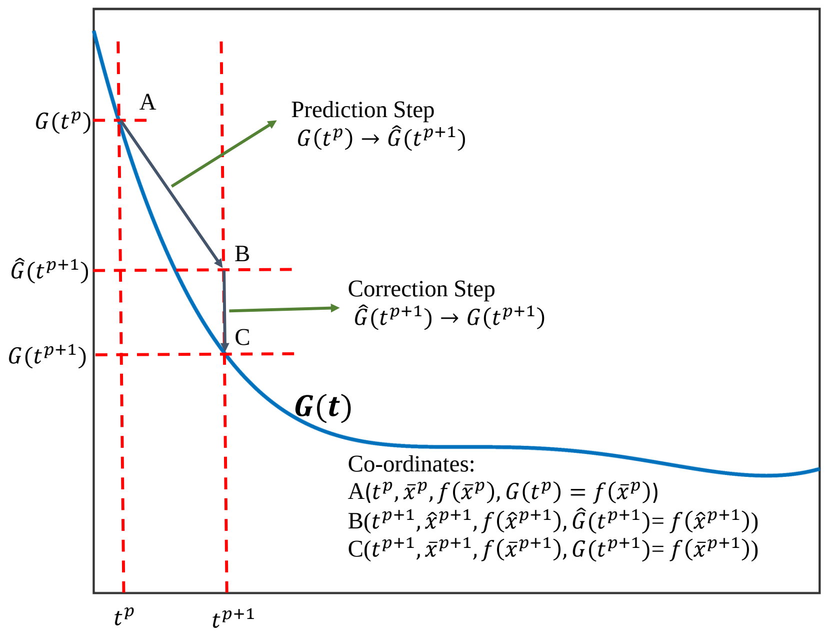
The overall schematic of UNIPOPT is shown in Figure 4.2. The basic idea of the algorithm is to iteratively sample in the t-space, construct surrogate model and narrow down the bounds around the minima. The outline of the first iteration of UNIPOPT is given. The first iteration starts with setting t^l (first sample in t-space) to t^L and a positive starting step length, α^0 ($\alpha^0 = \frac{t^U - t^L}{T}$) is selected by dividing the t-space into T parts. For $t = t^L$, based on Property 4.4, the optimal solution of P5 is ($\bar{x}^1 = x^L$). This will allow us to use EPIC given in Section 4.3.2.2 to obtain the optimum solution at $t = t^2$. Assume that G has been evaluated at t^p and equivalently, the optima of P5 ($\bar{x}^p$) is known. Now, the next sample (t^{p+1}) at which G needs to be evaluated has to be determined. The details of the criteria to select the step length α^p to decide the parameter value

t^{p+1} ($= \alpha^p + t^p$) is given in Section 4.3.3.

The next step involves comparing the lower envelope value at t^{p+1} with that at the previous point (t^p). If $G(t^{p+1})$ is less than $G(t^p)$, p is increased by 1 and the next t is selected. The objective function corresponding to the parameter value t^{p+1} being more than that at the previous iteration implies that $G(t)$ has started to increase and that optima (atleast local) lies within [t^l,t^p]. A univariate surrogate model is then constructed using samples in the interpolation set, $\mathbb{Z}$ and the corresponding values on the lower envelope. After developing the surrogate model, it is optimized and the bounds $[t^l, t^u]$ are updated. The samples that already exist in the bounds are reused in the next iteration for building surrogate models. The details of building and optimizing the univariate surrogate models, and the bounds updating strategy is given in Section 4.3.4. This procedure is repeated until the size of the bounds on t is smaller than a pre-specified tolerance.

Note that the surrogate model of G is developed and optimized only after G has been explored to a certain extent. Doing this increases the chances of locating the global minima. Ideally, to further improve the chances of obtaining global minima, it would be desirable to move to the next outer iteration after the space from t^l to t^u has been covered but that would lead to even more function calls.

While there are a number of alternatives to address different issues and achieve some effect, each decision that was finally taken related to algorithmic steps were based on computational experiments. Of course, the alternatives tested were by no means exhaustive. The algorithm comprising of combination of different alternatives was applied to a subset of problems and the alternatives that did not result in favorable results were weeded out. For instance, an alternative is to use fixed step length, α but it was observed that using variable step length, as used in the algorithm is more effective. Another choice is scaling the x variables. It was observed from computational experiments that scaling x helps in reducing number of evaluations. This analysis is not shown here in order to limit the length of the chapter.



$G(t^p)$
A
Prediction Step
$G(t^p) \to \hat{G}(t^{p+1})$
$\hat{G}(t^{p+1})$
B
Correction Step
$\hat{G}(t^{p+1}) \to G(t^{p+1})$
$G(t^{p+1})$
C
$G(t)$
Co-ordinates:
A($t^p, \bar{x}^p, f(\bar{x}^p), G(t^p) = f(\bar{x}^p)$)
B($t^{p+1}, \hat{x}^{p+1}, f(\hat{x}^{p+1}), \hat{G}(t^{p+1}) = f(\hat{x}^{p+1})$)
C($t^{p+1}, \bar{x}^{p+1}, f(\bar{x}^{p+1}), G(t^{p+1}) = f(\bar{x}^{p+1})$)
t^p
t^{p+1}

Figure 4.3: Illustration of pointwise evaluation of $G(t)$ from $t = t^p$ to $t = t^{p+1}$. This involves two major steps: (1) prediction from A to B, and (ii) correction from B to C.

4.3.2 Finding the Lower Envelope using Envelope PredIctor and Corrector (EPIC) Method

In this section, an algorithm is described for point evaluation of $G(t)$ at $t = t^{p+1}$ when it is known at $t = t^p$. Solving P5 to obtain $G(t)$ at a given point $t = t^{p+1}$ from an arbitrary point is a challenging problem. However, if a good initial guess is provided, the problem can be solved with relatively ease. Therefore, EPIC is proposed that involves predicting (Section 4.3.2.1) and then correcting (Section 4.3.2.2) the solution of P5. The overall idea of EPIC is illustrated in Figure 4.3. Assume that the minima of P5 ($\bar{x}^p$) is known at t^p, a prediction step is taken to predict the minima of P5 ($\hat{x}^{p+1}$) and obtain $\hat{G}(t^{p+1})$ at t^{p+1}. Furthermore, a correcting step is taken to converge to the minima ($\bar{x}^{p+1}$).

4.3.2.1 Predicting G(t): Obtaining $\hat{G}(t^{p+1})$ from $G(t^p)$

In this section, a method to estimate $G(t^{p+1})$ and $\bar{x}^{p+1}$ is described assuming $G(t^p)$ and $\bar{x}^p$ are known. A well known result by [163] is used that provides a technique to estimate the sensitivity of a local solution of a general nonlinear program to small perturbation in the problem parameters. This result provides an explicit expression of the partial derivatives of the local optima and its corresponding multipliers with respect to the parameters. Let $\bar{y}^p = [\bar{x}^p, \bar{\nu}_L^p, \bar{\nu}_U^p, \bar{\lambda}^p]$ denote vector of minima and the corresponding Lagrangian multipliers of P5 at the parameter value $t = t^p$. $\bar{\nu}_L^p$, $\bar{\nu}_U^p$, and $\bar{\lambda}^p$ are the Lagrange multipliers corresponding to g_L, g_U, and h_1 respectively in P5 at the parameter value $t = t^p$ and the corresponding optimal solution $\bar{x}^p$. The result though valid for general nonlinear constrained parametric problem is stated for P5 in Theorem 4.1.

Theorem 4.1. ***Basic Sensitivity Theorem*** *[163]. Suppose assumptions of Property 4.3 hold true, then (i) $\bar{x}^p$ is a local minimizer and the Lagrange multipliers are unique, (ii) there exists a once continuously differentiable vector function y =$[x(t), \nu_L(t), \nu_U(t), \lambda(t)]$ such that $x(t)$ satisfies the KKT conditions of P5 at t in the neighborhood of t^p:*

$$\frac{dy(t)}{dt} = -M^{-1}N \tag{4.2}$$

where,

$$M = \begin{bmatrix} \nabla^2 L & \nabla g_{L,1} & \dots & \nabla g_{L,n} & \nabla g_{U,1} & \dots & \nabla g_{U,n} & \nabla h_1 \\ -\nu_{L,1}\nabla^T g_{L,1} & -g_{L,1} & & & & & & \\ \vdots & & \ddots & & & & & \\ -\nu_{L,n}\nabla^T g_{L,n} & & & -g_{L,n} & & & & \\ -\nu_{U,1}\nabla^T g_{U,1} & & & & -g_{U,1} & & & \\ \vdots & & & & & \ddots & & \\ -\nu_{U,n}\nabla^T g_{U,n} & & & & & & -g_{U,n} & 0 \\ \nabla^T h_1 & \dots & & 0 & & & & \end{bmatrix},$$

$$N = \left[\nabla_{tx}^2 L, -\nu_{L,1}\nabla_t^T g_{L,1}, \dots, -\nu_{L,n}\nabla_t^T g_{L,n}, -\nu_{U,1}\nabla_t^T g_{U,1}, \dots, -\nu_{U,n}\nabla_t^T g_{U,n}, \nabla_t^T h_1\right]^T,$$

$$g_{L,1} := x_1^L - x_1 \le 0,\ \ g_{L,n} := x_n^L - x_n \le 0,\ \ g_{U,1} := x_1 - x_1^U \le 0,\ \ g_{U,n} := x_n - x_n^U \le 0,$$

$$h_1 := \mathbf{1}^T x = \tau^L + t(\tau^U - \tau^L)$$

and

$$L(x,t) = f(x) + \sum_{i=1}^{n} \nu_{L,i} g_{L,i}(x) + \sum_{i=1}^{n} \nu_{U,i} g_{U,i}(x) + \lambda h_1(x,t)$$

A first order Taylor series approximation can be used to estimate the difference in the optimal solution of P5 at a point t^{p+1} and t^p by:

$$\Delta y^p = -(M(\bar{y}^p))^{-1}(N(\bar{y}^p))(t^{p+1} - t^p) \tag{4.3}$$

where $\Delta y^p = [\Delta x^p, \Delta\nu_L^p, \Delta\nu_U^p, \Delta\lambda^p]$, $\Delta x^p = x(t^{p+1}) - \bar{x}^p$, $\Delta\nu_L^p = \nu_L(t^{p+1}) - \bar{\nu}_L^p$, $\Delta\nu_U^p = \nu_U(t^{p+1}) - \bar{\nu}_U^p$, and $\Delta\lambda^p = \lambda(t^{p+1}) - \bar{\lambda}^p$. The above result has been used to obtain explicit solution in optimal control problems [164, 165, 166], where the objective function is typically quadratic and the constraints are linear with parameters appearing on the right hand side. Other forms of the above result with modified assumptions have been reported as well [167, 168, 169, 170, 171].

Theorem 4.1 provides a transformation to convert a point-to-set map (Ψ) to construct a linear approximation of the lower envelope. However, the sensitivity theorem requires Hessian of the original function. In black-box problems, this information is generally absent. If the problem being addressed were a classical nonlinear convex program and the steps were sufficiently small, it would have been possible to obtain the exact optima of P5 at t^{p+1}. In this work, relatively larger steps will be taken and the Hessian of the Lagrangian function is not known but rather approximated by the Hessian of a surrogate model.

Let $\hat{y}$ =$[\hat{x}(t), \hat{\nu}_L(t), \hat{\nu}_U(t), \hat{\lambda}(t)]$ be obtained by using the Hessian of the surrogate model in Sensitivity Theorem as shown below:

$$\frac{d\hat{y}(t)}{dt} = -\hat{M}^{-1}\hat{N} \tag{4.4}$$

where,

$$\hat{M} = \begin{bmatrix} \nabla^2 L^r & \nabla g_{L,1} & \dots & \nabla g_{L,n} & \nabla g_{U,1} & \dots & \nabla g_{U,n} & \nabla h_1 \\ -\hat{\nu}_{L,1}\nabla^T g_{L,1} & -g_{L,1} & & & & & & \\ \vdots & & \ddots & & & & & \\ -\hat{\nu}_{L,n}\nabla^T g_{L,n} & & & -g_{L,n} & & & & \\ -\hat{\nu}_{U,1}\nabla^T g_{U,1} & & & & -g_{U,1} & & & \\ \vdots & & & & & \ddots & & \\ -\hat{\nu}_{U,n}\nabla^T g_{U,n} & & & & & & -g_{U,n} & 0 \\ \nabla^T h_1 & \dots & & 0 & & & & \end{bmatrix}$$

$$\hat{N} = \left[\nabla^2_{tx} L^r, -\hat{\nu}_{L,1}\nabla^T_t g_{L,1}, \dots, -\hat{\nu}_{L,n}\nabla^T_t g_{L,n}, -\hat{\nu}_{U,1}\nabla^T_t g_{U,1}, \dots, -\hat{\nu}_{U,n}\nabla^T_t g_{U,n}, \nabla^T_t h_1\right]^T$$

and,

$$L^r(x,t) = f^r(x) + \sum_{i=1}^{n} \hat{\nu}_{L,i} g_{L,i}(x) + \sum_{i=1}^{n} \hat{\nu}_{U,i} g_{U,i}(x) + \hat{\lambda} h_1(x,t) \tag{4.5}$$

where $f^r(x)$ is the surrogate model of $f(x)$. The details of the procedure to develop surrogate model is given in Section 4.3.2.4. Furthermore, a first order Taylor series approximation of Eq. (4.4) can be used to estimate the difference in the optimal solution at a point t^{p+1} and t^p by:

$$\Delta\hat{y}^p = -(\hat{M}(\bar{y}^p))^{-1}(\hat{N}(\bar{y}^p))(t^{p+1} - t^p) \tag{4.6}$$

where $\Delta\hat{y}^p = [\Delta\hat{x}^p, \Delta\hat{\nu}^p_L, \Delta\hat{\nu}^p_U, \Delta\hat{\lambda}^p], \Delta\hat{x}^p = \hat{x}(t^{p+1}) - \bar{x}^p$, $\Delta\hat{\nu}^p_L = \hat{\nu}_L(t^{p+1}) - \bar{\nu}^p_L$, $\Delta\hat{\nu}^p_U = \hat{\nu}_U(t^{p+1}) - \bar{\nu}^p_U$ and $\Delta\hat{\lambda}^p = \hat{\lambda}(t^{p+1}) - \bar{\lambda}^p$. For simplicity in notation, $\hat{x}(t^{p+1})$, $\hat{\nu}_L(t^{p+1})$, and $\hat{\nu}_U(t^{p+1})$ are denoted as $\hat{x}^{p+1}$, $\hat{\nu}^{p+1}_L$, and $\hat{\nu}^{p+1}_U$ respectively. Note that since the constraints in P5 are linear and box constraints, the Hessian of the surrogate model of Lagrange function would simply be Hessian of the original function ($\nabla^2 L^r = \nabla^2 f^r$). Furthermore, Eq. (4.6) requires Lagrange multiplier values corresponding to the optima of P5 at t^p. The details of estimating the Lagrange multipliers is given in Section 4.3.2.3.

Next, the difference in the solutions given by Eq. (4.3) and Eq. (4.6) is shown to be bounded. For simplicity in notation, $M(\bar{y}^p)$, $\hat{M}(\bar{y}^p)$, $N(\bar{y}^p)$, and $\hat{N}(\bar{y}^p)$ are denoted as M_p, $\hat{M}_p$, N_p, and $\hat{N}_p$, respectively. The step length is defines as $\alpha = t^{p+1} - t^p$.

Theorem 4.2. *Let $\kappa_{\hat{M}_p}$ and κ_{M_p} denote the condition number of $\hat{M}_p$ and M_p respectively, and $\kappa_H \Delta^p$ denote the bounds corresponding to the fully quadratic model. Δy^p, $\Delta \hat{y}^p$ denote the solutions of Eq.* (4.3) *and Eq.* (4.6) *respectively. For simplicity in notation, let us denote $M(\bar{y}^p)$ and $\hat{M}(\bar{y}^p)$ as M_p and $\hat{M}_p$, respectively. Assume that the surrogate model is fully quadratic and the assumptions of Theorem 4.1 are satisfied at t^p, then the difference between the solution given by Eq.* (4.3) *(Δy^p)and Eq.* (4.6) *($\Delta \hat{y}^p$) is:*

$$||\Delta \hat{y}^p - \Delta y^p|| \leq \frac{\alpha \kappa_{M_p} \kappa_{\hat{M}_p} \kappa_H \Delta^p ||N_p||}{||M_p|| ||\hat{M}_p||} \tag{4.7}$$

Proof. Rewriting first-order sensitivity approximation (Eq. (4.3)) compactly when the Hessian is known, the following is obtained:

$$\Delta y^p = -(M_p)^{-1}(N_p)\alpha \tag{4.8}$$

Taking norm on both sides and using the Cauchy-Schwarz inequality,

$$||\Delta y^p|| \leq \alpha ||{M_p}^{-1}||.||N_p|| \tag{4.9}$$

Using the definition of condition number of M_p as $\kappa_M = ||M_p^{-1}||.||M_p||$ and substituting in Eq. (4.9).

$$||\Delta y^p|| \leq \alpha \frac{\kappa_{M_p} ||N_p||}{||M_p||} \tag{4.10}$$

Rewriting Eq. (4.6) compactly:

$$\hat{M}_p \Delta \hat{y}^p = -\hat{N}_p \alpha \tag{4.11}$$

N_p is a vector with $3n+1$ rows, where all the elements except the last are zero. Therefore, $N_p = \hat{N}_p$

holds true and this relationship allows us to write

$$\hat{M}_p \Delta \hat{y}^p = M_p \Delta y^p \tag{4.12}$$

Subtracting $\hat{M}_p \Delta y^p$ on both sides

$$\hat{M}_p(\Delta \hat{y}^p - \Delta y^p) = (M_p - \hat{M}_p)\Delta y^p \tag{4.13}$$

Taking norm on both sides and using the Cauchy-Schwarz inequality,

$$||\Delta \hat{y}^p - \Delta y^p|| \leq ||\hat{M}_p^{-1}|| ||M_p - \hat{M}_p|| ||\Delta y^p|| \tag{4.14}$$

Utilizing the definition of condition number ($\kappa_{\hat{M}_p} = ||\hat{M}_p^{-1}||_2.||\hat{M}_p||_2$) of $\hat{M}_p$, Eq. (4.10),

$$||\Delta \hat{y}^p - \Delta y^p|| \leq \frac{\kappa_{\hat{M}_p}}{||\hat{M}_p||} ||M_p - \hat{M}_p|| \frac{\alpha \kappa_{M_p} ||N_p||}{||M_p||} \tag{4.15}$$

It can be verified using definitions of M, $\hat{M}$ that the following relation holds,

$$||M_p - \hat{M}_p|| = ||\nabla^2 L - \nabla^2 L^r|| \tag{4.16}$$

where, L and L^r are the Lagrangian function and surrogate model of the Lagrangian function respectively. Using Eq. (4.16) in Eq. (4.15), the following is shown

$$||\Delta \hat{y}^p - \Delta y^p|| \leq \frac{\alpha \kappa_{M_p} ||N_p||}{||M_p||} \frac{\kappa_{\hat{M}_p}}{||\hat{M}_p||} ||\nabla^2 L - \nabla^2 L^r|| \tag{4.17}$$

Assuming the model to be fully quadratic ($||\nabla^2 f - \nabla^2 f^r||_2 \leq \kappa_H \Delta^p$), the above expression can be rewritten more compactly as follows:

$$||\Delta \hat{y}^p - \Delta y^p|| \leq \frac{\alpha \kappa_{M_p} \kappa_{\hat{M}_p} \kappa_H \Delta^p ||N_p||}{||M_p|| ||\hat{M}_p||} \tag{4.18}$$

$\square$

Due to approximations involved, Eq. (4.6) is an approximate solution of P5 at the parameter value t^{p+1}. Nevertheless, the point suggested by the first-order approximation of the sensitivity theorem in Eq. (4.6) serves as an initial guess for a trust-region based optimization algorithm to solve (P5) and obtain the optima.

Remark 4.2. *Although Eq.* (4.6) *is a powerful result in estimating an optima of a parametric problem, there are assumptions involved which may not always be satisfied even if $\bar{x}^p$ is a minima. For instance, when the components of $\bar{x}^p$ are at the bounds, LICQ is not satisfied, and $\hat{M}_p$ becomes singular. Similarly, if the multiplier corresponding to an active constraint is zero, SCS is violated and that leads to one of the rows of $\hat{M}_p$ becoming zero leading to its singularity. In case this happens, the following strategy is opted to obtain an initial guess for solving P5 that satisfies the equality constraint ($\sum_{i=1}^{n} \hat{x}_i^{p+1} = t^{p+1}$):*

$$\hat{x}^{p+1} = \bar{x}^p + \frac{t^{p+1} - t^p}{n} \tag{4.19}$$

While Eq. (4.19) may sometimes lead to arbitrary point, a point needs to be generated for the algorithm to progress. Due to approximations involved in Eq. (4.6), it is possible that the predicted point given by Eq. (4.6) ($\hat{x}^{p+1}$) or Eq. (4.19) is infeasible. Therefore, the suggested optima is made feasible by projecting in the feasible space:

$$\hat{x}^{p+1} = P_{\mathbb{B}}(\hat{x}^{p+1}) \tag{4.20}$$

where $\mathbb{B}$ denotes the set of points in $[x^l, x^u]$.

4.3.2.2 Correcting G(t): Converging to $G(t^{p+1})$ from $\hat{G}(t^{p+1})$

After Eq. (4.6) yields a predicted point, it needs to be corrected or atleast verified if the optimality conditions are satisfied. In other words, $\hat{G}(t^{p+1})$ is already known by Eq. (4.6) that is close to $G(t^{p+1})$ and an algorithm is needed to converge to $G(t^{p+1})$. The goal of finding $G(t^{p+1})$

is equivalent to solving P5 at the parameter value t^{p+1}. The choice of the correcting algorithm and the nature of the problem affects the overall convergence of UNIPOPT. In case the problem is non-convex, it should be globally optimized. On the other hand, if the problem is convex, using a local method will suffice. However, available global solvers for blackbox problems do not guarantee convergence in finite number of evaluations. Therefore, a local method is used as the correcting algorithm. Specifically, a trust-region method is used that is known to converge quickly if a good initial guess is provided. The goal here is to apply a trust-region based n-dimensional search for $G(t^{p+1})$ and $\bar{x}^{p+1}$ starting from $\hat{G}(t^{p+1})$ and $\hat{x}^{p+1}$. While single trust-region based methods [54, 52] can be used, method based on managing two trust-regions [90, 53, 172] is applied. Single trust-region is easy to manage, it has been reported that using two trust-regions (ρ, Δ) is more efficient [90]. The basic approach is to use samples to construct a surrogate model $f^r(x)$ to approximate $f(x)$ and solve the following optimization problem iteratively in a trust-region framework:

$$\begin{aligned} \min_{x} \quad & f^r(x) \\ \textit{s.t.} \quad & \mathbf{1}^T x = \tau^L + t^p(\tau^U - \tau^L) \\ & ||x - x_k|| \leq \Delta_k \end{aligned} \tag{4.21}$$

The optimization problem in Eq. (4.21) is solved to obtain a new candidate point. If the new point reduces the objective function, it becomes the trust-region center for the next iteration and the trust-region size is potentially increased. Conversely, if the new point does not reduce the objective function, it could be due to inaccuracy of the surrogate model or the minima has been obtained. If the current point is indeed a minima, the trust-region sizes will continue to decrease until it is within a pre-specified tolerance and the algorithm will converge. On the other hand, the surrogate model may be inaccurate either due to poor geometry of the samples or the trust-region size is too large for the surrogate model to approximate the original function effectively. If the reason is former, the geometry of the sample set is improved and if it is latter, the trust-region size is decreased.

The trust-region algorithm used in this work is based on two trust-regions (ρ_k, Δ_k) and given

Algorithm 4.1 Correcting Algorithm for Converging to $G(t^{p+1})$ from $\hat{G}(t^{p+1})$

```
1: STEP 0: Let $\mathbb{Y}_k$ denote the interpolation set at iteration $k$, $c^{(j)} \in \mathbb{Y}_k$ denote the $jth$ interpolation point and $\psi_k^r$ be the criticality measure defined using surrogate model. Select initial $2n+1$ interpolation points, initial guess ($x_0$), initial trust region radius ($\rho_0$= $\Delta_0$), other parameters ($\eta_0$, $\eta_1$, $\lambda^{dec}$, $\lambda^{inc}$, $\epsilon_\rho$) and initialize the iteration counter $k = 0$.
2: STEP 1: Construct the surrogate model, solve Eq. (4.21) and obtain $x^r$.
3: STEP 2: Update the trust-regions and interpolation set.
4: $c^f = arg \max_{c^{(j)} \in \mathbb{Y}_k} ||c^{(j)} - x_k||$
5: if ($f(x^r) < f(x_k)$) then, $x_{k+1} = x^r$ and $\mathbb{Y}_{k+1} = \mathbb{Y}_k \backslash c^f \cup x^r$
6: end if
7: if ($||x^r - x_k|| \leq 0.5\rho_k$) then,
8:     if ($|f(x^r) - f^r(x^r)| \leq \frac{1}{8}\psi_k^r \rho_k^2$) then,
9:         if ($\rho_k \leq \epsilon_\rho$) then, STOP!
10:        else $\rho_{k+1} = \max\{\epsilon_\rho, 0.1\rho_k\}$, $\Delta_{k+1} = \max\{0.5\rho_k, \rho_{k+1}\}$.
11:        end if
12:    else $\Delta_{k+1} = \max\{0.1\Delta_k, \rho_k\}$, $\rho_{k+1} = \rho_k$
13:        $r_k = -1$
14:        $c^f = arg \max_{c^{(j)} \in \mathbb{Y}_k} ||c^{(j)} - x_k||$
15:        if ($||c^f - x_k|| \geq 2\Delta_k$) then,
16:            if (max$\{\Delta_k, ||x^r - x_k||\} \leq \rho_k$) then,
17:                if ($\rho_k \leq \epsilon_\rho$) then, STOP.
18:                else $\rho_{k+1} = \max\{\epsilon_\rho, 0.1\rho_k\}$, $\Delta_{k+1} = \max\{0.5\rho_k, \rho_{k+1}\}$.
19:                end if
20:            end if
21:        else Initiate model improvement step to obtain a new point, $\bar{x}^l$ that replaces $c^f$, i.e.,
22:            $\mathbb{Y}_{k+1} = \mathbb{Y}_k \backslash c^f \cup \bar{x}^l$
23:        end if
24:    end if
25: else Calculate $r_k = \frac{f(x^r) - f(x_k)}{f^r(x^r) - f^r(x_k)}$
26:    if ($r_k \geq \eta_1$ ) then, $\Delta_k = \lambda^{inc}\Delta_k$
27:    else if ($\eta_0 \leq r_k < \eta_1$) then, $\Delta_k = \Delta_k$
28:    else if ($r_k < \eta_0$) then, $\Delta_k = \lambda^{dec}\Delta_k$
29:    end if
30:    $\Delta_{k+1} = \Delta_k$
31:    $c^f = arg \max_{c^{(j)} \in \mathbb{Y}_k} ||c^{(j)} - x_k||$
32:    if ($r_k \geq \eta_0$) then,
33:        $\mathbb{Y}_{k+1} = \mathbb{Y}_k \backslash c_k^f \cup x^r$
34:        $\rho_{k+1} = \rho_k$
35:    else
36:        Perform steps stated in lines 15-24.
37:    end if
38: end if
39: Step 3: Increment the iteration number, $k = k + 1$ and go to Step 1.
```

in Algorithm 4.1. The Correcting Algorithm starts by obtaining $2n + 1$ design points. The details of the sampling scheme is given in Section 4.3.2.4. The algorithm also needs an initial guess and is provided by the prediction step (Eq. (4.6)). For simplicity in notation, the initial point is simply represented by x_0. The initial sizes of the trust-regions are also provided such that $\rho_0 = \Delta_0$. The factors by which the trust-region size (Δ) is increased or decreased are defined as λ^{inc} and λ^{dec} respectively. Note that the trust-region size ρ is always decreased by a factor of 10. The values of constants η_0 and η_1 are set and the iteration counter k is initialized to 0.

The trust-region radius ρ_k is a lower bound on Δ_k. While the parameter ρ_k is always decreased, the trust-region radius Δ_k can be decreased, increased or kept constant provided it is lower than ρ_k. This allows the algorithm to take steps exceeding ρ_k. The parameter ρ_k is defined to manage the samples in a smaller trust-region while Δ_k is specified to consider a bigger region where the surrogate model is considered. ρ_k is decreased whenever the objective function has stopped decreasing because of the constraint on Δ_k.

Once the simulations are performed at the design points, the surrogate model is developed. Eq. (4.21) is solved at each iteration to obtain a new point represented by x^r. A check is performed to see if the step length is small compared to the trust-region radius ρ_k. If this is true, it indicates that either x^r is near the optima or the surrogate model is inaccurate. If x^r gives a better objective function, the interpolation point farthest from the current iterate is replaced by x^r. A check on the accuracy of the surrogate model is then performed by comparing the difference between original function value $f(x^r)$ and the surrogate model $f^r(x^r)$ with reference to the criticality measure (ψ_k^r) and the trust-region radius ρ_k. If the difference is small, the model is deemed accurate and convergence test on ρ_k is performed. If ρ_k is less than some pre-specified tolerance, the algorithm converges or the trust-region sizes ρ_k and Δ_k are decreased reflecting that the actual optima is near and the current trust-region ρ_k is restricting the algorithm to decrease the objective function. If the model is not accurate, Δ_k is decreased and an attempt is made to improve the geometry of the samples. It is then checked if the distance of farthest interpolating point from the current iterate is more than $2\Delta_k$. If such a point, c^f exists, model improvement step is initiated that replaces the farthest point. If c^f does not exist, then the interpolation set is well poised and ρ_k is checked if it is less than the maximum of trust-region radius Δ_k and step length. ρ_k is decreased if the convergence criteria is not satisfied.

If the norm distance between the x^r and x_k is more than half of ρ_k, Δ_k is updated by comparing the ratio of actual decrease to the predicted decrease (r_k). If r_k is more than a specific parameter η_0, the new point x^r is included in the interpolation set and the farthest point is excluded. If the ratio r_k is less, this indicates that the model is inaccurate and a model improvement step is initiated to obtain $\bar{x}^l$ that replaces the farthest point. Same steps as mentioned in the previous paragraph are then repeated.

4.3.2.3 Estimating Lagrange Multipliers and Criticality Measure

As mentioned earlier, the point given by Eq. (4.6) serves as an initial guess for the correcting algorithm (Section 4.3.2.2). When Eq. (4.6) is applied from t^{p-1}, it also provides an estimate of the Lagrange multipliers corresponding to the optima of P5 at t^p. However, this estimate of the Lagrange multipliers may not be accurate since Eq. (4.6) does not guarantee the exact optima for a general nonlinear problem. Therefore, once Algorithm 4.1 yields the optimal point of P5 at the parameter value t^p, the corresponding multipliers are estimated by solving a mixed integer linear program (MILP). The MILP is formulated such that if a candidate point x_k is optimal, KKT conditions are satisfied and positive multipliers corresponding to the active constraints are obtained. Binary variables are introduced to ensure that whenever the constraints are active, the corresponding Lagrange multipliers are positive. Solving this problem serves two purposes: 1) The objective function gives the criticality measure, and 2) The Lagrange multipliers are estimated that can be used in Eq. (4.6).

$$\psi_k^r = \min_{\substack{SP_i, SN_i,\\ \nu_{L,i}, \nu_{U,i}, \lambda\\ z_{L,i}, z_{U,i}}} \sum_{i=1}^{n} SP_i + SN_i \tag{4.22a}$$

$$\text{s.t.} \quad \nabla f^r\big|_{x_{k,i}} + \lambda - \nu_{L,i} + \nu_{U,i} = SP_i - SN_i \quad \forall i \in \{1, \ldots, n\} \tag{4.22b}$$

$$z_{L,i} - M^B(x_i^L - x_{k,i}) \geq 1 \qquad \forall i \in \{1, \ldots, n\} \tag{4.22c}$$

$$z_{L,i}(x_i^L - x_{k,i}) \geq 0 \qquad \forall i \in \{1, \ldots, n\} \tag{4.22d}$$

$$\nu_{L,i} + M^B z_{L,i} \geq 0 \qquad \forall i \in \{1, \ldots, n\} \tag{4.22e}$$

$$\nu_{L,i} \leq M^B z_{L,i} \qquad \forall i \in \{1, \ldots, n\} \tag{4.22f}$$

$$\nu_{L,i} + M^B(1 - z_{L,i}) \geq L^S \qquad \forall i \in \{1, \ldots, n\} \tag{4.22g}$$

$$z_{U,i} - M^B(-x_i^U + x_{k,i}) \geq 1 \qquad \forall i \in \{1, \ldots, n\} \tag{4.22h}$$

$$z_{U,i}(-x_i^U + x_{k,i}) \geq 0 \qquad \forall i \in \{1, \ldots, n\} \tag{4.22i}$$

$$\nu_{U,i} + M^B z_{U,i} \geq 0 \qquad \forall i \in \{1, \ldots, n\} \tag{4.22j}$$

$$\nu_{U,i} \leq M^B z_{U,i} \qquad \forall i \in \{1, \ldots, n\} \tag{4.22k}$$

$$\nu_{U,i} + M^B(1 - z_{U,i}) \geq L^S \qquad \forall i \in \{1, \ldots, n\} \tag{4.22l}$$

$$\nu_{L,i}, \nu_{U,i}, SP_i, SN_i \geq 0 \qquad \forall i \in \{1, \ldots, n\} \tag{4.22m}$$

$$z_{L,i}, z_{U,i} \in \{0, 1\} \qquad \forall i \in \{1, \ldots, n\} \tag{4.22n}$$

where ∇f^r is the Jacobian of the surrogate model, M^B and L^S are arbitrarily large and small numbers respectively, while z_L, z_U are binary variables. In the above optimization problem, the slack variables (SP_i, SN_i) are minimized such that at the optima of P5, the Jacobian of the surrogate model of the Lagrangian function defined in Eq. (4.5) is equal to zero. The Lagrange multipliers are desired to be positive when the inequality constraints are active so that SCS is not violated. To do this, binary variables (z_L, z_U) are introduced. Lets assume that i^{th} component of x_k is at its lower bound ($x_{k,i} = x_i^L$). Eq. (4.22c) ensures that $z_{L,i} = 1$, Eq. (4.22d) and Eq. (4.22e) become trivial, Eq. (4.22f) sets the upper bound on the Lagrange multiplier ($\nu_{L,i}$) to a large number and Eq. (4.22g) sets the lower bound of $\nu_{L,i}$ as a non-zero small positive number. Thus, the Lagrange multipliers corresponding to the active constraints are positive. On the other hand, Eq. (4.22h) and Eq. (4.22l) becomes trivial, Eq. (4.22i) sets $z_{U,i} = 0$, and Eq. (4.22j) and Eq. (4.22k) ensures that the corresponding Lagrange multiplier is zero.

ψ_k^r defined above can also be used as a criticality measure. Derivative-based criticality measure has been been extensively used in the nonlinear programming literature [173]. The primary idea of the measure is to estimate if further decrease in the objective function is achievable while satisfying the linearized constraints. These measures have been adapted for use in derivative-free optimization context [70, 174, 73], where the derivative of the function is approximated by that of the surrogate model. The objective function of the problem defined in Eq. (4.22) can be written as follows:

$$\psi_k^r = ||\nabla f^r(x_k) + \xi^r|| \tag{4.23}$$

where $\xi^r = \lambda - \nu_L + \nu_U$. Similar problem as Eq. (4.22a)-4.22n with exact criticality measure by using derivative of the function may also be defined as follows:

$$\psi_k = ||\nabla f(x_k) + \xi|| \tag{4.24}$$

When $\psi_k = 0$ holds at a point x_k, a candidate point is said to be critical. At the final iteration of the Correcting Algorithm, a critical point is returned (denoted by $\bar{x}^p$). The MILP optimization problem in Eq. (4.22) is then solved to obtain the corresponding Lagrange multipliers ($\bar{\nu}_L^p = \nu_L$, $\bar{\nu}_U^p = \nu_U$, $\bar{\lambda}^p = \lambda$).

4.3.2.4 Interpolation Set and Surrogate Modeling

As mentioned earlier, the original function is assumed to be unknown and derivatives are unavailable. Therefore, input-output data is used to build an inexpensive data-driven/surrogate/metamodel/reduced-order model to approximate the original function. Sampling and modeling are critical steps in any model-based blackbox optimization algorithm. Samples should be provided such that it spans the entire space and the model should have the ability to capture the curvature and multimodality feature of the original function.

The purpose of the sampling phase is to obtain points at which the simulation should be performed and forms the primary step of developing predictive models. Interested reader may

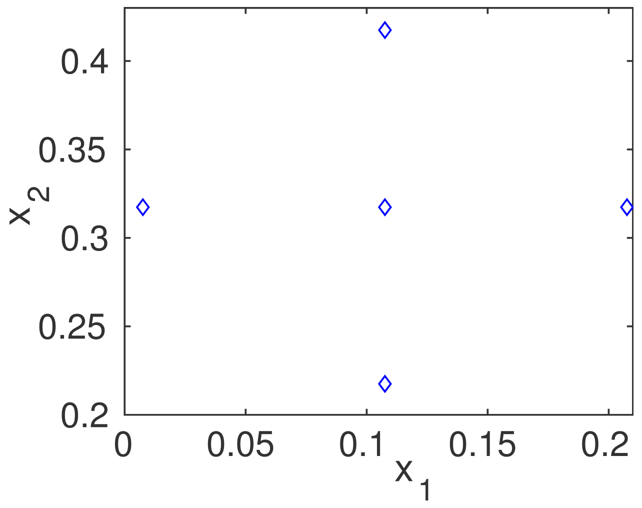


Figure 4.4: Initial interpolating samples generated around a point

refer to a recent review article by [100] for more information about several static and adaptive sampling schemes. The procedure by [90] is followed to obtain an initial interpolation set $\mathbb{Y} = \{c^1, c^2, \ldots, c^{2n+1}\}$. Rules to generate more points was also proposed but in the algorithm, the number of points are restricted to $2n+1$. c^i and c^{n+i} are defined for $i = 1, 2, \ldots, n$:

$$\begin{aligned} c^{i+1} &= c_i^1 + \rho_0 e_i, \quad c^{n+i} = c_i^1 - \rho_0 e_i, \quad if \quad x_i^L < c_i^1 < x_i^U \\ c^{i+1} &= c_i^1 + \rho_0 e_i, \quad c^{n+i} = c_i^1 + 2\rho_0 e_i, \quad if \quad c_i^1 = x_i^L \\ c^{i+1} &= c_i^1 - \rho_0 e_i, \quad c^{n+i} = c_i^1 - 2\rho_0 e_i, \quad if \quad c_i^1 = x_i^U \end{aligned} \tag{4.25}$$

where e_i is the i^{th} component of the coordinate vector, c^1 and ρ_0 are the trust-region center and size respectively. The set of samples with $c^1 = (0.1076, 0.3174)$ and $\rho_0 = 0.1$ are shown in Figure 4.4.

In contrast to derivative-based nonlinear programs where Taylor based models are used to approximate the function, interpolating or regression models are used in blackbox optimization algorithms. In the literature, several surrogate models including quadratic [175, 176, 90]), polynomial [61], kriging [55] and radial basis functions [52, 60, 177] have been used for optimization of blackbox problems. [78] proposed to select surrogate model amongst quadratic, kriging and signomial based on the cross validation error. In this work, the performance of cubic radial basis function and quadratic models are compared on test problems. In this section, the details of constructing quadratic model are omitted and only radial basis function will be discussed.

Radial basis functions (RBFs) [60] are a broad class of interpolating functions that are given by linear combination of nonlinear basis functions as follows:

$$s(x) = \sum_{j=1}^{|\mathbb{Y}|} \omega_j \phi(||x - c^j||_2) + p(x) \tag{4.26}$$

The basis function (ϕ) can be linear ($\phi = r$), cubic ($\phi = r^3$), multiquadratic ($\phi = \sqrt{\gamma^2 + r^2}$), thin plate spline ($\phi = r^2 log\ \ r$) and Gaussian ($\phi = e^{-\gamma r^2}$). In general, $p(x)$ is a polynomial function. In this work, cubic is used as the basis function and linear function as $p(x)$ and refer it to as cubic radial basis function (CRBF). The form of the surrogate model used is:

$$f^r(x) = \sum_{i=1}^{n} b_i x_i + \sum_{j=1}^{|\mathbb{Y}|} \omega_j \left(\sqrt{\sum_{i=1}^{n} (x_i - c_i^j)^2} \right)^3 \tag{4.27}$$

where b_i and ω_j are the unknown parameters.

A linear optimization problem is formulated that imposes interpolating and conditional definiteness conditions [60] to estimate the model parameters and it is solved using GAMS/CPLEX 12.6.0.1:

$$\min_{\substack{b_i,\omega_j \\ SP1_i,SN1_i \\ ,SP2_j,SN2_j}} \quad \sum_{i=1}^{n} (SP1_i + SN1_i)) + \sum_{j=1}^{|\mathbb{Y}|} (SP2_j + SN2_j) \tag{4.28a}$$

$$s.t. \quad \sum_{i=1}^{n} b_i c_i^j + \sum_{j'=1}^{|\mathbb{Y}|} \omega_{j'} \left(\sqrt{\sum_{i=1}^{n} (c_i^{j'} - c_i^j)^2} \right)^3 = f_j + SP2_j - SN2_j, \quad \forall j = \{1, \cdots, |\mathbb{Y}|\} \tag{4.28b}$$

$$\sum_{j=1}^{|\mathbb{Y}|} \omega_j c_i^j = SP1_i - SN1_i, \quad \forall i = \{1, \cdots, n\} \tag{4.28c}$$

$$SP1_i, SN1_i, SP2_j, SN2_j \geq 0 \tag{4.28d}$$

The above parameter estimation problem is formulated as a feasibility problem with slack variables ($SP1_i, SN1_i, SP2_j, SN2_j$). The sum of the slack variables is minimized so that the interpolation condition in Eq. (4.28b) and conditional definiteness in Eq. (4.28c) are satisfied. A similar linear optimization problem is formulated to estimate the parameters of quadratic model.

4.3.2.5 Model Improvement

It is important to account for the error between the objective function and surrogate model to guarantee convergence of Algorithm 4.1 to a first-order critical point. To ensure convergence to the optima of P5, the following condition need to be satisfied:

$$||\nabla f(x_k) - \nabla f^r(x_k)|| \leq C_1 \rho_k \tag{4.29}$$

where, ρ_k and x_k are the trust-region size and center at iteration k respectively. The parameter C_1 depends on the parameter $\Lambda_{\mathbb{Y}}$ that indicates the poisedness of the interpolation set. The condition given by Eq. (4.29) ensures that the surrogate model satisfies Taylor-like error bounds. The basic idea of introducing this concept is to ensure that as the trust-region step approaches zero, the algorithm converges to the true local optima. Therefore, as a consequence of Eq. (4.29), a descent direction would ultimately be found as the trust-region or line-search step is decreased unless the current iterate is an optimum solution. Hence, it is critical to bound $\Lambda_{\mathbb{Y}}$ especially when the iterations fail to provide a better point. $\Lambda_{\mathbb{Y}}$ provides a basis to estimate how well the interpolating sample set $\mathbb{Y}$ spans the search space and is defined as follows:

$$\Lambda_Y \geq \max_{i \in \mathbb{Y}} \max_{x \in \mathbb{B}} |l_i(x)| \tag{4.30}$$

where $\mathbb{B}$ denotes the set of points in $[x^l, x^u]$. For a set of interpolating points, $\mathbb{Y} = \{c^1, \ldots, c^{p'}\}$, a basis of polynomials $l_j(x), j = 0, \cdots, q'$ of degree $\leq d'$, is called basis of Lagrange polynomials if the following holds:

$$l_i(y^j) = \delta_{ij} \tag{4.31}$$

To uniquely obtain Lagrange polynomial basis of $d' = 1$ requires $p' = q' = n + 1$ samples, while $d' = 2$ uses $p' = q' = \frac{1}{2}(n + 1)(n + 2)$ samples. As mentioned earlier, the Correcting Algorithm maintaints a set of $p' = 2n + 1$ interpolation points at every iteration. Therefore, under determined Lagrange polynomials of $d' = 2$ are denoted and represented as:

$$l_f(x) = a + s^T x + \frac{1}{2} x^T H x \tag{4.32}$$

Note that the model improvement step is called to replace a point c^f and the new point is obtained by optimizing the Lagrange polynomial corresponding to c^f represented as $l_f(x)$. Hence, obtaining Lagrange polynomials corresponding to all the interpolating points is unnecessary as was done in [90]. A convex optimization is formulated, where norm of the Hessian of the Lagrange polynomial corresponding to the point c^f is minimized to estimate its parameters using GAMS/CONOPT [178]:

$$\begin{aligned} \min_{a,g,H} \quad & ||\nabla^2 l_f|| \\ \textit{s.t.} \quad & l_f(y^i) = \delta_{if} \quad \forall i \in \mathbb{Y} \end{aligned} \tag{4.33}$$

Pivoting algorithms have also been proposed in the literature [36, 52] to maintain poised set of interpolation points. However, the absolute value of the Lagrange polynomial, $l_f(x)$ is maximized to obtain a new candidate point $\bar{x}^l$ and c^f is replaced. Lagrange polynomials are nonconvex with bilinear terms and ideally should be optimized globally but that can be computationally demanding. Therefore, the following optimization problem is solved locally using CONOPT [178]:

$$\begin{aligned} \max_{x} \quad & \pm\, l_f(x) \\ \textit{s.t.} \quad & \mathbf{1}^T x = \tau^l + t^p(\tau^u - \tau^l) \\ & x \in [x^l, x^u] \end{aligned} \tag{4.34}$$

4.3.3 Selecting α

If the lower envelope ($G(t)$) continues to decrease, i.e., $G(t^p) \leq G(t^{p-1})$, a step is needed to select next parameter value t^{p+1} at which P5 is solved. An adaptive step length rule is used which is governed by number of evaluations taken by the correcting algorithm. The premise is that the number of evaluations taken by correcting algorithm is dependent on the distance of the initial guess and the optimal solution. Furthermore, this distance is dependent on the step length. Therefore, if the step length is small, correcting algorithm would take less number of evaluations and more function calls would be needed for larger step length. Let N^p denote the number of evaluations taken by the correcting algorithm at parameter value t^p. Let α^p denote the step length at t^p to obtain the new design point t^{p+1} at which $G(t)$ is next evaluated. The updating scheme for α^p is:

$$\begin{aligned} &\alpha^p = \text{min}\{t^u - t^p, 0.5\alpha^{p-1}\}, \quad if \;\; N^p > \text{max}_t(N^t) \quad \forall t \in \{1, \cdots, p-1\} \\ &\alpha^p = \text{min}\{t^u - t^p, 1.5\alpha^{p-1}\}, \quad otherwise \end{aligned} \tag{4.35}$$

4.3.4 Updating t^l and t^u

In this section, the methodology adopted to update bounds on t is described. At a given iteration of UNIPOPT, if an increase in the lower envelope is observed (i.e., $G(t^{p+1}) > G(t^p)$), a surrogate model to approximate $G(t)$ is constructed using samples in the interpolation set $\mathbb{Z} = \{t^1, \cdots, t^P\}$. Eq. (4.36) is then solved to obtain a new point t^r:

$$\begin{aligned} &\min_t \;\; G^r(t) \\ &s.t. \quad t \in [t^l, t^u] \end{aligned} \tag{4.36}$$

The current iterate t_0 is chosen as the design point in $\mathbb{Z}$ corresponding to the minimum of $G(t)$:

$$t_0 = arg \; \min_{t \in \mathbb{Z}} \; G(t) \tag{4.37}$$

EPIC is then applied from the closest interpolating point to t^r to obtain $G(t^r)$. The bounds on t are updated using conditions given in Eq. (4.38).

$$R = \frac{G(t^r) - G(t_0)}{G^r(t^r) - G^r(t_0)}$$
$$\hat{\Delta} = \begin{cases} \lambda^{inc}(t^u - t^l), & \text{if } \; R \geq \eta_1 \\ t^u - t^l, & \text{if } \; \eta_0 \leq R < \eta_1 \\ \lambda^{dec}(t^u - t^l), & \text{if } \; R < \eta_0 \end{cases} \tag{4.38}$$

The bounds in the original space as well as the univariate space are then updated:

$$t^l = t_0 - \hat{\Delta}, \quad t^u = t_0 + \hat{\Delta} \tag{4.39}$$

Before moving to the next iteration, the samples on the lower envelope within the new bounds are stored and will be used in the constructing the surrogate model in the next iteration. Note that the surrogate model to approximate G is full linear when G is differentiable because through the construction of the algorithm, the interpolation set $\mathbb{Z}$ always lie within the trust-region and are well-poised. The reader can refer [70] for more details of the algorithm.

4.4 Convergence

In order to prove the convergence, it is important to prove that the correcting algorithm as well as the overall framework used to obtain and optimize the lower envelope ($G(t)$) converges to a first-order critical point. Next, the convergence proof of the correcting algorithm is provided.

4.4.1 Convergence Proof of Correcting Algorithm to First-order Critical Point

The global convergence for unconstrained blackbox optimization for the form of algorithm given in Algorithm 1 was proven by [179] and [180]. To the best of my knowledge, the convergence analysis for the constrained case has not been done. The steps in [180] will be followed to extend the analysis for constrained case. The proof presented here is applicable to black-box problems with known constraints for the form of algorithm as Algorithm 4.1. The proof also does not require an initial point to be feasible but solving Eq. (4.21) exactly will yield atleast a feasible point. The analysis is valid when the surrogate model, f^r is an interpolating quadratic model. The model can be represented at iteration k as Q_k with the following functional form:

$$Q_k(x) = Q_k(x_k) + (x - x_k)^T g_k + \frac{1}{2}(x - x_k)^T G_k (x - x_k) \tag{4.40}$$

The following is assumed to hold true:

A1 The objective function, f is continuous differentiable.

A2 The objective function, f is bounded from below.

A3 The surrogate model interpolates the original function at the trust-region center (x_k), $Q_k(x_k) = f(x_k)$.

A4 The Hessian of the surrogate model is bounded, $||\nabla^2 Q_k(x)|| \leq \mathcal{Q}^H$.

A5 The Hessian of the objective function is bounded, $||\nabla^2 f(x)|| \leq \mathcal{F}^H$.

A6 Let $x^r = x_k + d_k$ with $||d_k|| \leq \rho_k$ and it is possible to find a point that reduces the surrogate model as follows:

$$Q_k(x_k) - Q_k(x_k + d_k) \geq C_0 \psi_k^r \rho_k \tag{4.41}$$

where, $C_0 \in (0, 1)$. The proof is provided next.

The first result is taken from [179] which gives the bound on the difference of the gradients of the original model and the surrogate model at the trust-region center.

Lemma 4.1. *Let ρ_k be the trust-region radius at iteration k. Then a constant C_1 exists such that*

$$||\nabla Q_k(x_k) - \nabla f(x_k)|| \leq C_1 \rho_k \tag{4.42}$$

Proof. The result holds when there are atleast $n + 1$ samples. For details of the proof, refer [179] and Lemma 4.2 in [180]. □

A result is presented next that bounds the difference between ψ_k and ψ_k^r. The result is motivated by the proof done in Lemma 3.5 in [181].

Lemma 4.2. *Assuming that SCS condition is satisfied at a point x_k, then*

$$|\psi_k - \psi_k^r| \leq C_1 \rho_k \tag{4.43}$$

Proof. Case 1. Assume $\psi_k \geq \psi_k^r$,

$$\begin{aligned} 0 \leq \psi_k - \psi_k^r &= ||\nabla f_k + \xi|| - ||\nabla Q_k + \xi^r|| \\ &= ||\nabla f_k + \xi|| - ||\nabla f_k + \xi^r|| + ||\nabla f_k + \xi^r|| - ||\nabla Q_k + \xi^r|| \end{aligned} \tag{4.44}$$

Note that $||\nabla f_k + \xi|| - ||\nabla f_k + \xi^r|| \leq 0$ (since ξ minimizes ψ_k). Expanding the terms in the assumption that $\psi_k \geq \psi_k^r$:

$$\begin{aligned} &||\nabla f_k + \xi|| - ||\nabla Q_k + \xi^r|| \geq 0 \\ &\Rightarrow ||\nabla f_k + \xi^r|| - ||\nabla Q_k + \xi^r|| \geq 0 \end{aligned} \tag{4.45}$$

Second inequality holds since ξ minimizes ψ_k. Subsequently, using reverse triangle inequality and Lemma 4.1:

$$0 \leq ||\nabla f_k + \xi^r|| - ||\nabla Q_k + \xi^r|| \leq C_1 \rho_k \tag{4.46}$$

Using Eq. (4.44) and Eq. (4.46), the following is obtained:

$$0 \leq \psi_k - \psi_k^r \leq C_1 \rho_k \tag{4.47}$$

Now, considering the second case.

Case 2. Assume that $\psi_k^r > \psi_k$.

$$\begin{aligned} 0 \leq \psi_k^r - \psi_k &= ||\nabla Q_k + \xi^r|| - ||\nabla f_k + \xi|| \\ &= ||\nabla Q_k + \xi^r|| - ||\nabla Q_k + \xi|| + ||\nabla Q_k + \xi|| - ||\nabla f_k + \xi|| \end{aligned} \tag{4.48}$$

Similarly, $||\nabla Q_k + \xi^r|| - ||\nabla Q_k + \xi|| \leq 0$ and $||\nabla Q_k + \xi|| - ||\nabla f_k + \xi|| > 0$. Again, using triangle inequality and Lemma 4.1:

$$0 < ||\nabla Q_k + \xi|| - ||\nabla f_k + \xi|| \leq C_1 \rho_k \tag{4.49}$$

Finally, the following is obtained:

$$0 < \psi_k^r - \psi_k \leq C_1 \rho_k \tag{4.50}$$

From Eq. (4.50) and Eq. (4.47), the desired result stated in Eq. (4.43) is proven. □

Lemma 4.3. *For all k, the following holds:*

$$|Q_k(x_k + d_k) - f(x_k + d_k)| \leq (C_1 + \frac{1}{2}(\mathcal{Q}^H + \mathcal{F}^H))\rho_k^2 \tag{4.51}$$

Proof. Refer [179] and Lemma 4.4 in [180]. □

Next, the conditions are derived that ensures that an iteration is successful. An iteration is considered successful if:

$$\frac{f(x_k) - f(x_k + d_k)}{Q_k(x_k) - Q_k(x_k + d_k)} = r_k > \eta, \tag{4.52}$$

where $\eta \in (0, 1]$.

Lemma 4.4. *Let C_2 be a constant defined as follows:*

$$C_2 = (3C_1 + 2\mathcal{Q}^H + 2\mathcal{F}^H) max(1, \frac{1}{C_0(1-\eta)}) \tag{4.53}$$

If at iteration k, the following holds true

$$\psi_k > C_2 \rho_k, \tag{4.54}$$

then the iteration k is considered to be successful, i.e., it satisfies Eq. (4.52).

Proof. From the assumption in Eq. (4.41), following will hold as well,

$$Q_k(x_k + d_k) \leq Q_k(x_k) - C_0 \psi_k^r \rho_k + \frac{1}{8} C_0 \mathcal{Q}^H \rho_k^2 \tag{4.55}$$

Using triangle inequality, Eq. (4.54) and Lemma 4.2,

$$\psi_k^r \geq \psi_k - |\psi_k - \psi_k^r| > (C_2 - C_1)\rho_k \tag{4.56}$$

Using the above equation in Eq. (4.55),

$$Q_k(x_k) - Q_k(x_k + d_k) > C_0(C_2 - C_1 - \frac{\mathcal{Q}^H}{2})\rho_k^2 \tag{4.57}$$

Using definition of C_2, he have that

$$C_2 - C_1 - \frac{\mathcal{Q}^H}{2} > 0 \tag{4.58}$$

Furthermore,

$$C_2 - C_1 - \frac{\mathcal{Q}^H}{2} > C_2 - (1 + \frac{1}{C_0(1-\eta)})C_1 - \frac{1}{2}(1 + \frac{1}{C_0(1-\eta)})\mathcal{Q}^H - \frac{1}{2}\frac{1}{C_0(1-\eta)}\mathcal{F}^H > 0 \tag{4.59}$$

From Eq. (4.59),

$$C_2 - C_1 - \frac{\mathcal{Q}^H}{2} > \frac{1}{C_0(1-\eta)}(C_1 + \frac{1}{2}(\mathcal{Q}^H + \mathcal{F}^H)) \tag{4.60}$$

Finally,

$$\frac{C_1 + \frac{1}{2}(\mathcal{Q}^H + \mathcal{F}^H)}{C_2 - C_1 - \frac{\mathcal{Q}^H}{2}} < C_0(1-\eta) \tag{4.61}$$

$$
\begin{aligned}
|r_k - 1| &= \left| \frac{f(x_k + d_k) - f(x_k)}{Q_k(x_k + d_k) - Q_k(x_k)} - 1 \right| \\
&\leq \left| \frac{f(x_k + d_k) - Q_k(x_k + d_k)}{Q_k(x_k) - Q_k(x_k + d_k)} \right| + \left| \frac{f(x_k) - Q_k(x_k)}{Q_k(x_k) - Q_k(x_k + d_k)} \right|
\end{aligned} \tag{4.62}
$$

Using assumption A3, Lemma 4.3 and Eq. (4.57),

$$
|r_k - 1| \leq \frac{C_1 + \frac{1}{2}(\mathcal{Q}^H + \mathcal{F}^H)}{C_0(C_2 - C_1 - \frac{\mathcal{Q}^H}{2})} \tag{4.63}
$$

Using Eq. (4.61) completes the second part of the proof that $r_k > \eta$. □

Lemma 4.5. *The number of iterations at which the trust-region size, ρ_k remains the same are finite.*

Proof. By construction of the algorithm, an iterate is only acceptable when a decrease in the objective function is obtained. Therefore, the sequence of iterates $(f(x_k))$ are monotonically decreasing. From assumption A2,

$$
\lim_{k \to \infty} (f(x_k) - f(x_{k+1})) = 0 \tag{4.64}
$$

Suppose by contradiction that the number of iterations are infinite. By construction of the algorithm, ρ_k is decreased when the iteration is not successful and the model is ensured to be accurate. Since the trust-region size, ρ_k is not being decreased, the number of successful iterations are infinite. When the iterations are successful, following holds,

$$
f(x_k) - f(x_k + d_k) > \eta(Q_k(x_k) - Q_k(x_k + d_k)) \tag{4.65}
$$

Using Eq. (4.57),

$$
f(x_k) - f(x_k + d_k) > \eta(C_2 - C_1 - \frac{\mathcal{Q}^H}{2})\rho_k^2 \tag{4.66}
$$

This violates Eq. (4.64) and therefore the number of iterations corresponding to ρ_k should be finite. □

Corollary 4.1. *The sequence of the trust-region size, ρ_k generated by the algorithm converges to zero.*

Proof. Due to fact that ρ_k is monotonically decreasing, this is a direct consequence of the last lemma. From Eq. (4.66) and Eq. (4.64), it can be concluded that $\lim_{k\to\infty} \rho_k = 0$. $\square$

Lemma 4.6. *The sequence of points generated by the algorithm has a critical limit point.*

Proof. Consider set of unsuccessful iterations $\mathbb{N}$ when the objective function could not be decreased. From Lemma 4.4, for an unsuccessful iteration,

$$\psi_k \leq C_1 \rho_k \quad \forall k \in \mathbb{N} \tag{4.67}$$

From Corollary 4.1, it is known that the trust-region size tends to zero. Therefore,

$$\liminf_{k\to\infty} \psi_k = 0 \tag{4.68}$$

This completes the proof that the limit point is a critical point. $\square$

The previous result (Eq. (4.68)) is a weak convergence proof. A stronger result would be that $\lim_{k\to\infty} \psi_k = 0$. The results provided next will ensure that the algorithm actually converges to a critical point.

Let k_ϵ be the first iteration so that trust-region size ρ_k satisfies

$$\rho_k \leq \frac{\epsilon}{C_2}, \quad \forall k \geq k_\epsilon \tag{4.69}$$

where $\epsilon > 0$. From Corollary 4.1, it is known that such iteration will exist. Since the trust-region ρ_k is always decreased, for all the iterations $k \geq k_\epsilon$, the condition on the trust-region given in Eq. (4.69) will hold.

Lemma 4.7. *For all the iterations $k \geq k_\epsilon$, the following condition on criticality measure holds:*

$$|\psi_{k+1} - \psi_k| < \epsilon, \quad \forall k \geq k_\epsilon \tag{4.70}$$

Proof. Using mean value theorem on ∇f,

$$\frac{\nabla f(x_{k+1}) - \nabla f(x_k)}{x_{k+1} - x_k} = \nabla^2 f(x_c) \tag{4.71}$$

where x_c is some point that lies between x_k and x_{k+1}. Taking norm on both sides, using Cachy-Schwarz inequality, assumption A5 and the fact that $||x_{k+1} - x_k|| \leq \rho_k$.

$$||\nabla f(x_{k+1}) - \nabla f(x_k)|| \leq \mathcal{F}^H \rho_k \tag{4.72}$$

Using Eq. (4.69) and definition of C_2,

$$\mathcal{F}^H < C_2 \leq \frac{\epsilon}{\rho_k}, \quad \forall k \geq k_\epsilon \tag{4.73}$$

Finally, using Eq. (4.72) and Eq. (4.73),

$$||\nabla f(x_{k+1}) - \nabla f(x_k)|| < \epsilon, \quad \forall k \geq k_\epsilon \tag{4.74}$$

Following similar procedure as in Lemma 4.2, the following two cases are considered:

Case 1. Assume $\psi_{k+1} \geq \psi_k$,

$$\begin{aligned} 0 \leq \psi_{k+1} - \psi_k &= ||\nabla f(x_{k+1}) + \xi_{k+1}|| - ||\nabla f(x_k) + \xi_k|| \\ &= ||\nabla f(x_{k+1}) + \xi_{k+1}|| - ||\nabla f(x_{k+1}) + \xi_k|| + ||\nabla f(x_{k+1}) + \xi_k|| - ||\nabla f(x_k) + \xi_k|| \end{aligned} \tag{4.75}$$

$||\nabla f(x_{k+1}) + \xi_{k+1}|| - ||\nabla f(x_{k+1}) + \xi_k|| \leq 0$ (since ξ_{k+1} minimizes ψ_{k+1}). Expanding the terms in the assumption that $\psi_{k+1} \geq \psi_k$,

$$\begin{aligned} &||\nabla f(x_{k+1}) + \xi_{k+1}|| - ||\nabla f(x_k) + \xi_k|| \geq 0 \\ &\Rightarrow ||\nabla f(x_{k+1}) + \xi_k|| - ||\nabla f(x_k) + \xi_k|| \geq 0 \end{aligned} \tag{4.76}$$

Subsequently, using reverse triangle inequality and from Eq. (4.74),

$$0 \leq \psi_{k+1} - \psi_k < \epsilon, \quad \forall k \geq k_\epsilon \tag{4.77}$$

Case 2. Assume $\psi_k > \psi_{k+1}$:

$$\begin{aligned}\psi_k - \psi_{k+1} &= ||\nabla f(x_k) + \xi_k|| - ||\nabla f(x_{k+1}) + \xi_{k+1}|| \\ &= ||\nabla f(x_k) + \xi_k|| - ||\nabla f(x_k) + \xi_{k+1}|| + ||\nabla f(x_k) + \xi_{k+1}|| - ||\nabla f(x_{k+1}) + \xi_{k+1}||\end{aligned} \tag{4.78}$$

Note that $||\nabla f(x_k) + \xi_k|| - ||\nabla f(x_k) + \xi_{k+1}|| \leq 0$ (since ξ_k minimizes ψ_k). Expanding the terms in the assumption that $\psi_k > \psi_{k+1}$,

$$\begin{aligned}&||\nabla f(x_k) + \xi_k|| - ||\nabla f(x_{k+1}) + \xi_{k+1}|| > 0 \\ &\Rightarrow ||\nabla f(x_k) + \xi_{k+1}|| - ||\nabla f(x_{k+1}) + \xi_{k+1}|| > 0\end{aligned} \tag{4.79}$$

Subsequently, using reverse triangle inequality and from Eq. (4.74),

$$0 < \psi_k - \psi_{k+1} < \epsilon, \quad \forall k \geq k_\epsilon \tag{4.80}$$

From Eq. (4.77) and Eq. (4.80), the desired result stated in Eq. (4.70) is obtained. □

Lemma 4.8. *If for an iteration $k \geq k_\epsilon$, the following holds,*

$$\psi_k > \epsilon \tag{4.81}$$

then k is a successful iteration and

$$f(x_k) - f(x_{k+1}) > \frac{2}{3}\rho_k \epsilon \eta \tag{4.82}$$

Proof. Using Eq. (4.81) and Eq. (4.69),

$$\psi_k > C_2\rho_k \tag{4.83}$$

Hence, k is a successful iteration by Lemma 4.4. Using triangle inequality, Eq. (4.81) and Lemma 4.2,

$$\psi_k^r \geq \psi_k - |\psi_k - \psi_k^r| > \epsilon - C_1\rho_k, \quad \forall k \geq k_\epsilon \tag{4.84}$$

Using this and Eq. (4.69) in Eq. (4.41),

$$Q_k(x_k) - Q_k(x_k + d_k) > (\epsilon - C_1\rho_k)\rho_k > (\epsilon - \frac{C_1}{C_2}\epsilon)\rho_k, \quad \forall k \in k_\epsilon \tag{4.85}$$

Using the definition of C_2, it is known that $C_2 > 3C_1$ and the fact that the iteration is successful,

$$f(x_k) - f(x_k + d_k) > \frac{2}{3}\epsilon\rho_k\eta, \quad \forall k \geq k_\epsilon \tag{4.86}$$

□

Theorem 4.3. *If the assumptions A1-A6 hold true, then all the limit points of the sequence of points generated by Algorithm 4.1 is a critical point, i.e.* $\lim_{k\to\infty}\psi_k = 0$.

Proof. This result is proven by contradiction. Lets define a set of iterations $\mathbb{K}$ such that

$$\mathbb{K} = \{k \in \mathbb{S} \mid \psi_k > 4\epsilon\} \tag{4.87}$$

where $\mathbb{S}$ denotes a set of successful iterations. Assume that the set $\mathbb{K}$ is infinite. Assume that $k \geq k_\epsilon + 2$ and let $q_k > k$ be the first index such that $\psi_{q_k+1} \leq \epsilon$. The iteration q_k is guaranteed to exist due to Lemma 4.6. Using triangle inequality and Lemma 4.7, the following is obtained,

$$\psi_{k-1} \geq \psi_k - |\psi_k - \psi_{k-1}| > 3\epsilon \tag{4.88}$$

Similarly, $\psi_{k-2} > 2\epsilon$. Therefore, all the iterations between $[k-2, q_k]$ are successful due to Lemma 4.8. Using reverse triangle inequality, and the definition of q_k and k,

$$|\psi_{q_k+1} - \psi_k| \geq ||\psi_{q_k+1}| - |\psi_k|| > 3\epsilon \tag{4.89}$$

Using the above relation, the triangle inequality and Lemma 4.7,

$$3\epsilon < |\psi_{q_k+1} - \psi_k| = |\sum_{i=1}^{q_k+1-k} (\psi_{k+i} - \psi_{k+i-1})| < \theta\epsilon \tag{4.90}$$

where $\theta = q_k + 1 - k$. Using the Mean Value Theorem, assumption A5 and the triangle inequality,

$$|\nabla f(x_{q_k+1}) - \nabla f(x_k)| \leq \mathcal{F}^H ||x_{q_k+1} - x_k|| \tag{4.91}$$

The trust-region size from iteration k to q_k are the same because by the construction of the algorithm, it is not reduced when the iterations are successful. Using this fact and Eq. (4.91),

$$|\nabla f(x_{q_k+1}) - \nabla f(x_k)| \leq \mathcal{F}^H \theta \rho_k \tag{4.92}$$

Using above equation, it can be proved using the procedure in Lemma 4.7 that

$$|\psi_{q_k+1} - \psi_k| \leq \mathcal{F}^H \theta \rho_k \tag{4.93}$$

From Eq. (4.93) and Eq. (4.90),

$$\theta > \frac{3\epsilon}{\mathcal{F}^H \rho_k} \quad \text{and} \quad \theta > 3 \tag{4.94}$$

It is known that the sequence of iterates $f(x_k)$ is non-increasing,

$$f(x_k) - f(x_{q_k+1}) \geq \sum_{i=1}^{q_k+1-k} (f(x_{k+i-1}) - f(x_{k+i})) \quad \forall k \in \mathbb{K} \tag{4.95}$$

Furthermore, using Lemma 4.8 and Eq. (4.94),

$$f(x_k) - f(x_{q_k+1}) > \frac{2}{3}\theta\rho_k\epsilon\eta > \frac{2\epsilon^2\eta}{\mathcal{F}^H} \tag{4.96}$$

By assumption A2, $f(x_k) - f(x_{q_k+1}) \to 0$ leading to contradiction in Eq. (4.96). Therefore the assumption in Eq. (4.87) that $\mathbb{K}$ is infinite must not hold. $\square$

4.4.2 Convergence of Overall Framework

Since the overall procedure is similar to NOWPAC [70], interested reader may refer their paper for the convergence proof. Note that if the following assumptions are satisfied, the algorithm for optimizing G will converge to the first-order critical point:

1. G is continuously differentiable and have Lipschitz continuous gradients.
2. The function G is bounded from below.
3. The Hessian of the G is bounded such that $||\nabla^2 G|| \leq \mathcal{G}^H$

4.5 Computational Studies

The UNIPOPT algorithm has been applied to a test suite of 393 problems. These problems are listed in [37]. The test suite comprises of 232 nonconvex smooth and 161 convex nonsmooth problems and the distribution of problems with respect to number of variables is provided in Figure 4.5. UNIPOPT is compared with other model-based BBO solvers. Specifically, the performance of UNIPOPT is compared to that of BOBYQA SNOBFIT, ORBIT, and IMFIL.

4.5.1 Illustrative Example 4.2

UNIPOPT and other solvers are applied on a two-dimensional nonconvex smooth problem, `denschnc` to demonstrate how the search space is explored by each of the solvers. The following

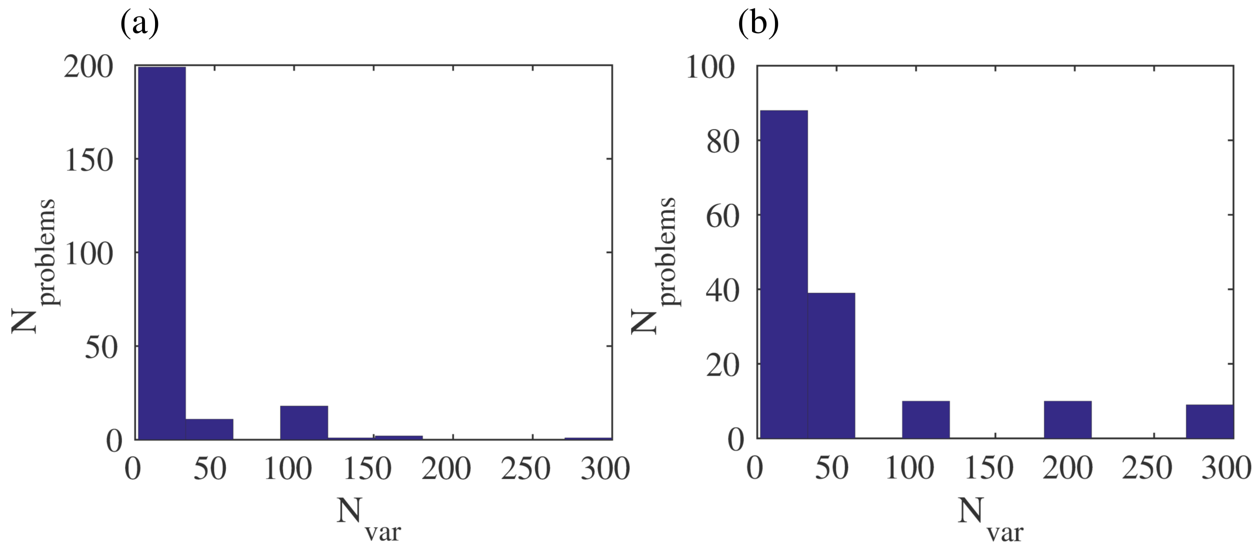


Figure 4.5: Distribution of problems with number of variables for (a) 232 nonconvex smooth and (b) 161 convex nonsmooth problems.

problem is optimized by each of the solvers:

$$\begin{aligned} &\min_{x} \; (-2+x_1^2+x_2^2)^2 + (-2+e^{(x_1-1)}+x_2^3)^2 \\ &\textit{s.t.} \quad x_1, x_2 \in [-9, 11] \end{aligned} \tag{4.97}$$

The global solution of this problem is 0. The samples used by each of the solver is provided in Figure 4.6 where the x and y axis represent the normalized variables, x_1 and x_2 respectively. It

Table 4.1: Best function values reported by solvers on `denschnc` and corresponding number of evaluations.

Solvers	BOBYQA	IMFIL	ORBIT	SNOBFIT	UNIPOPT
f^{min}	3.64	0.158	3.646	0.1834	1E-6
No. of evaluations	315	82	63	169	1512

can clearly be seen that all other solvers except SNOBFIT explore only local region and therefore

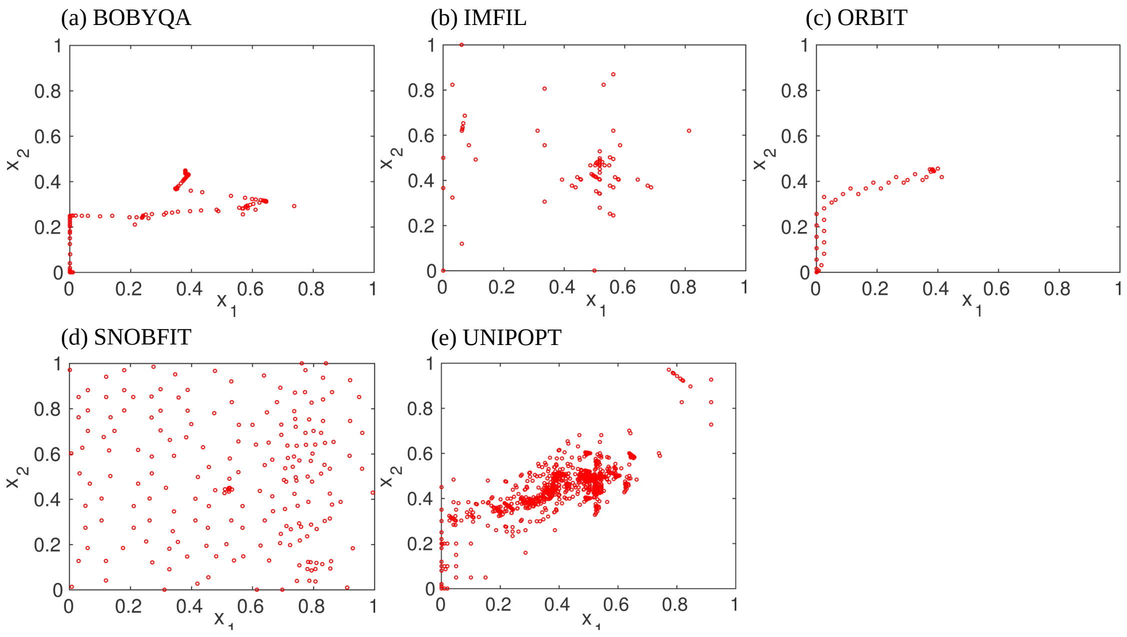


Figure 4.6: Distribution of samples used by (a) BOBYQA, (b) IMFIL, (c) ORBIT, (d) SNOBFIT, and (e) UNIPOPT.

converges to local optima. It can be observed that UNIPOPT has better exploration characteristics compared to BOBYQA, IMFIL, ORBIT. This is also evident from the solution obtained and are listed in Table 4.1. Although, the number of evaluations taken by UNIPOPT for a two-dimensional problem might seem a lot, the value reported in the table correspond to the final minimum value achieved by the algorithm. If a less accurate solution is acceptable, the required number of evaluations will decrease. For instance, if a solution of 0.01 is deemed satisfactory, UNIPOPT requires 1027 evaluations. At the end of first outer iteration, minimum value of 0.1977 is achieved in 137 evaluations. Table 4.2 shows for the first iteration of the UNIPOPT framework, the initial guess provided by the prediction step, final point obtained by correcting algorithm, corresponding function values and the norm difference between the two points. Note that the variables are normalized. It can be observed in the table that for this example, sensitivity theorem provides a good initial guess which is close to the actual optima. At each t^p before the lower envelope (G) starts increasing, sensitivity approximation given by Eq. (4.6) provides a sequence of points at which the objective function is decreasing, i.e. the trend of G is being followed.

Table 4.2: Initial guess and optima generated during the first iteration of the UNIPOPT framework for Illustrative Example 2.

t^p	$\hat{x}^p$	$\bar{x}^p$	$f(\hat{x}^p)$	$f(\bar{x}^p)$	$\|\|\hat{x}^p - \bar{x}^p\|\|_2$	iterations needed (k)
0	(0,0)	(0,0)	5.59E+5	5.59E+5	0	8
0.05	(0.05,0.05)	(0,0.1)	2.8E+5	1.35E+5	0.0707	8
0.1	(0,0.2)	(0,0.2)	2.69E+4	2.69E+4	0	21
0.175	(0,0.35)	(0.0477,0.3023)	6.98E+3	5.87E+3	0.0675	18
0.2125	(0.0412,0.3838)	(0.1076,0.3174)	4.45E+3	3.12E+3	0.0939	17
0.2687	(0.1578,0.3797)	(0.1953, 0.3422)	1.18E+3	962.26	0.0530	19
0.3531	(0.2610,0.4452)	(0.3184, 0.3878)	155.18	57.05	0.0812	20
0.4797	(0.4332,0.5262)	(0.5226, 0.4368)	3.41	0.23	0.1264	24
0.543	(0.5915, 0.4944)	(0.5658, 0.5201)	70.57	48.57	0.0363	17

4.5.2 Experimental Setup for the Solvers

All the algorithms are given a limit of 10,000 function evaluations in order to get the complete sense of the performance of all the solvers. So, depending on the computational budget of a user, appropriate solver can be chosen. In the computational comparison in this section, UNIPOPT uses cubic radial basis function (CRBF) as the surrogate model. Although, the convergence result shown in Section 4.4 holds when the surrogate model is quadratic, CRBF performs more efficiently. The algorithmic parameters for UNIPOPT are set as follows: $\epsilon_t = 10^{-3}$, $\eta_0 = 0.1$, $\eta_1 = 0.7$, $\lambda^{dec} = 0.5$, $\lambda^{inc} = 3$, $T = 20$, and $\epsilon_\rho = 10^{-5}$. For many of the problems, the bounds on the variables are not provided. So, when this happens, the following strategy is used to get the bounds:

$$x_i^L = x_i^* - 10e_i, \quad x_i^U = x_i^* + 10e_i \quad \forall i = \{1, \ldots, n\} \tag{4.98}$$

where, x_i^* represents the global solution found by BARON [182] or LINDOGLOBAL [183] and e_i represents the unit vector. For some problems, if appropriate bounds are not provided, that resulted in numerical difficulties. For example, some of the problems had exponential terms that led to very large objective function values. The bounds are heuristically tightened so that no numerical difficulties are encountered. Nevertheless, all the solvers are provided the same variable bounds. The lower bound (x^L) is given as the initial guess to all the solvers. It is assumed that no derivative is available and only function evaluations are possible. Two convergence criteria are used to compare the performance of the solvers. The first criteria is the ability of a solver to obtain a solution that is close to the global minima. Specifically, a solver is said to have solved a problem if a point x^f is such that it is within d fraction of the global minima (x^*):

$$f(x^f) \leq \max((1+d)f(x^*), f(x^*)+d) \tag{4.99}$$

The second criteria used for comparison is to test if the reduction in the objective obtained by a solver is comparable to the maximum reduction achieved. The criteria is defined as follows:

$$f(x_0) - f(x^f) \geq (1-\beta)(f(x_0) - f_M) \tag{4.100}$$

where x_0 is the initial guess, $1-\beta$ is the degree of reduction desired and f_M is the least of the minimum values obtained by each of the solvers for a particular problem.

4.5.3 Computational Results

UNIPOPT is applied to a test suite of 393 test problems comprising of sets of 161 convex nonsmooth and 232 nonconvex smooth black-box problems, and the performance is compared to those of other model-based solvers. A total of 1965 problem instances were tested.The results are discussed in the subsequent sections.

4.5.3.1 Convex Nonsmooth Problems

The performance of the solvers is compared on a set of 161 convex nonsmooth problems. Figure 4.7 (a) shows the performance of different model-based solvers on 161 convex nonsmooth problems by using $d = 0.01$ in Eq. (4.99) and $\beta = 0.001$ in Eq. (4.100) with CRBF as the surrogate model. In Figure 4.7(a), the y-axis indicates the fraction of problems that satisfy the two convergence criteria within 10,000 function evaluations and the x-axis represents the solvers.

UNIPOPT solves more number of problems compared to all other algorithms. It is able to obtain a solution within 1% of the global minima for 36 of 161 problems. BOBYQA ranks second by solving 15 problems followed closely by ORBIT that solves 13 problems. Although IMFIL and SNOBFIT are expected to be suited for nonsmooth problems, they are able to solve only 8 and 2 problems, respectively. Based on criteria in Eq. (4.100), UNIPOPT, BOBYQA, SNOBFIT, IMFIL and ORBIT solves 103, 74, 75, 80 and 94 problems, respectively. This implies that for majority of the problems, UNIPOPT is either finding the least or very close to the least function value compared to other solvers. The performance of the solvers for $d = 0.05$ and $\beta = 0.1$ is shown in Figure 4.7(b). It can be observed that on relaxing the convergence criteria, the performance of all the solvers improve. Especially, the performance of ORBIT increases dramatically and it is able to find a solution that is close to global optimum for 37 problems. However, UNIPOPT is still competitive and solves 39 problems. BOBYQA, SNOBFIT and IMFIL satisfy the global solution criteria for 23, 9 and 23 problems, respectively. In general, ORBIT and UNIPOPT performs better than BOBYQA. Note that in this comparison, ORBIT and UNIPOPT uses radial basis function as the surrogate model while BOBYQA uses quadratic function. One reason could be that radial basis function might be more effective in approximating nonsmooth problems. When β is increased to 0.1 in Eq. (4.100), IMFIL's performance increases dramatically. IMFIL and UNIPOPT performs superior and are able to solve 145 and 150 problems, respectively. BOBYQA, SNOBFIT and ORBIT solves 81, 111 and 115 problems, respectively. Based on the computational results, it can be concluded that UNIPOPT is competitive in finding an optimal solution and it also consistently finds the least objective function value compared to other solvers.

4.5.3.2 Nonconvex Smooth Problems

The performance of the solvers is also compared on a set of 232 nonconvex smooth problems. Figure 4.8(a) shows the performance of the solvers based on the convergence criteria given in Eq. (4.99) and Eq. (4.100) with $d = 0.01$ and $\beta = 0.001$, respectively while restricting the number of function evaluations to 10,000. UNIPOPT ranks first by obtaining a solution within 1% of the global minima for 144 problems. On the other hand, BOBYQA ranks second and

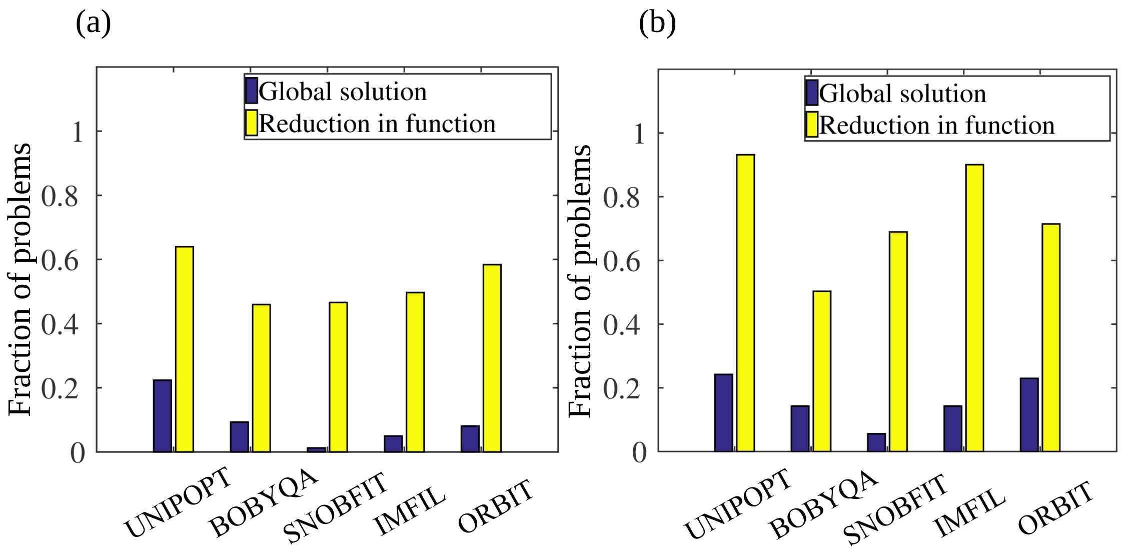


Figure 4.7: Fraction of problems solved by each of the solvers for 161 convex nonsmooth problems using CRBF. The performance is compared based on the two criteria given in Eqs. 4.99 and 4.100. The values of the parameters used is Figure (a) are $d = 0.01$ and $\beta = 0.001$, and Figure (b) are $d = 0.05$ and $\beta = 0.1$.

solves 124 problems. SNOBFIT, IMFIL and ORBIT are able to solve 82, 84 and 93 problems, respectively. When the solvers are compared based on their ability to reduce the initial function value, UNIPOPT, BOBYQA, SNOBFIT, IMFIL and ORBIT are able to solve 210, 173, 153, 187 and 190 problems, respectively. The convergence criteria are further relaxed by choosing $d = 0.05$ and $\beta = 0.1$ and the comparison results are shown in Figure 4.8(b). UNIPOPT, BOBYQA, SNOBFIT, IMFIL and ORBIT attains a solution within 5% of the global minima for 168, 133, 92, 94, and 110 problems, respectively. As expected, all the solvers satisfy the relaxed convergence criteria for more number of problems. However, the performance rank remains unchanged. Based on the criteria in Eq. (4.100) with $\beta = 0.1$, UNIPOPT, BOBYQA, SNOBFIT, IMFIL and ORBIT solves 228, 191, 209, 210 and 211 problems, respectively. From this computational comparison, it can be concluded that UNIPOPT explores the global space more efficiently compared to other solvers and this makes it competitive for solving nonconvex problems.

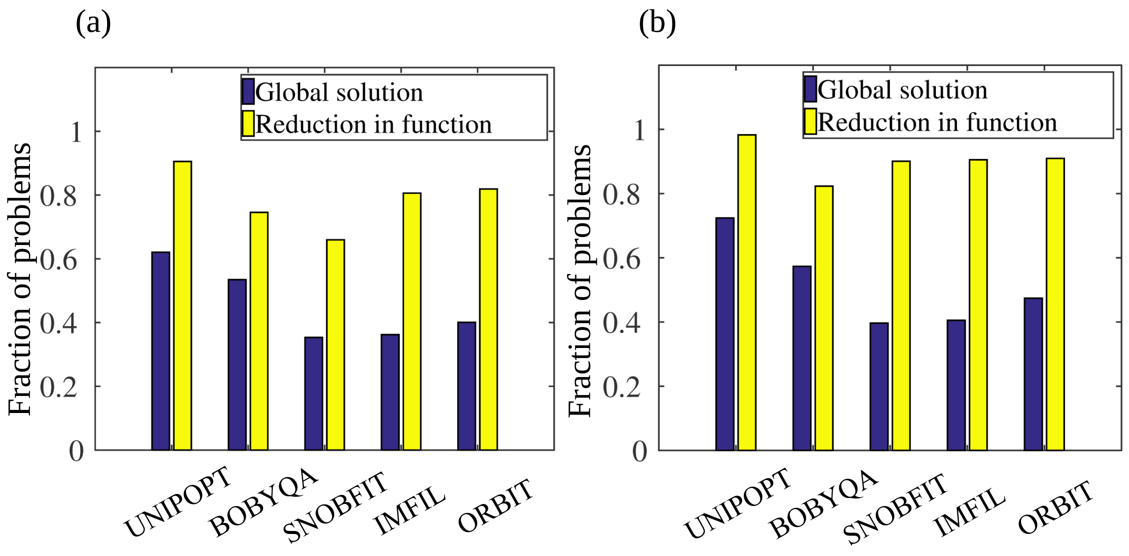


Figure 4.8: Fraction of problems solved by each of the solvers for 232 nonconvex smooth problems using CRBF. The performance is compared based on the two criteria given in Eqs. 4.99 and 4.100. The values of the parameters used is Figure (a) are $d = 0.01$ and $\beta = 0.001$, and Figure (b) are $d = 0.05$ and $\beta = 0.1$.

4.5.4 Comparison of Performance of Quadratic Model with Cubic Radial Basis Function

While both quadratic model and cubic radial basis function are used in existing solvers, it is not conclusive which one is better to use. Therefore, their performance is tested in the context of the proposed algorithm. Figure 4.9 compares the performance of the UNIPOPT on 232 nonconvex smooth problems using quadratic model and cubic radial basis function as the surrogate model in the Correcting Algorithm. When the CRBF model is used in the algorithm, it performs slightly better compared to the case when quadratic model is used. Their performance is also compared on a set of 161 convex nonsmooth problems. The comparison is shown in Figure 4.9. The two variants are compared based on their ability to obtain global minima using $d = 0.01$ in Eq. (4.99). It can be seen that UNIPOPT performs better when CRBF model is used.

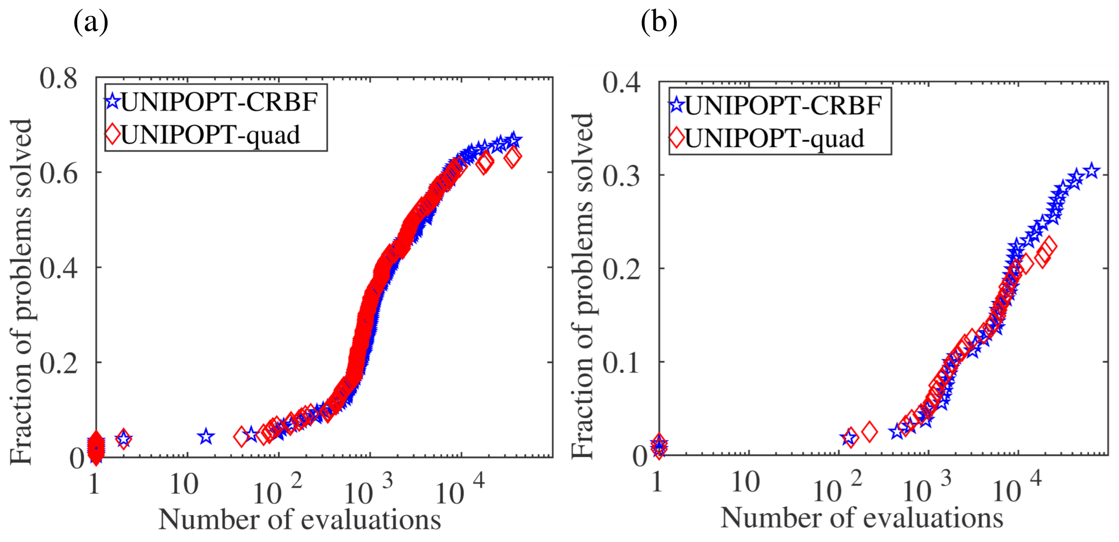


Figure 4.9: Fraction of problems solved as a function of number of evaluations for (a) 232 nonconvex smooth problems and (b) 161 convex nonsmooth problems using CRBF and quadratic model. A problem is deemed to be solved if the solution is within 1% of the global minima.

4.5.5 Comparison of Strategies for Optimizing the Lower Envelope

Besides the strategy presented earlier to optimize the lower envelope, there are other alternatives that can be used to optimize it. In this work, golden section type and Fibonacci algorithm were compared with the model based strategy presented earlier on a subset of 81 problems. The list of the problem name is provided in the supplementary material. Both golden section and Fibonacci algorithms are based on direct search that aims to reduce the variable bounds iteratively. In the implementation of golden section and Fibonacci algorithms, whenever value of G was required at new parameter value t^p, the predictor step was implemented from the closest point at which the optima was known. In the implementation of golden section type algorithm, the step length is taken to be 0.2. The comparison is given in Figure 4.10 and it can be observed that the golden search is economical in terms of number of evaluations but model based approach solves more number of problems. The comparison is shown to illustrate that there are many different strategies that can be applied for optimizing $G(t)$. However, in this work, the model-based approach is used for optimizing $G(t)$ when comparing on the complete set of problems.

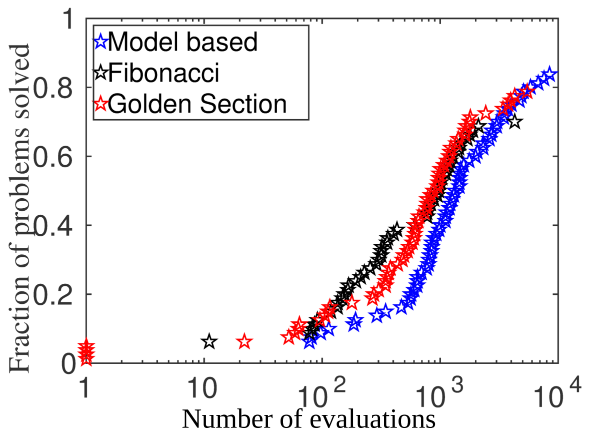


Figure 4.10: Comparison of model-based strategy, Golden section and Fibonacci algorithms for optimizing G. The strategies were compared on a set of 81 problems.

4.6 Conclusions

In this chapter, a novel approach for solving black-box problems with bound constraints is proposed. The concept of projecting multidimensional variables to a univariate space is proposed that is given by the linear combination of the variables. This leads to a point-to-set map because of the existence of multiple function values corresponding to the univariate variable. A function on this map is identified such that its minima is also the minima of the original function. This function is called the lower envelope. The UNIPOPT framework is developed to solve the multi-dimensional problem by identifying the lower envelope and minimizing it such that the obtained minima corresponds to the minima of the original problem. A two-step method, EPIC (Envelope PredIctor and Corrector) is proposed which finds discrete points on the lower envelope curve. The first step predicts the point by using approximating the Fiacco's Sensitivity theorem and the second step corrects the point given in the first step to obtain the exact point on the lower envelope. Once the lower envelope is obtained, it is then optimized using univariate surrogate model. Theoretical bounds on the difference between the solution given by exact and approximate sensitivity theorem is also provided. The convergence of UNIPOPT is also shown. The performance of the algorithm is compared with other model-based solvers on an extensive set of 161 convex nonsmooth and 232 nonconvex smooth problems. Overall, UNIPOPT performs better compared to other model-based approaches.

5. OPTIMIZATION OF CONSTRAINED BLACK-BOX PROBLEMS: A TWO-PHASE TRUST-REGION FRAMEWORK[1]

In Chapter 2 and Chapter 3, grey-box problems for which upper bounds of the diagonal elements of the Hessian were globally optimized. Chapter 4 further proposed a projection based algorithm for black-box problems with bound constraints. In addition to bound constraints, many practical engineering problems often comprises of black-box constraints and that makes the optimization problem even more challenging as illustrated in Figure 1.2. Therefore, in this chapter, black-box problems are addressed that can have constraints of both known and unknown algebraic forms. Specifically, an optimization algorithm is proposed that can handle BPKC, BPUC and BPHC. The form of the optimization problem addressed here can be represented as follows:

$$\begin{aligned} \min_{x} \quad & f(x) \\ \textit{s.t.} \quad & gu_i(x) \le 0 \quad \forall i \in \{1, \dots, p\} \\ & g_j(x) \le 0 \quad \forall j \in \{1, \dots, q\} \\ & x \in [x^L, x^U] \end{aligned} \tag{5.1}$$

where $x \in \mathbb{R}^n$ are the decision variables, $f(x) : \mathbb{R}^n \to \mathbb{R}$ and $g_i(x) : \mathbb{R}^n \to \mathbb{R}$, $i \in \{1, \dots, p\}$ are C^1 functions with hidden expressions in a black-box simulator. However, $g_j(x) : \mathbb{R}^n \to \mathbb{R}$, $i \in \{1, \dots, q\}$ are functions with known algebraic expressions.

In spite of the advances made in blackbox optimization for solving box-constrained or unconstrained problems, there are a lot of challenges that still need to be addressed for constrained problems. Firstly, many of the current available algorithms require a feasible initial guess. Secondly, many of the solvers treat all the constraints as black-box. However, in many practical applications, some of the constraints are known and critical problem insights are neglected by treating them as

[1]Reprinted with permission from I. Bajaj, S. S. Iyer, and M. M. F. Hasan, "A trust region-based two phase algorithm for constrained black-box and grey-box optimization with infeasible initial point" *Computers & Chemical Engineering*, vol. 116, pp. 306-321. Copyright 2018 Elsevier.

black-box. Thirdly, there is a need for an effective strategy to handle hard constraints that lead to simulation failure. Fourthly, finding global minima for nonconvex problems is also a challenge.

In this work, an optimization algorithm for constrained black-box problems is presented. The proposed algorithm has two phases: (i) feasibility phase, and (ii) optimization phase. The feasibility phase finds a feasible point while the optimization phase attempts to decrease the objective function. Both phases of the algorithm are based on developing and optimizing surrogate models in a trust region framework. The two phase algorithm does not require a feasible initial point. A novel optimization-based sampling strategy is also proposed which can handle hard constraints effectively so that the samples generated are feasible with respect to these constraints. The efficiency of the algorithm is demonstrated by applying on a set of 92 nonlinear test problems from GlobalLib and a chemical engineering case study. The algorithm is also compared to NOMAD [150, 184] and COBLYA [69], two widely used solvers for constrained black-box problems.

The chapter is organized as follows. The notations are the same as used in Chapter 4. Pertinent fundamental concepts are given in Section 5.1. Section 5.2 provides the details of the algorithm while Section 5.3 describes the numerical test problems and the chemical engineering case study. Some conclusions are made in Section 5.4.

5.1 Preliminaries

5.1.1 Trust Region Framework

In a trust-region method [173], a surrogate model is typically constructed in the neighborhood of a point x_k such that the model is believed to be an adequate representation of the original function in the region around x_k. The trust-region is defined to be the set of points around a point x_k such that:

$$B(x_k, \Delta_k) = \{x \in \mathbb{R}^n : ||x - x_k|| \leq \Delta_k\}$$

When the analytical form of the objective is known, the model Q is considered quadratic such that it incorporates the first and second order derivative information. The form of the model is as follows:

$$Q(x) = f(x_k) + \nabla^T f(x_k)x + x^T \nabla^2 f(x_k)x$$

A new point, $\bar{x}$ is obtained by solving a subproblem of the form:

$$\begin{aligned} \min_{x} \quad & Q(x) \\ \textit{s.t.} \quad & x \in B(x_k, \Delta_k) \end{aligned} \tag{5.2}$$

It is not even necessary to solve Problem 5.2 exactly but a point $\bar{x}$ providing sufficient decrease in the model is admissible. For unconstrained optimization problems with known derivative information of the objective function, the decisions about moving trust-region center and changing size are based on the ratio of actual reduction in the objective function to the predicted reduction, i.e.

$$\rho_k = \frac{f(x_k) - f(\bar{x})}{Q(x_k) - Q(\bar{x})} \tag{5.3}$$

Given the parameters $0 \leq \eta_0 \leq \eta_1 < 1$, $0 < \gamma_{de} < 1$ and $\gamma_{in} > 1$, the trust-region size is decided by the following rule:

$$\Delta_{k+1} = \begin{cases} \gamma_{in}\Delta_k, & \text{if } \rho_k \geq \eta_1 \\ \Delta_k, & \text{if } \eta_0 \leq \rho_k < \eta_1 \\ \gamma_{de}\Delta_k, & \text{if } \rho_k < \eta_0 \end{cases} \tag{5.4}$$

The trust-region center is updated as follows:

$$x_{k+1} = \begin{cases} \bar{x} & \text{if } \rho_k \geq \eta_0 \\ x_k, & \text{if } \rho_k < \eta_0 \end{cases} \tag{5.5}$$

The size of the trust-region is increased if ρ_k is appreciable and it is decreased otherwise. The algorithm converges to a second-order critical point (x^*) when at each iteration, (5.2) is solved approximately and the trust-region center and radius are updated as per the rules given by Eq.

(5.4) and Eq. (5.5). It is always possible in derivative-based methods to decrease the trust-region sufficiently and obtain a better objective function value unless the trust-region center is an optimal point. However, critical information is lost when the derivatives are unavailable. For optimization problems with the absence of first-order and second-order derivative information, input-output data are used to develop surrogate models. In BBO, the trust-region not only restricts the step length to the region where model is considered appropriate but also forms the basis for choosing samples to build data-driven models. In this case, the trust-region size is not the only factor that prevents us to obtain a better objective function. The models need to be sufficiently accurate before the decision to decrease the trust-region is taken. [175] formalized the concept of accuracy by defining fully linear and fully quadratic models. Ensuring fully quadratic models require the original function to be twice continuously differentiable function. In this work, the function is assumed to be once continuously differentiable and therefore focus on fully linear class of models.

Definition 5.1. *Assume that the function, $f(x)$ and its data-driven surrogate model, $f^r(x)$ are continuously differentiable and ∇f and ∇f^r are Lipschitz continuous. $f^r(x)$ is said to be fully linear if the error between the gradient of the model and the function is bounded such that*

$$||\nabla f(x) - \nabla f^r(x)|| \leq \kappa_{df}\Delta \quad \forall x \in B(x_k, \Delta) \tag{5.6}$$

And also, the error between the function and model satisfy the following relationship:

$$|f(x) - f^r(x)| \leq \kappa_f \Delta^2 \quad \forall x \in B(x_k, \Delta) \tag{5.7}$$

where the constants $\kappa_{df}, \kappa_f > 0$ are dependent on the Lipschitz constants of the surrogate model and the function and a constant Λ that measures the poisedness of the interpolation points [175]. Note that the parameters that can be controlled are, Λ and the trust-region size, Δ. The relations (5.6) and (5.7) indicate that the gradient of the function and the function itself can be approximated well if the trust-region size is small and the samples are well poised. If the sample set is well poised, the model is said to be fully linear. In other words, if the surrogate model is not able to obtain a better point, consideration should be given on constructing accurate models to estimate the direction of decrease. If the model is accurate and the trust-region is sufficiently small, a better objective function will be obtained. These bounds are similar to those of Taylor models and are essential to ensure convergence. The update rule for trust-region center remains the same but the criteria for updating the trust-region size is given as follows:

$$\Delta_{k+1} = \begin{cases} \gamma_{in}\Delta_k, & \text{if } \rho_k \geq \eta_1 \\ \Delta_k, & \text{if } \rho_k < \eta_1 \text{ and } f^r \text{ is not fully linear} \\ \gamma_{de}\Delta_k, & \text{if } \rho_k < \eta_1 \text{ and } f^r \text{ is fully linear} \end{cases} \tag{5.8}$$

For constrained black-box problems with black-box constraints, in addition to developing data-driven models for objective function, the models are also constructed for constraints. The surrogate models for the constraints are also ensured to satisfy the fully linear property, i.e.

$$\begin{aligned} &||\nabla g_i(x) - \nabla g_i^r(x)|| \leq \kappa_{dg_i}\Delta \quad \forall x \in B(x_k, \Delta) \\ &|g_i(x) - g_i^r(x)| \leq \kappa_{g_i}\Delta^2 \quad \forall x \in B(x_k, \Delta) \end{aligned} \tag{5.9}$$

5.2 Two Phase Algorithm

Next, an outline of the algorithm for solving Problem (5.1) is given and the detailed description of each of the components is also provided.

5.2.1 Outline of the Algorithm

The overall scheme is shown in Figure 5.1. The algorithm is designed such that the goal of finding a feasible point and the goal of decreasing the objective function are considered independently. The overall algorithm consists of two phases: the first is the feasibility phase that focuses on finding a feasible point and the second phase is termed as optimization phase that focuses on decreasing the objective function while maintaining feasibility at most of the iterations. One may argue that it might be more efficient to consider decreasing the objective function and infeasibility

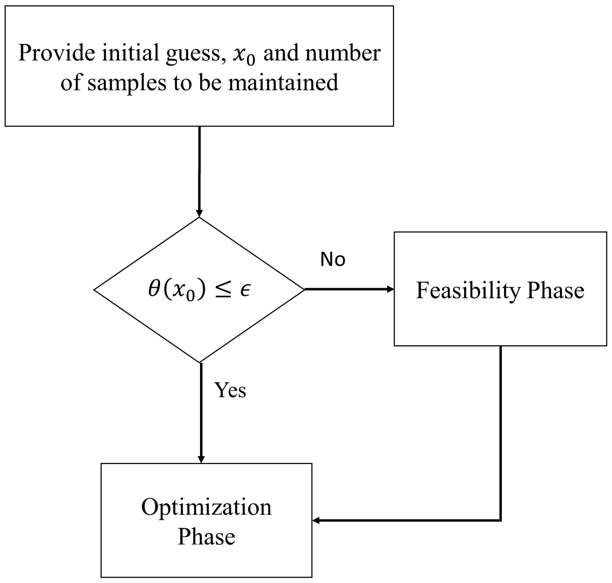

Provide initial guess, x_0 and number of samples to be maintained
$\theta(x_0) \le \epsilon$
No
Feasibility Phase
Yes
Optimization Phase

Figure 5.1: Schematic of the overall algorithm.

simultaneously as is done in penalty-based and filter methods. However, from a practitioner's point of view, it is preferred that when the optimization code stops, at least a feasible point is obtained. This is especially important when the problem is black-box and simulations are computationally expensive.

The basic idea of the optimization phase is to construct surrogate models of the objective function and the black-box constraints and solve the optimization subproblems. The updating rule for the objective function and the constraints is based on feasibility of the candidate point as well as the decrease in objective function. It is not necessary to construct fully linear models at each iteration but only when criticality measure is below a certain threshold so that Eq. (5.6), (5.7) and 5.9 provide meaningful bounds. Another instance when the construction of fully linear models become necessary is when the algorithm could not decrease the objective function or the constraint violation.

5.2.2 Feasibility Phase

The feasibility phase is initiated when the initial point provided by the user is infeasible. The degree of infeasibility is measured at a point by defining a smooth constraint violation function for unknown constraints as follows:

$$\theta(x) = \sum_{i=1}^{p} (max(0, g_i(x)))^2 \tag{5.10}$$

Note that $\theta = 0$ when a point is feasible and is positive otherwise. The optimization problem to be solved during feasibility phase is defined as follows:

$$\begin{aligned} \min_{x} \quad & \theta(x) \\ \textit{s.t.} \quad & g_j(x) \leq 0 \quad \forall j \in \{1, \ldots, q\} \\ & x \in [x^L, x^U] \end{aligned} \tag{5.11}$$

Problem (5.11) is a black-box problem with known constraints.

The feasibility problem is a box-constrained problem with black-box objective when there are no known constraints. It becomes a black-box problem subject to known constraints when some of the constraints are known. Both of these problems are relatively easy to solve. A trust-region framework is adopted to solve feasibility problem. The decision regarding updating of trust-region center and size are similar to that of unconstrained minimization of black-box problems. For the case when the closed form of all the constraints is known, the sampling technique used in the chapter always ensure the feasibility of samples. The overall algorithm for the Feasibility phase (FPA) is given in Algorithm 5.1. The algorithm initiates by considering the entire space as the initial trust-region and obtaining $m + 1$ feasible samples with respect to known constraints. The reason for considering the entire space initially is explore the problem globally and thereby increasing the chances of obtaining global minima. The algorithm allows the user to provide an initial point enabling them to exploit the systems knowledge. The function evaluation is then performed on the samples and the constraint violation values are collected. A surrogate model for the constraint violation, θ^r is then developed using the samples data and optimized using ANTIGONE [103] to obtain a candidate point, x^{fop} by solving:

$$\begin{aligned} \min_{x} \quad & \theta^r(x) \\ \textit{s.t.} \quad & g_j(x) \leq 0 \qquad \forall j \in \{1, \ldots, q\} \\ & ||x - x_k|| \in \Delta_k \end{aligned} \tag{5.12}$$

The constraint violation is calculated at x^{fop} and is defined as $\theta(x^{fop})$. If $\theta(x^{fop})$ is less than a pre-specified tolerance, θ_{al}, a feasible point is obtained and the algorithm goes to the optimization phase. If the constraint violation is not below the tolerance, it is then checked if a decrease in the constraint violation is obtained compared to the current iterate. If this is the case, the new trust-region center is updated to x^{fop}. The interpolation point which is distant from x^{fop} and damaging the geometry of the sample set is replaced. The first term on the right hand side of Eq. (5.14) measures the distance and the second measures poisedness. The interpolation set is updated by removing a point that maximizes the combined criteria of distance and poisedness. It is possible that Problem (5.11) has a local minima at an infeasible point and in order to give an indication to the user to try with a new point, it is important to check if a point x^{fop} is locally optimal. This is done by the comparing criticality measure with the size of the trust-region. The criticality measure for the feasibility problem is given by:

$$\begin{aligned} \psi^{\theta}(x^{fop}) = \min_{d} \quad & \nabla\theta^r(x^{fop})d \\ \textit{s.t.} \quad & g_j(x^{fop}) + \nabla g_j(x^{fop})^T d \leq 0 \quad \forall j \in \{1, \ldots, q\} \\ & ||d|| \leq 1 \end{aligned} \tag{5.13}$$

If both trust-region size and criticality measure is small, x^{fop} is an optimal point for problem (5.11). If the criticality measure, ψ^{θ} at x^{fop} is smaller than some tolerance ($\epsilon_{\psi^{\theta}}$), it is possible that x^{fop} is an optimal point of Problem (5.11). However, bounds given by Eq 5.9 is ineffective unless the trust-region is small. Therefore, the size of the trust-region is checked if it is less than some threshold value (ϵ_{Δ}). In that case, a local minima is achieved but is an infeasible point and the algorithm should be started with a different initial point. If the trust-region size is not below a threshold, the criticality measure gives an indication of proximity to the optima and it is beneficial to decrease the trust-region. Before decreasing the trust-region, it is checked if the surrogate models are fully linear.

If the criticality measure is above ϵ_{ψ^θ}, a parameter defining the ratio of reduction in the constraint violation to the predicted reduction by the surrogate model, ρ_k^θ is calculated. If ρ_k^θ is above a certain threshold value (η_0), this indicates that the model is sufficiently accurate within the current trust-region size and that the trust-region size can be further increased to allow for larger step length. The trust-region is only decreased if a decrease in the constraint violation is not obtained and the model is fully linear. Another possibility of not being able to obtain a decrease in constraint violation could be due to the inaccuracy of the surrogate model. When this happens, a model improvement step is called to construct a fully linear model. The model improvement algorithm (MIA) is discussed in Section 5.2.4 in detail.

5.2.3 Optimization Phase

Once a feasible point is obtained, optimization phase is triggered. The feasible point obtained in the feasibility phase is included in the initial set of samples for the optimization phase. This allows the surrogate models in the optimization phase to have the information about at least one point in the feasible region. With this information, the optimization phase then focuses on finding points with better objective function values in the feasible region. This phase attempts to find a non-increasing sequence of feasible iterates which converges to at least a local minima. Similar to the feasibility phase, the optimization phase considers m interpolation points in the entire space to explore the global features of the problem. A feasible point x_0^f is also provided as one of the initial samples to give an indication to the model about the feasible region. The pseudocode of the optimization phase is given in Algorithm 5.2. It is similar in spirit to the feasibility phase except for the rule to update the trust-region center. After performing the function evaluation at the sample set, Y_k, surrogate models are constructed for the objective function ($f^r(x)$) and the constraints ($g_i^r(x)$) with unknown closed forms. The following problem is then solved to obtain a candidate

Algorithm 5.1 Feasibility Phase Algorithm (FPA)

```
STEP 0: Choose initial $m$ interpolation points, initial guess ($x_0$), initial trust-region radius $\Delta_0 = \Delta_{max} = ||x^U - x^L||$, $\eta_0$, $\gamma_{de}$, $\gamma_{in}$, $k = 0$, $\epsilon_\Delta$, $\epsilon_{\psi^\theta}$ and $\theta_{al}$.
STEP 1: Solve Eq. (4.28a) to construct the surrogate model, $\theta^r$ for constraint violation, $\theta$ and solve (5.12) to obtain $x^{fop}$.
STEP 2: Update the next iterate, trust-region and interpolation set. Define
    $$y_k^o = arg\max_{y^j \in Y_k} ||y^j - x^{fop}||^2 |l_j(x^{fop})| \quad (5.14)$$
if ($\theta(x^{fop}) \le \theta_{al}$) then, STOP and return $x^{fop}$
end if
if ($\theta(x^{fop}) < \theta(x_k)$) then, $x_{k+1} = x^{fop}$ and $Y_{k+1} = Y_k \backslash y_k^o \cup x^{fop}$
end if
if ($\psi^\theta(x^{fop}) \le \epsilon_{\psi^\theta}$) then,
    if ($\Delta_k \le \epsilon_\Delta$) then, STOP, no feasible point found. Restart from a different initial point.
    else
        if (model is fully linear) then, $\Delta_{k+1} = \gamma_{de}\Delta_k$
        else $\Delta_{k+1} = \Delta_k$ and initiate MIA
        end if
    end if
else Calculate $\rho_k^\theta = \frac{\theta(x^{fop}) - \theta(x_k)}{\theta^r(x^{fop}) - \theta^r(x_k)}$
    if ($\rho_k^\theta \ge \eta_0$) then, $\Delta_{k+1} = \gamma_{in}\Delta_k$
    else if ($\rho_k^\theta < \eta_0$ and $\theta(x^{fop}) < \theta(x_k)$) then, $\Delta_{k+1} = \Delta_k$
    else if ($\theta(x^{fop}) \ge \theta(x_k)$ and the model is not fully linear) then, $\Delta_{k+1} = \Delta_k$ and initiate MIA and replace at most 2 sampling points
    else if ($\theta(x^{fop}) \ge \theta(x_k)$ and the model is fully linear) then, $\Delta_{k+1} = \gamma_{de}\Delta_k$
    end if
end if
Step 3: Increment $k$ by one and go to Step 1.
```

point, x^{sop}:

$$\begin{aligned} \min_x \quad & f^r(x) \\ s.t. \quad & g_i^r(x) \le 0 \qquad \forall i \in \{1, \ldots, p\} \\ & g_j(x) \le 0 \qquad \forall j \in \{1, \ldots, q\} \\ & ||x - x_k|| \in \Delta_k \end{aligned} \quad (5.15)$$

where g_i^r is the surrogate model for the ith constraints whose form is unknown. The constraint violation at x^{sop} is calculated and checked if it is less than a pre-specified tolerance (θ_{tol}). If it is less, it is examined if a decrease in the objective function is obtained. If that is the case, then the trust-region center is updated to x^{sop}. The interpolation set is also updated by including x^{sop} and discarding the worst point. To determine if x^{sop} is optimal, the criticality measure ($\psi^f(x^{sop})$) is calculated and examined if it is less than some tolerance (ϵ_{ψ^f}). The criticality measure corresponding to the optimization phase is calculated by:

$$\begin{aligned} \psi^f(x^{sop}) = \min_d \quad & \nabla f^r(x^{sop})d \\ \textit{s.t.} \quad & g_i^r(x^{sop}) + \nabla g_i^r(x^{sop})^T d \leq 0 \quad \forall i \in \{1, \ldots, p\} \\ & g_j(x^{sop}) + \nabla g_j(x^{sop})^T d \leq 0 \quad \forall i \in \{1, \ldots, q\} \\ & ||d|| \leq 1 \end{aligned} \tag{5.16}$$

The criteria about the trust-region size is also verified before giving the certificate of optimality. The criticality measure can be small due to inaccuracy of the surrogate models. Therefore, before the trust-region is decreased, the fully linear property of the model is ensured to hold true. If the criticality measure is above the threshold (ϵ_{ψ^f}), the ratio of actual reduction in the objective function to the predicted reduction given by the surrogate model is calculated. Note that it is not reasonable to assume the predicted reduction to be always positive as it happens in classic derivative- based problems. The reason for predicted reduction being negative can be due to inaccuracy of the surrogate models. Therefore, it is ensured that there is an actual reduction in the objective function before the the trust-region size is increased. And it is kept constant if the ratio, ρ_k^f is less. The trust-region size is decreased when the model is fully linear and there is no improvement in the objective function.

In many problem instances, it may happen that the objective function can be decreased while maintaining feasibility only along a thin domain. In extreme cases, such a thin domain can be an equality constraint. It is not trivial to obtain a search direction when the derivatives are not available. NOWPAC and CONORBIT addressed this issue by adding a small offset in problem (5.15). Once it is ensured that the model is fully linear, the trust-region size is decreased. The filter approach e.g., [76] accepts infeasible point if the new point reduces the objective function sufficiently compared to the constraint violation. The criticality measure is compared to the trust-region size to verify if it is indeed the case of thin domain. If the criticality measure is more than a fraction of the trust-region size, the tolerance on the allowable constraint violation is increased so that slightly infeasible solution is admissible. Once a better objective function is obtained, the tolerance on the constraint violation is reset to the original value. It is possible that problem (5.15) becomes infeasible at certain iterations. This is due to finite tolerance on constraint violation, all the samples may be slightly infeasible. When this happens, the following subproblem is solved:

$$
\begin{aligned}
\min_{x} \quad & f^r(x) \\
\textit{s.t.} \quad & g_i^r(x) \leq g_i(x_k) && \forall i \in \{1, \dots, p\} \\
& g_j(x) \leq 0 && \forall j \in \{1, \dots, q\} \\
& ||x - x_k|| \in \Delta_k
\end{aligned}
\tag{5.17}
$$

At this point, it should be pointed out that the tolerance on constraint violation can be selected as 0 but that may slow the progression of the algorithm.

5.2.4 Model Improvement

It is known that to ensure the surrogate models to satisfy fully linear property, the set of interpolating points need to be well-poised and is very critical to ensure convergence of BBO algorithms [54]. In broad terms, the meaning of well-poised samples is that their geometry is adequate in the sense that it should span all the directions of the space. It is possible to choose an initial set of samples which are well-poised. The models need to be ensured to be fully linear. A very simple but crude way for doing so is to replace all the samples at each iteration such that all of them lie in the trust-region, $B(x_k, \Delta_k)$. However, this approach is not efficient and most likely will result in large number of evaluations. Therefore, assuming that the initial set of interpolating samples are well-poised, a mechanism is needed that replaces one sample at each iteration while improving the geometry of the samples. Also, note that the models are not required to be fully-linear at all

Algorithm 5.2 Optimization Phase Algorithm (OPA)

```
STEP 0: Choose initial $m$ interpolation points, feasible point obtained from feasibility phase ($x_0{}^f$), initial trust-region radius $\Delta_0 = \Delta_{max} = ||x^U - x^L||$, $\eta_0$, $\gamma_{de}$, $\gamma_{in}$, $k = 0$, $\epsilon_\Delta$, $\epsilon_{\psi^f}$ and $\theta_{al}$, $\theta_{tol} = \theta_{al}$, $\nu < 1$.
STEP 1: Solve Eq. (4.28a) to construct surrogate model, $f^r$ for the objective function and set of surrogate models, $g_i^r \ \forall i \in \{1, \cdots, p\}$ to approximate unknown constraints. Solve Problem (5.15) to obtain $x^{sop}$.
if (Problem (5.15) is infeasible) then, solve Problem (5.17) to obtain $x^{sop}$
    if ($\psi^f(x^{sop}) > \nu\Delta_k$) then, Replace all the interpolation point by Eq. (??)
    end if
end if
STEP 2: Update the next iterate, trust-region and interpolation set.
$y_k^o = arg \max_{y^j \in Y_k} ||y^j - x^{sop}||^2 |l_j(x^{sop})|$
if ($\theta(x^{sop}) > \theta_{tol}$) then,
    if (Model is fully linear) then, $\Delta_{k+1} = \gamma_{de}\Delta_k$
        if ($\psi^f(x^{sop}) > \nu\Delta_k$) then, $\theta_{tol} = 10\theta_{al}$
        end if
    else Initiate MIA
    end if
else
    if ($f(x^{sop}) < f(x_k)$) then, set $\theta_{tol} = \theta_{al}$, $x_{k+1} = x^{sop}$, $Y_{k+1} = Y_k \backslash y_k^o \cup x^{sop}$
    end if
    if ($\psi^f(x^{sop}) \leq \epsilon_{\psi^f}$) then,
        if ($\Delta_k \leq \epsilon_\Delta$) then, STOP!
        else
            if (Model is fully linear) then, $\Delta_{k+1} = \gamma_{de}\Delta_k$
            else $\Delta_{k+1} = \Delta_k$ and initiate MIA
            end if
        end if
    else Calculate $\rho_k^f = \frac{f(x^{sop}) - f(x_k)}{f^r(x^{fop}) - f^r(x_k)}$
        if ($\rho_k^f \geq \eta_0$ and $f(x^{sop}) < f(x_k)$) then, $\Delta_{k+1} = \gamma_{in}\Delta_k$
        else if ($\rho_k^f < \eta_0$ and $f(x^{sop}) < f(x_k)$) then, $\Delta_{k+1} = \Delta_k$
        else if ($f(x^{sop}) \geq f(x_k)$ and the model is not fully linear) then, $\Delta_{k+1} = \Delta_k$ and initiate MIA
        else if ($f(x^{sop}) \geq f(x_k)$ and the model is fully linear) then, $\Delta_{k+1} = \gamma_{de}\Delta_k$
        end if
    end if
end if
Step 3: Increment $k$ by one and go to Step 1.
```

iterations but only when the objective function is not reduced or the criticality measure is below a certain threshold.

To this end, the self-correcting geometry approach proposed by [185] is employed. The idea of the method is to replace an existing point in the interpolation set, Y_k by either x^{fop} or x^{sop} if the algorithm is in the feasibility phase or the optimization phase, respectively.

In the literature, one of the metrics of well-poisedness is given by Lagrange polynomials [186]. If a set of interpolation points $Y_k = \{y_0, y_1, \ldots, y_m\}$ is defined, correspondingly a basis of polynomials $l_i(x), i = 0, 1, \ldots, m$ is basis for Lagrange polynomials such that:

$$l_i(y_j) = \delta_{ij} \tag{5.18}$$

where δ_{ij} is the Kronecker delta. Clearly, if the degree of the Lagrange polynomials is 2, the number of interpolation points required to uniquely determine its coefficients is $m^Q = \frac{(n+1)(n+2)}{2}$. But, this is not favorable when the computational budget is limited. In this chapter, the number of interpolation points that is less than m^Q are maintained. To compute the coefficients of the Lagrange polynomials, a convex optimization problem that minimizes the Hessian of the Lagrange polynomials is solved:

$$\begin{aligned} &\min \ ||\nabla^2 l_i|| \\ &\textit{s.t.} \quad l_i(y^j) = \delta_{ij} \end{aligned} \tag{5.19}$$

A set of interpolation points is said to be Λ-poised if the corresponding Lagrange polynomials are bounded, i.e.

$$\max_{i=0,1,\ldots,m} \ \max_{x \in B} \ |l_i(x)| \leq \Lambda \tag{5.20}$$

The criteria used to replace a point is based on combined poisedness and distance measure such that on replacement, the geometry of the samples are improved. The model improvement algorithm is called by either the optimization phase or the feasibility phase whenever the geometry of the samples need to be improved. A pseudocode for MIA is given in Algorithm 5.3.

Firstly, the set of points lying outside the current trust-region are identified denoted by $\mathbb{F}_k$. The point which maximizes the distance and poisedness measure is replaced. If $\mathbb{F}_k$ is empty, a set of non-poised close points, $\mathbb{C}_k$ is identified. Again, a point that maximizes the combined distance and poisedness criteria is identified. If both the sets $\mathbb{F}_k$ and $\mathbb{C}_k$ are empty, then the current interpolation set is well-poised and the surrogate model is fully linear.

Algorithm 5.3 Model Improvement Algorithm (MIA)

```
STEP 0: Choose parameters $\beta \geq 1$, $\Lambda > 1$ and given a candidate point $x^{op}$.
STEP 1: Define a set consisting of distant interpolation points such that:
        $$\mathbb{F}_k = \{y^j \in Y_k| \ ||y^j - x^{op}|| > \beta\Delta_k \text{ and } |l_{k,j}| \neq 0\} \quad (5.21)$$
if ($\mathbb{F}_k \neq \emptyset$) then, amongst the farthest points, replace a point $y^r \in \mathbb{F}_k$ such that
        $$y^r \in arg\max_{y^j \in \mathbb{F}_k} ||y^j - x^{op}||^2 |l_j(x^{op})| \quad (5.22)$$
Define the new interpolation set, $Y_{k+1} = Y_k \setminus \{y^r\} \cup \{x^{op}\}$
else if ($\mathbb{F}_k = \emptyset$) then  Define a set consisting of close interpolation points such that:
        $$\mathbb{C}_k = \{y^j \in Y_k \setminus \{x_k\}| \ ||y^j - x^{op}|| \leq \beta\Delta_k \text{ and } |l_{k,j}(x^{op})| > \Lambda\} \quad (5.23)$$
    if ($\mathbb{C}_k \neq \emptyset$) then, Amongst the closest point, replace a point $y^r \in \mathbb{C}_k$
        $$y^r \in arg\max_{y^j \in \mathbb{C}_k} ||y^j - x^{op}||^2 |l_j(x^{op})| \quad (5.24)$$
Define the new interpolation set, $Y_{k+1} = Y_k \setminus \{y^r\} \cup \{x^{op}\}$
    else  The model is fully linear
    end if
end if
```

5.2.5 Surrogate Modeling

As discussed earlier, surrogate models θ^r, f^r and g_i^r are developed. Surrogate model is an inexpensive approximation of the unknown equations that enables one to utilize derivative-based algorithms to guide the search towards the true optimum of the problem. The surrogate model serves the purpose of implicitly approximating derivative of the underlying function. The surrogate models considered are interpolation or regression based. Certain surrogate model might be well suited for some problems while another surrogate model might be more appropriate for other problems. The surrogate models that satisfy the fully linear property are of prime interest. Quadratic ([90], [54], [73], [172]) and radial basis function ([52], [174], [75]) are some of the commonly used surrogate models to approximate the original function. The surrogate models are built for the constraint violation function during feasibility phase and for the objective function and unknown constraints during optimization phase. Similar to Chapter 4, cubic radial basis function (CRBF) is used as the surrogate model and the details of the model are given in Section 4.3.2.4.

5.2.6 Initial Sampling

In many of black-box optimization problems, the simulation is computationally expensive. In this scenario, it is critical that initial set of samples to be selected spans the entire space efficiently. Several sampling schemes are available in the literature and interested reader may refer a recent review paper by [100] to learn about various static and adaptive sampling schemes. Two alternate approaches for sampling are used depending on the constraints of the problem. If all the constraints of the problem are black-box, then the approach of [90] is used as the first sampling strategy and given by Eq. (4.25).

In many engineering applications, all the constraints may not be completely black-box and may have a subset of known constraints. It is beneficial to take the advantage of known constraints by sampling only in the feasible region [187]. The known constraints may be unrelaxable which means that the simulation fails at certain samples that do not satisfy the unrelaxable constraints. [78] suggested a two step strategy to obtain feasible samples. The first step involved obtaining a large number of samples using Latin Hypercube design (LHD) and filtering out the ones which are infeasible with respect to known constraints. In the next step, distance criteria is used to select the final set of samples. A key issue with this approach is that it is difficult to determine appropriate number of samples to be given by LHD *a priori*.

Therefore, in the second sampling strategy, an optimization based sampling strategy is proposed such that a desired number of samples, which are space filling and feasible with respect to known constraints, is obtained. Optimization formulations based on entropy, integrated mean squared error, minimax and maximin distances and discrepancies can be used as metric to measure space filling characteristics. The wrap-around L^2 discrepancy [188] is minimized and the known constraints are incorporated resulting in the following nonlinear problem:

$$\begin{aligned} \min_{u} \quad & \left(\frac{4}{3}\right)^n + \frac{1}{m^2}\sum_{j=1}^{m}\sum_{j'=1}^{m}\prod_{i=1}^{n}\left[\frac{3}{2} - |u_i^j - u_i^{j'}|(1 - |u_i^j - u_i^{j'}|)\right] \\ \textit{s.t.} \quad & g_j(u) \leq 0 \qquad \forall j \in \{1, \ldots, q\} \\ & ||u - u_k|| \in \hat{\Delta}_k \end{aligned} \tag{5.25}$$

Here, u_i^j is the scaled ith component of jth sample. Both of the sampling strategies are deterministic in nature and the only parameter that will change the output of the algorithm is the initial guess. Therefore, the proposed strategy is deterministic in nature as long as the initial guess remains the same.

While the filter approach of [76] considers decreasing objective function and constraint violation simultaneously by building a pareto front, the two aspects are considered independently. It is sometimes not possible to get a point which is feasible and improves the objective function at the same time especially when the feasible domain is thin. The filter approach of [76] deals with this problem by accepting an infeasible point if it decreases the objective function sufficiently. However, in the proposed approach, the tolerance on constraint violation is increased by checking if criticality measure is above a threshold, size of trust-region is below a threshold and the model is fully linear. The algorithm proposed in this chapter is similar in spirit to NOWPAC and CONORBIT in the sense that it is neither based on penalty or filter approach. A key difference, however is that no assumption on the availability of an initial feasible point. Secondly, a margin or offset is not used in the constraints. NOWPAC and CONORBIT also assume that all the constraints are black-box and relaxable. It is possible in many cases that the simulation fails if the samples do not satisfy certain constraints. The proposed blackbox optimization framework can handle these constraints efficiently.

5.3 Computational Studies

In this section, computational results for the two phase algorithm are provided. After demonstrating the algorithm on a nonlinear 2 dimensional example, comparison of the performance of the proposed method is also provided. Moreover, the two-phase algorithm is also compared to COBYLA and NOMAD. The details of the algorithm parameters with their values used is provided in Table 5.1. The application of the algorithm is also demonstrated on a chemical engineering case study.

5.3.1 An Illustrative Example

The two phase algorithm is illustrated using a two-dimensional nonlinear smooth problem, `st_e18`. The problem can be stated as follows:

$$\begin{aligned} \min_{x} \quad & x_1 + x_2 \\ \textit{s.t.} - & x_1^2 - x_2^2 \leq -1 \\ & x_1^2 + x_2^2 \leq 4 \\ & -x_1 + x_2 \leq 1 \\ & x_1 - x_2 \leq 1 \\ & x_1 \in [-2, 2] \\ & x_2 \in [-2, 2] \end{aligned} \tag{5.26}$$

In this numerical problem, although the analytical functional form is known, the derivative information is not utilized and only simulations can be performed. The variable bounds are assumed to be known and only the nonlinear constraints are considered black-box.

The algorithm starts by providing the lower bound, i.e., $(-2, -2)$ as the initial guess. Clearly, the initial guess is infeasible and feasibility phase is therefore initiated. Initial samples generated at iteration 0 using Eq. 4.25 and the corresponding constraint violation values are listed in Table 5.2. None of the initial samples are feasible. Figure 5.2 shows the behavior of the constraint

Table 5.1: Algorithm parameters.

Parameters	Values
Threshold for changing the trust-region size (η_0)	0.1
Factor of decreasing the trust-region size (γ_{de})	0.5
Factor of increasing the trust-region size (γ_{in})	3
Minimum trust-region size (ϵ_Δ)	10^{-6}
Tolerance on criticality measure for feasibility problem (ϵ_{ψ^θ})	10^{-3}
Allowable constraint violation for the final solution (θ_{al})	10^{-8}
Comparison factor of trust-region size and criticality measure (ν)	0.1
Factor indicating a particular point to be farther compared to the trust-region size (β)	1.2
Threshold on poisedness measure (λ)	2

Table 5.2: Samples and corresponding constraint violation at iteration 0 of feasibility phase.

Sample no.	Sample	θ
1	(-2,-2)	16
2	(2,-2)	25
3	(-2,2)	25
4	(0,-2)	1
5	(-2,0)	1

violation function. Note that the feasible region is small and a systematic method is needed to obtain a feasible point. In the next step, surrogate model for the scaled constraint violation is developed and the model parameters are estimated by solving Eq. (4.28a). The functional form of the obtained surrogate model is:

$$\begin{aligned} f^r(x) = &-1.185x_1 - 1.185x_2 + 2.278\Big(\sqrt{(x_1)^2+(x_2)^2}\Big)^3 + 0.427\Big(\sqrt{(x_1-1)^2+(x_2)^2}\Big)^3 + \\ &0.427\Big(\sqrt{(x_1)^2+(x_2-1)^2}\Big)^3 - 0.853\Big(\sqrt{(x_1-0.5)^2+(x_2)^2}\Big)^3 - \\ &0.853\Big(\sqrt{(x_1)^2+(x_2-0.5)^2}\Big)^3 \end{aligned} \tag{5.27}$$

The above surrogate model is then optimized and the sample (0.4199,0.4199) is obtained. When the function is evaluated at this point, there is improvement in the constraint violation. The interpolation point which maximizes the combined distance and poisedness measure is replaced

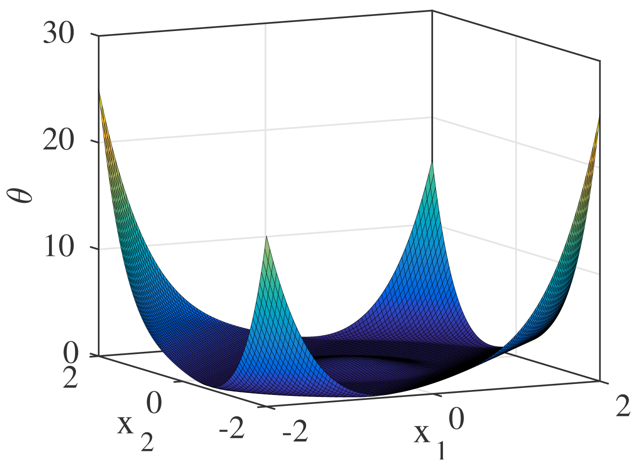


Figure 5.2: Constraint violation function for `st_e18`.

Table 5.3: Samples and corresponding Lagrange polynomials at iteration 0.

Sample	Lagrange Polynomials (l)
(-2,-2)	$-1 - 0.25x_1 - 0.25x_2 + 0.5(0.25x_1^2 + 0.25x_2^2)$
(2,-2)	$0.25x_1 + 0.5(0.25x_1^2)$
(-2,2)	$0.25x_2 + 0.5(0.25x_2^2)$
(0,-2)	$1 + 0.5(-0.5x_1^2)$
(-2,0)	$1 + 0.5(-0.5x_2^2)$

by the new point. The Lagrange polynomials used to measure the poisedness is listed in Table 5.3. Note that the Lagrange polynomials do not depend on the objective function values but rather only on set of interpolation points. The criticality measure at this point is 1.426×10^{-9} which is less than the pre-specified tolerance, ϵ_{ψ^θ} but the size of the trust-region is more than the tolerance, ϵ_Δ. Repeating the steps as listed in Algorithm 5.1, a feasible point is obtained in 8 iterations and requires a total of 13 function evaluations. Figure 5.3(a) shows the movement of iterates during the feasibility phase. The feasibility phase starts with an infeasible point and ultimately gives a feasible solution Figure 5.4(a) gives the progress of the iterates with cumulative number of evaluations.

Figure 5.3 shows the movement of the iterates during the optimization phase. It starts with the objective value of -1.5897 and finally obtains an objective function value of -2.823. The global minima of the problem is -2.8284. The progress of minimum objective function value with number

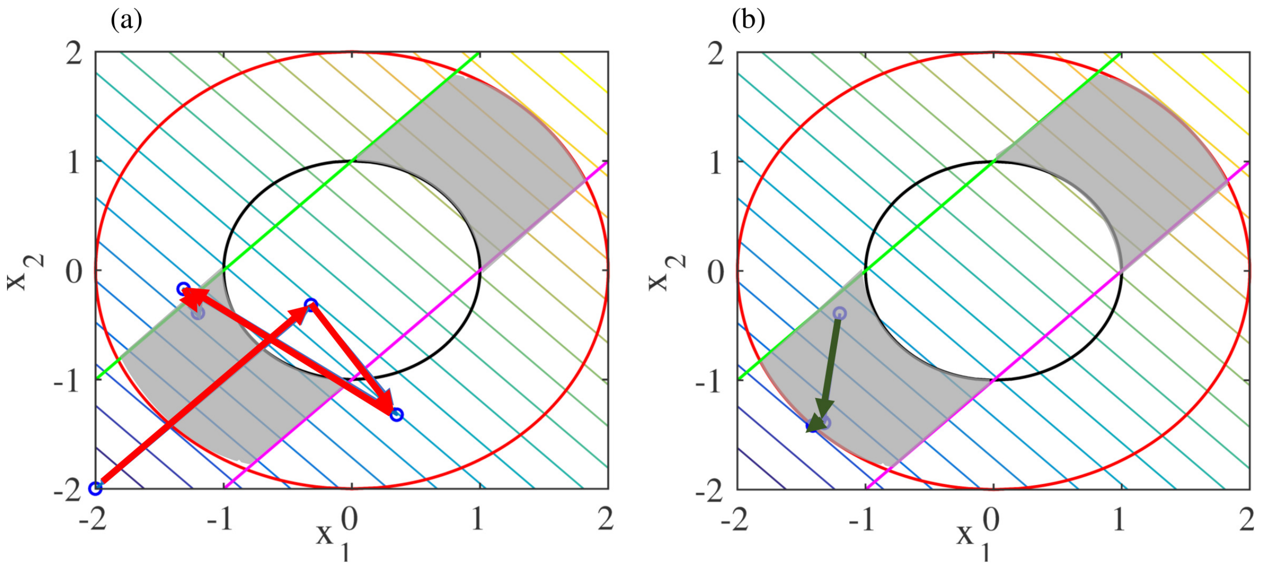


Figure 5.3: The figure shows the contour plot of the objective function and the feasible region is shown by the grey region. It also illustrates the progression of the iterates in (a) feasibility phase and (b) optimization phase.

of evaluations during the optimization phase is given in Figure 5.4(b). The algorithm is able to obtain a solution which is within 1% of the global minima in 24 function evaluations.

5.3.2 GlobalLib Test Problems

The two-phase approach is applied to a comprehensive list of 92 test problems from GlobalLib [189] for which the global minima are known. The test problems are the same as those used by [78]. All the problems have only inequality constraints. The most difficult problem are solved by assuming that all the constraints have unknown analytical functional form. The dimensions of these 92 problems vary from 1 to 65 while the number of constraints range from 1 to 81. The distribution of the problems with dimension and constraints is given in Figure 5.5. For the problems which did not have lower and upper bounds for the variables given in the problem formulation, the lower and upper bounds are assigned as follows:

$$\begin{aligned} x_i^L &= x_i^{glob} - 10 \quad \forall i \in \{1, \dots, n\} \\ x_i^U &= x_i^{glob} + 10 \quad \forall i \in \{1, \dots, n\} \end{aligned} \tag{5.28}$$

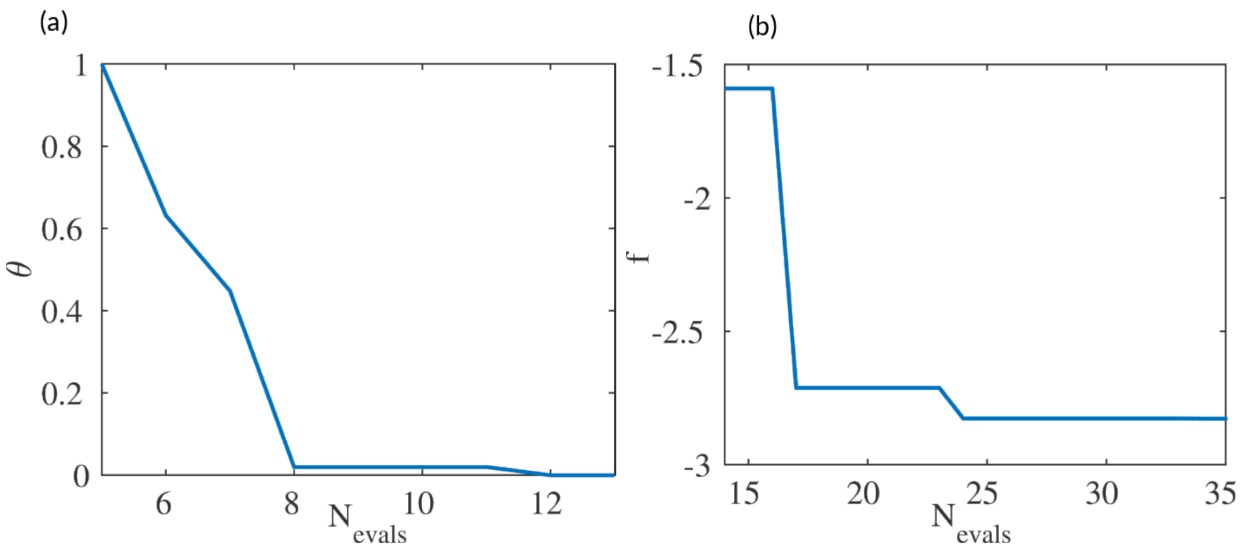


Figure 5.4: Decreasing (a) constraint violation, and (b) objective function with number of evaluations

It is assumed that a single function evaluation gives the values of the objective function and all the constraints. For all the problems, lower bound on the variables is provided as the initial guess. Table 1 in the supplementary material lists each of the problem's dimension, number of constraints and whether the initial point is feasible or infeasible. The algorithm is implemented in MATLAB while GAMS/ANTIGONE is used to solve the surrogate optimization problems. At each iteration, $2n + 1$ samples are maintained for developing surrogate models.

In derivative-based solvers, the performance measure is generally the time taken by a solver s to solve a problem p. The computational expense required by a blackbox solver includes the simulation time and the time required to solve all the subproblems. The numerical examples used in this chapter require negligible computation cost for simulation and therefore, computation time is not an appropriate metric. The simulation cost can be much higher in other applications and therefore an appropriate metric to measure the performance is the number of samples required to satisfy a convergence test. Two major approaches to compare the performance of blackbox optimizations have been proposed in the literature [190]. One of the approaches is to use a performance profile. It compares solvers by defining the performance ratio for a solver. Let the set of test problems

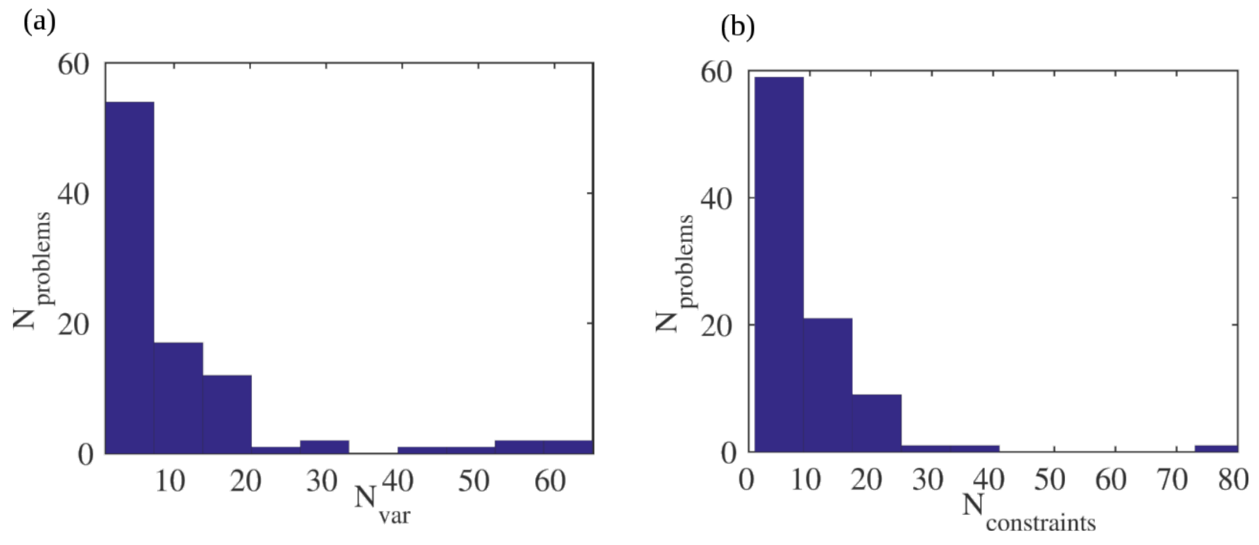


Figure 5.5: Distribution of problems with (a) number of variables, and (b) number of constraints.

be $\mathbb{T}$, set of solvers/algorithms be $\mathbb{S}$ and the performance measure be $N_{t,s}$. To get a performance profile, the performance ratio is defined as follows:

$$r_{t,s} = \frac{N_{t,s}}{\min\limits_{s \in \mathbb{S}} N_{t,s}} \tag{5.29}$$

The ratio, $r_{t,s}$ is 1 for the best solver and greater than 1 for worse solvers. The performance profile for a solver is defined as the fraction of problems for which the performance ratio is at most α:

$$\varrho_s(\alpha) = \frac{1}{|\mathbb{T}|} \text{size}\{t \in \mathbb{T} : r_{t,s} \leq \alpha\} \tag{5.30}$$

where $|\mathbb{T}|$ denotes the cardinality of $\mathbb{T}$. $\varrho_s(\alpha)$ represents the fraction of problems that has performance ratio less than α. A solver with higher value of $\varrho_s(\alpha)$ is favorable compared to the lower values.

While the performance profile provides a ranking of the solvers relative to each other, it does not give the complete information about the fraction of problems solved with the number of function evaluations by an individual solver. A data profile considers the absolute performance of a solver and determines its ability to solve an optimization problem. Data profile represents the fraction of problems solved within α function evaluation and defined as follows:

$$d_s(\alpha) = \frac{1}{|\mathbb{T}|}\text{size}\{t \in \mathbb{T} : N_{t,s} \leq \alpha\} \tag{5.31}$$

The comparison is shown using both performance and data profiles. The convergence test used to certify a problem to be solved is also critical for the analysis of different algorithms. Two convergence criteria are examined in the computational comparison. The first criterion is to compare the absolute merit of the solution. One of the approaches that can be used is to compare the accuracy with the global solution as done by [37]. Therefore, one of the convergence test used in this study is that a solver is said to have solved a problem if the constraint violation is less than 10^{-8} and the solution is within 1% of the global solution. A solver is able to solve a problem if a point $\hat{x}$ is obtained by the solver that satisfies:

$$\theta(\hat{x}) <= 10^{-8} \quad \& \quad f(\hat{x}) \leq \max(1.01 f(x^*), f(x^*) + 0.01) \tag{5.32}$$

where x^* is the global solution. The above criterion is often used when a user is interested in the accuracy of the solution. However, when the computational budget of the user is limited, the ability of a solver to improve an initial point can also be of interest. For an unconstrained problem, the criteria is defined as follows:

$$f(\hat{x}) \leq f(x^*) + \tau(f(x^0) - f(x^*)) \tag{5.33}$$

This criterion can be interpreted as follows. If a solver is able to reduce the objective function by at least $1 - \tau$ times the maximum reduction possible. The smaller value of τ implies that a solution closer to global minima is desired.. For constrained problems, the ability to improve the constraint violation corresponding to the initial point is critical. Therefore, a merit function is defined that incorporates the constraint violation information:

$$\varphi(x) = f(x) + \upsilon\theta(x) \tag{5.34}$$

In this chapter, a fixed penalty parameter value of $\upsilon = 1000$ is chosen. The second criterion is based on the merit function and is defined similar to (5.33) by replacing f with φ.

$$\varphi(\hat{x}) \leq \varphi(x^{*}) + \tau(\varphi(x^{0}) - f(x^{*})) \tag{5.35}$$

5.3.2.1 Importance of Feasibility Phase

CONORBIT, COBYLA, NOWPAC solves Eq. (5.15) iteratively and do not have any feasibility phase in their algorithm. Eq. (5.15) embeds the information about the constraints as well as the objective function. Therefore, it is important to elucidate the advantage of using the feasibility phase. To understand the contribution of the feasibility phase in convergence of the algorithm, it is tested with and without feasibility phase on the entire set of numerical test problems. Figure 5.6 illustrates the performance of the algorithm with and without feasibility phase. If the feasibility phase is not used, a total of 40 problems are solved while 56 problems are solved when feasibility phase is used. Therefore, the feasibility phase plays a critical role to provide superior starting point for the optimization phase and is important for convergence.

5.3.2.2 Computational Comparison with Other Solvers

The algorithm is compared with NOMAD and COBYLA. NOMAD [184] is an implementation of Mesh Adaptive Direct Search algorithm in C++ and has an option to deal with constraints using extreme barrier, filter technique or progressive barrier approach. Progressive barrier option is chosen to handle constraints while comparing on the numerical problems. All other parameters were set to default except the maximum number of evaluations is allowed to be 10000.

COBYLA [186], on the other hand, approximates the objective function and the constraints using linear functions that interpolated at the vertices of a simplex. The algorithm is based on a

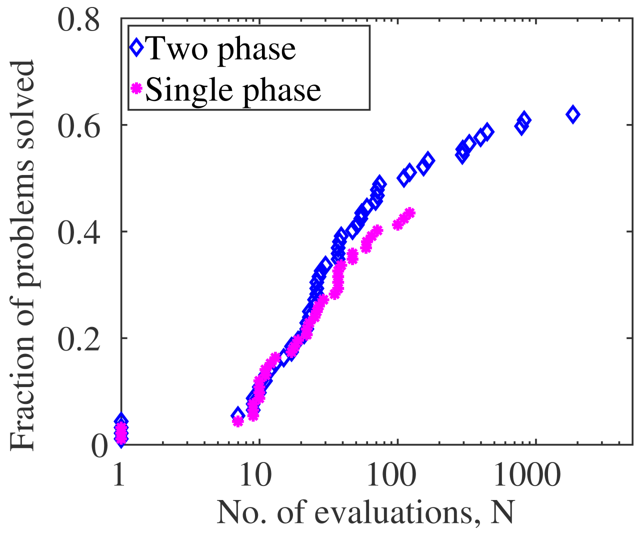


Figure 5.6: Data profile indicating the fraction of problems solved with number of evaluations based on the convergence test that the solution is within 1% of the global minima (Eq. (5.32)).The figure compares the performance of the two phase algorithm with the algorithm when no feasibility phase is involved.

trust-region framework and the new point obtained at each iteration either improves the geometry of the simplex or improves a merit function. Default parameters are used except the maximum number of evaluations are allowed to be 10000 and the lower bound on the trust-region size is set to 10^{-6}.

The data profile based on convergence criteria in (5.32) is given in Figure 5.7. The proposed two phase approach solves 56 of the 92 numerical problems, while COBYLA and NOMAD is able to attain a solution within 1% of the global minima for 38 and 33 problems, respectively. The two phase method is not only better in terms of obtaining global minima for this set of problems, it is also competitive in terms of number of evaluations. NOMAD in general takes more number of evaluations indicating that a model-based approach is more suited for smooth problems. Note that COBYLA approximates the objective function and the constraints using linear models and is able to perform marginally better than NOMAD.

The performance profile is given in Figure 5.11. Note that for $r_{t,s} = 1$, the proposed approach performs better or the same for 60 of the 92 problems. The plot of data and performance profile for

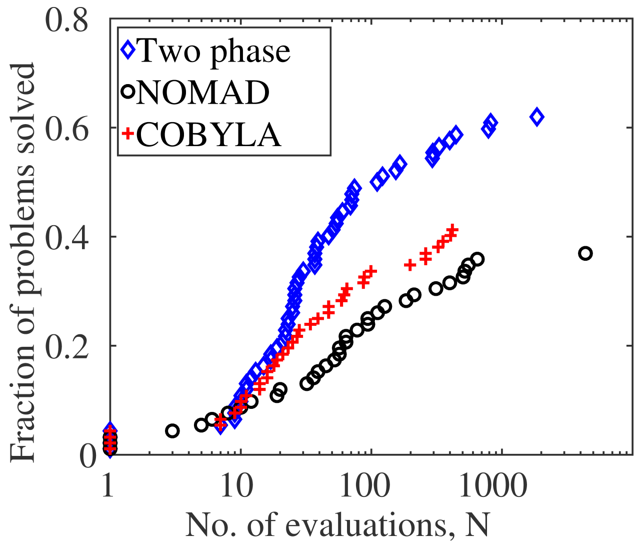


Figure 5.7: Data profile indicating the fraction of problems solved with number of evaluations based on the convergence test that the solution is within 1% of the global minima (Eq. (5.32)).

convergence test (5.35) is presented next. These plots are done based on three different required levels of reduction in the merit function: $\tau = 10^{-1}, 10^{-3}, 10^{-6}$. The data profiles are given in Figure 5.8 based on convergence test (5.35) with the requirement that a solution is obtained which achieves 90% reduction in the merit function compared to maximum reduction possible. As seen in Figure 5.8, the proposed two phase algorithm satisfies this criteria for 78 out of 92 problems while COBYLA and NOMAD satisfy the convergence test for 53 and 59 problems, respectively. This indicates that even though, there are certain problems which can not be solved to global optimality, significant reduction in at least constraint violation is achieved. When τ is decreased further to 10^{-3}, as expected, the performance deteriorates for all the algorithms. However, the proposed approach still outperforms the other two and COBYLA solves more number of problems compared to NOMAD. Decreasing τ further to 10^{-6} reduces the number of problems that satisfy the convergence test (5.35) but the rank of performance of the solver still remains the same.

The performance profiles with convergence test (5.35) are given in Figure 5.12(a), 5.12(b), and 5.12(c) for $\tau = 10^{-1}, 10^{-3}$ and 10^{-6}, respectively. For $\tau = 10^{-1}$, at small alpha values, proposed algorithm slightly outperforms COBYLA but with increasing α, COBYLA performs better. NO-

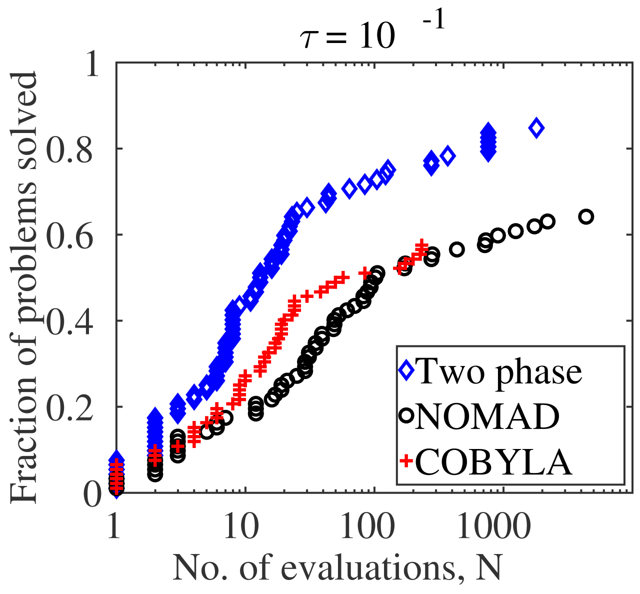


Figure 5.8: Data profile indicating the fraction of problems solved with number of evaluations based on the convergence test that a significant reduction in the final merit function is obtained when compared to its value at the initial point (Eq. (5.35) with $\tau = 10^{-1}$).

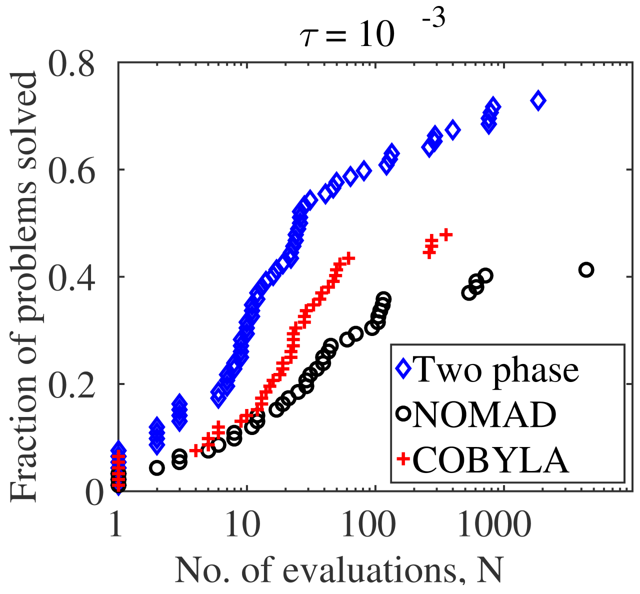


Figure 5.9: Data profile indicating the fraction of problems solved with number of evaluations based on the convergence test that a significant reduction in the final merit function is obtained when compared to its value at the initial point (Eq. (5.35) with $\tau = 10^{-3}$).

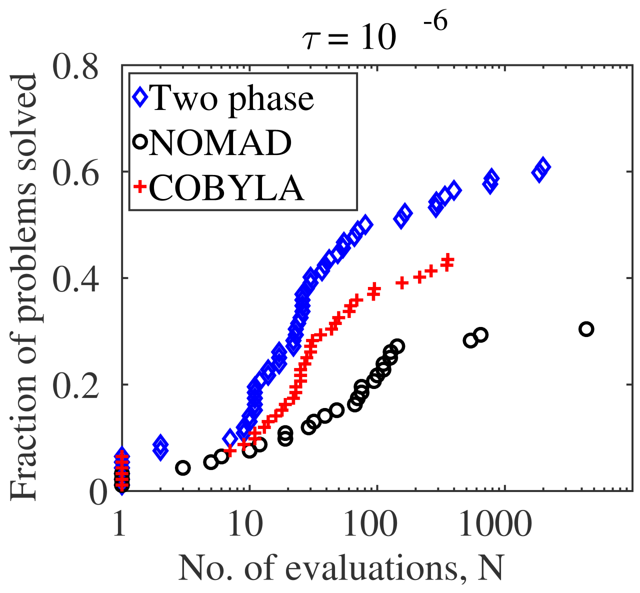


Figure 5.10: Data profile indicating the fraction of problems solved with number of evaluations based on the convergence test that a significant reduction in the final merit function is obtained when compared to its value at the initial point (Eq. (5.35) with $\tau = 10^{-6}$).

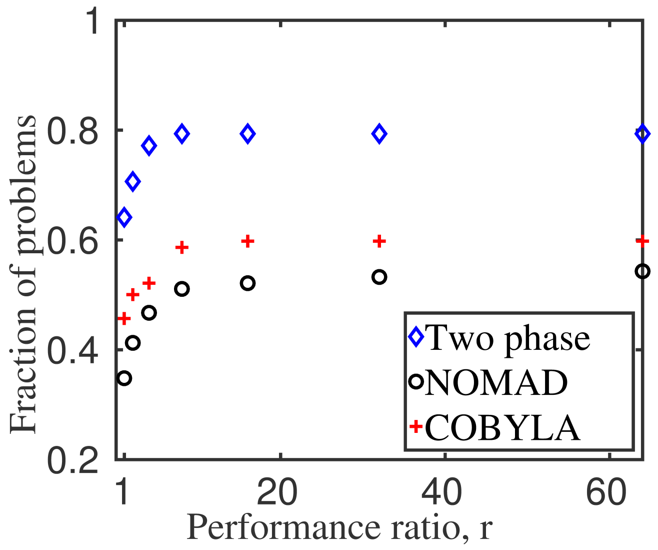


Figure 5.11: Performance profile indicating the relative performance of the solvers based on the convergence test that the solution is within 1% of the global minima (Eq. (5.32)).

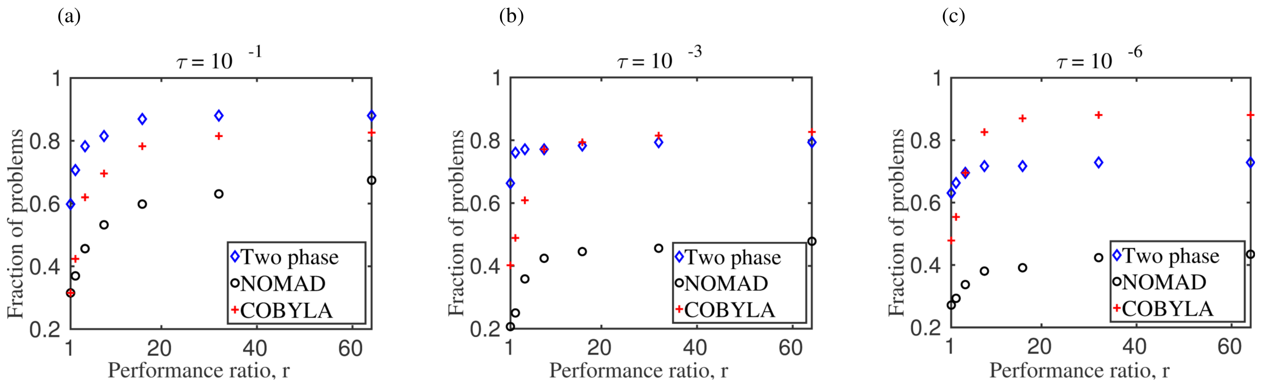


Figure 5.12: Performance profile indicating the relative performance of the solvers based on the convergence test that a significant reduction in the final merit function is obtained when compared to its value at the initial point (Eq. (5.35)) for (a) $\tau = 10^{-1}$, (b) $\tau = 10^{-3}$, (c) $\tau = 10^{-6}$

MAD performs inferior compared to the other methods on these set of test problems. When the required level of reduction is increased, the proposed method outperforms other methods and this can be observed in Figure 5.12(b) and 5.12(c). The proposed approach is efficient in terms of the required number of evaluations. This is due to the use of an efficient model improvement algorithm which replaces no more than 2 interpolating points whenever the geometry of the samples deteriorates. The success of tje method can also be attributed to the use of global solver (ANTIGONE in this case) for solving the sub-problems, the algorithm for improving the geometry of the samples, and the initialization of the initial trust-region for the entire space. The initial trust-region allows for globally exploring the search space, while using a global solver gives an indication of the global solution if the surrogate model is able to approximate the original model well in the limited number of initial samples. However, using a global solver is computationally expensive. The performance of the algorithm is compared on a few problems using local and global solver to solve the subproblems. It is observed that although using local solver is computationally inexpensive, more problems converged to the global minima for the cases when the global solver is used. However, this is not conclusive and requires further investigation.

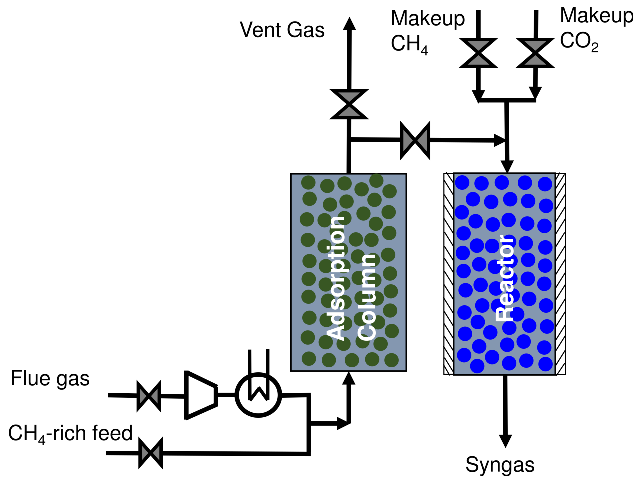


Figure 5.13: Schematic of the process for integrated carbon capture and conversion.

5.3.3 Optimal design of a Cyclic and Integrated Carbon Capture and Conversion Process

A chemical engineering case study is now presented which has combination of known and unknown analytical form of the constraints with black-box constraints. Due to emergence of process intensification [191, 192] efficient BBO algorithms are needed more than ever. Process intensification merges processes with different aims to lessen number of units in order to reduce cost and/or energy consumption. In these cases, different phenomena interact in a complex manner and obtaining even the appropriate operating conditions and design that satisfies regulatory and performance constraints is not trivial [1]. Therefore, an optimization strategy is needed to asses if the intensified process is economically viable and that it meets quality and regulatory constraints. A multifunctional process is proposed that integrates adsorption and reaction to directly utilize CO_2 from flue gas to produce syngas in [1]. The operating conditions and design parameters are optimized of the process using a preliminary version of the current algorithm. This chapter presents an improved optimization algorithm with more effective sampling strategy. Instead of performing multiple evaluations at each iteration, no more than 2 additional function calls are used.

The process configuration is given in Figure 5.13. The flue gas contains large amount of N_2 (around 86 %), while the rest is CO_2 and it is therefore important to separate CO_2 from N_2. The overall process contains two columns. The first is an adsorption column while second is a reactor. The process is configured such that CO_2 from flue gas is adsorbed more favorably than N_2 using an adsorbent. After adsorption, CO_2 is desorbed using methane by utilizing the gradient in concentration rather than pressure. This leads to energy savings since applying pressure swings is energy intensive. The adsorption and desorption steps are run in a cyclic manner until a cyclic steady-state is achieved. The outlet of the adsorption column contains CO_2, N_2 and CH_4. It would be advantageous to remove as much N_2 as possible from the gas mixture and keeping the loss of CO_2 and CH_4 to a minimum. The gas is then sent to the reaction section filled with catalyst that enables dry reforming. CO_2 and CH_4 need to be mixed in a ratio conducive for dry reforming and is considered to be one of the parameters for optimization.

5.3.3.1 Process Modeling

The process consists of two sections, namely adsorption and reaction. The adsorption section is dynamic in nature and is therefore, modeled by one-dimensional nonlinear algebraic partial differential equations (NAPDE). The model includes component-wise mass balance, energy balance and other relationships. The gas phase mass balance includes the accumulation, convection, axial dispersion and adsorption terms and is given by the following equation:

$$\frac{\partial c_i}{\partial t} = -\frac{\partial}{\partial z}\left(- cD\frac{\partial y_i}{\partial z} + c_i v\right) - \frac{(1-e)}{e}\frac{\partial q_i}{\partial t} \quad \forall i \in \mathbb{G}_A \tag{5.36}$$

Here c_i and y_i denotes the bulk concentration and mole fraction of the gaseous species and q_i is the loading on the solid adsorbent. $\mathbb{G}_A$ is the set of species in adsorption column, $\mathbb{G}_A = \{\mathrm{CO_2}, \mathrm{CH_4}, \mathrm{N_2}\}$. The variation of the pressure with time and along the bed is given by:

$$\frac{\partial P}{\partial t} = \frac{P}{T}\frac{\partial T}{\partial t} - T\frac{\partial}{\partial z}\left(\frac{Pv}{T}\right) - RT\frac{(1-e)}{e}\sum_{i\in\mathbb{G}_A}\frac{\partial q_i}{\partial t} \tag{5.37}$$

where P and T are the column pressure and temperature respectively. Assuming gas and adsorbent to be at the same temperature, energy balance of the streams are given by:

$$
\begin{aligned}
(1-e)\left(\rho_a C_{p,ad}\frac{\partial T}{\partial t} + C_{p,g}\frac{\partial(\sum_{i\in\mathbb{G}_A} q_i T)}{\partial t}\right) + eC_{p,g}\frac{\partial(\rho_g T)}{\partial t} + eC_{p,g}\frac{\partial(\rho_g v T)}{\partial z} \\
= K\frac{\partial^2 T}{\partial z^2} + (1-e)\sum_{i\in\mathbb{G}_A}(-\Delta H_i)\frac{\partial q_i}{\partial t} - \frac{2h_{ic}}{r_{ic}}(T - T_c)
\end{aligned}
\tag{5.38}
$$

The heat will also be transferred across the column wall and is given by equation 5.39 as follows.

$$
\rho_c C_{p,c}\frac{\partial T_c}{\partial t} = K_c\frac{\partial^2 T_c}{\partial z^2} + \frac{2r_{ic}h_{ic}}{r_{oc}^2 - r_{ic}^2}(T - T_c) - \frac{2r_{oc}h_{oc}}{r_{oc}^2 - r_{ic}^2}(T_c - T_{am}) \tag{5.39}
$$

where T_c is the wall temperature of the column and T_{am} is the ambient temperature in K respectively.

The linear driving force equation describing the rate of gas adsorption into the adsorbent is given by the following equation.

$$
\frac{\partial q_i}{\partial t} = k_i(q_i^* - q_i) \quad \forall i \in \mathbb{G}_A \tag{5.40}
$$

where k_i is the lumped mass transfer coefficient and q_i^* is the equilibrium gas loading. For a complete description of the NAPDE model with other empirical relations and boundary conditions and values of parameters used, the readers are referred to Eq. 9-19 of [1].

The outlet gas mixture from the adsorption column is mixed with additional CO_2 and CH_4. The primary reactions considered in the conversion of CO_2 and CH_4 to methane is the dry reforming and the reverse water gas shift reaction:

$$
\begin{aligned}
CH_4 + CO_2 \Leftrightarrow 2CO + 2H_2 \quad \Delta H_{298K} = 247kJ/mol \\
CO_2 + H_2 \Leftrightarrow CO + H_2O \quad \Delta H_{298K} = 41.7kJ/mol
\end{aligned}
\tag{5.41}
$$

The dry reforming reaction is endothermic and produces more number of moles in the product as compared to the reactants. So, according to Le Chatelier's principle, the reaction is favorable at high temperature and low pressure. Since operation of adsorption section is favored at low temperature, the outlet from the adsorption column is heated in the reaction section for favorable conversion. Although, the adsorption process is dynamic, the adsorption column outlet gas over a cycle is mixed with additional CO_2 and CH_4 and the mixture is then fed to the reactor at constant flow rate and composition. The reactor is thus operated at steady state and is modeled by one dimensional pseudohomogenous model assuming plug flow and isothermal behavior. The component mass balances in the reactor is given by:

$$\frac{dF_i^R}{dz} = \rho_r A_r r_i^g \quad \forall i \in \mathbb{G}_R \tag{5.42}$$

where F_i^R is the species flow rate and r_i is the rate of generation of each species. The heat duty required (Q_r) to maintain isothermal operation of the reactor is obtained by integrating the following equation describing the heat consumed by the reactions.

$$\frac{dQ_r}{dz} = \rho_r A_r \sum_m (-\Delta H_m) R_m \tag{5.43}$$

The pressure variation along the bed length of the reactor (P_r) is given by Ergun equation.

$$\frac{dP_r}{dz} = -\frac{\rho_{rg} v_r (1 - e_r)}{d_{ce} e_r^2} f \tag{5.44}$$

Here ρ_{rg} is the density of the gas phase in kg/m^3, v_r is the superficial velocity in m/s and f is the friction factor. Other important empirical relations, rate of reactions and boundary conditions are given in Eq. 22-26, 28-31 and 34-36 in [1].

5.3.3.2 Process Simulation and Constraints

The performance of the process is evaluated based on several metrics. Two very important metrics are utilization of CO_2 and cost of the process. Since the process steps involve venting of gases, it needs to be ensured that the greenhouse gases (CO_2 and CH_4) are emitted below those allowed by regulatory agencies. Additionally, in order for syngas to be utilized for further processes, it should be of high quality with adequate ratio of CO and H_2.

As mentioned earlier, the adsorption process is described by NAPDE model. To solve the model, the partial differential equations are spatially discretized using upwind differencing scheme to form a set of ordinary differential equations (ODEs). These set of ODEs are solved using ode23s in MATLAB. On solving the model, concentration and temperature profiles are obtained along the adsorption column length and over time. The adsorption process is a cyclic two-step process. During the first step, flue gas enters the column and CO_2 is selectively adsorbed. In the second step, CH_4-rich feed enters the column and CO_2 is desorbed due to concentration driving force. The initial condition of the column during second step is the final condition in the first step. The two steps are repeated in a cyclic manner until cyclic steady state (CSS) is attained. The input conditions for the reactor are calculated based on the solution obtained at CSS. The reactor model is given by ordinary differential equations (ODEs) and are solved using ode23s. For obtaining the simulations, the number of spatial discretizations and cycles used are both set to five.

Table 5.4: Decision variables for Problem (5.57).

Decision variables	x^L	x^U
Pressure set at the outlet of adsorption column (P_{OA})[bar]	1	10
Length of adsorption column (L_a)[m]	0.5	2.5
Reactor temperature (T_c) [K]	373	1223
Reactor bed length (L_r) [m]	0.5	10
Duration of step 1 (t_{1s}) [s]	10	t_{ct}
Total cycle time (t_{ct}) [s]	10	200
Venting start time (t_{v1}) [s]	0	t_{ct}
Venting end time (t_{v2}) [s]	0	t_{ct}
Makeup CO_2 before reaction ($F^M_{CO_2}$) [mol/s]	0	5
Makeup CH_4 before reaction ($F^M_{CH_4}$) [mol/s]	0	5

Iyer et. al. [1] utilized simulations to study the performance metrics of the process and identified key optimization parameters. The key decision variables along with the bounds are listed in

Table 5.4. Note that the process needs to occur in a chronological order such that it makes logical sense. The following set of unrelaxable constraints always need to be satisfied to ensure that the simulation does not fails:

$$10 \leq t_{1s} \leq t_{ct} \tag{5.45}$$

$$10 \leq t_{ct} - t_{1s} \leq t_{ct} \tag{5.46}$$

$$0 \leq t_{v1} \leq t_{ct} \tag{5.47}$$

$$0 \leq t_{v2} \leq t_{ct} \tag{5.48}$$

$$10 \leq t_{v2} - t_{v1} \leq t_{ct} \tag{5.49}$$

Eq. (5.45) states that the first step time does not exceed the cycle time. Eq. (5.46) puts a constraint that the second step should be less than the cycle time. The lower bound in Eq. (5.45) and 5.46 allows for the first and second step to run for at least 10 seconds. Similarly, the start and end time of venting should always be less than the cycle time is given in Eq. (5.47) and (5.48). Eq. (5.49) states that the difference in the start and ending times should be at least 10 seconds. In other words, the venting can not end before starting. The upper bound will always be satisfied due to the relations 5.47 and 5.48.

Note that the constraints (5.45)-(5.49) are expressed explicitly in terms of the decision variables. In this problem, there are constraints which can not be explicitly expressed in terms of the decision variables unless the model is discretized. Such constraints are referred as black-box constraints and the value of the constraint is obtained as a result of a simulation. Unlike constraints in (5.45)-(5.49), violating the black-box constraints will not result in a simulation failure. These constraints are given as follows:

$$\frac{l_{CH_4}^{max} - 10}{100} \leq 0 \tag{5.50}$$

$$\frac{U_{CO_2}^{min} - 90}{100} \geq 0 \tag{5.51}$$

$$SG^{min} \geq 0.9 \tag{5.52}$$

$$SG^{max} \leq 1.1 \tag{5.53}$$

$$y_{CH_4}^{msg} \leq 0.03 \tag{5.54}$$

$$y_{CO_2}^{msg} \leq 0.03 \tag{5.55}$$

$$y_{N_2}^{msg} \leq 0.1 \tag{5.56}$$

Eq. (5.50) gives an upper bound on the methane loss from the process since it is a greenhouse gas and will violate regulatory constraints. The constraint on the minimum overall utilization of CO_2 is given in Eq. (5.51). The requirement on minimum and maximum ratio of H_2/CO is given by inequalities (5.52) and Eq. (5.53). It is important to maintain this ratio so that syngas can be used in further downstream processes. The produced syngas should be of high quality with minimal CO_2, CH_4 and N_2 and it is imposed by Eqs. (5.54)-(5.56). Since the process is novel with large number of variables and complex interactions between the variables and constraints it is not trivial to obtain a feasible point using parametric studies. Two key performance metrics are of main interest for the process. The design and operating parameters are desired that yields minimum cost per ton of syngas produced and secondly maximize overall CO_2 utilization. The two objectives are black-box and the details of calculation procedure can be found in [1]. The resulting optimization problem is a grey-box problem. Rigorous optimization needs to be performed to balance different trade-offs.

The optimization problem is formulated as follows:

$$\begin{aligned} \min_{x} \quad & f(x,z) \\ \textit{s.t.} \quad & g_i(x,z) \leq 0 \quad \forall i \in \{1,\dots,p\}(Eq.(5.50)-5.56) \\ & g_j(x) \leq 0 \qquad \forall j \in \{1,\dots,q\}(Eq.(5.45)-5.49) \\ & h_u(x,z) = 0 \quad \forall u \in \{1,\dots,s\}(\text{process model}) \\ & x \in [x^L, x^U] \end{aligned} \tag{5.57}$$

where z represents intermediate variables in the process model. The variables z can be thought of as the black-box part of the optimization problem (5.57). The model equations $h(x,z) = 0$ are solved for z for given value of x and thereby, the objective function and constraint values ($g_i(x,z)$) are obtained.

5.3.3.3 Optimization Results

The following four case studies are performed:

Case 1: Minimize cost of production of syngas using natural gas as methane source,

Case 2: Maximize overall CO_2 utilization using natural gas as methane source,

Case 3: Minimize cost of production of syngas using biogas as methane source,

Case 4: Maximize overall CO_2 utilization using biogas as methane source.

The sampling scheme given by Eq. (5.25) is used to generate initial samples such that they are feasible with respect to the known constraints given by Eq. (5.45) - (5.49). Here, $6n + 1$ samples are maintained at all the iterations. Same initial guess was provided for all the case studies and is given in Table 5.5. The two phase algorithm is then applied to the problem to generate optimum results for the four cases. Figure 5.14 shows the progress of the best objective function values and the constraint violations obtained with number of evaluations for case 1. The initial point is infeasible and, therefore, the feasibility phase is triggered which finds a feasible point within 68 evaluations. Once a feasible point is obtained, the optimization phase is started that reduces

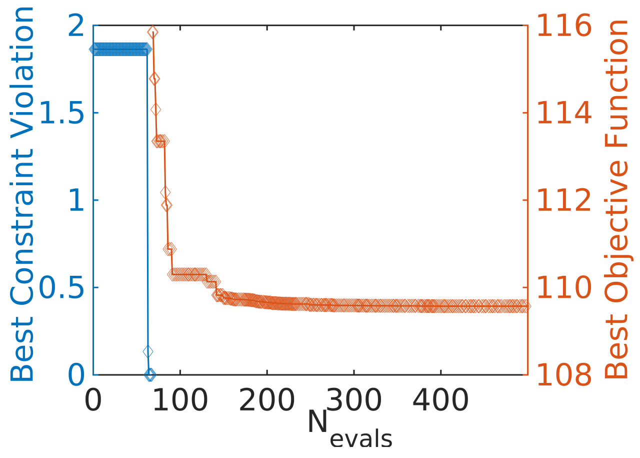


Figure 5.14: Progress of the best constraint violation and the best objective function value with number of evaluations for case 1.

the objective function value. The algorithm ultimately provides the optimal design and operating conditions that gives the production cost to be \$109.57. The value of the objective function, the corresponding decision variables at the optimum are given in Table 5.6. The constraint violation and the objective function at the initial point and the number of evaluations required for obtaining the final optimal result are also reported for all the cases.

Table 5.5: Initial guess for Problem (5.57).

Decision variables	x^0
P_{OA}	1.001
L_a	0.71
T_c	478.69
L_r	4.05
t_{1s}	135.85
t_{ct}	149.92
t_{v1}	29.76
t_{v2}	122.87
$F^M_{CO_2}$	1.26
$F^M_{CH_4}$	0

It is observed from Table 5.6 that the optimal pressure is at the lower bound throughout all

Table 5.6: Optimum results obtained for the four case studies with constraint violation and objective function at the initial point.

Case Studies	$f(\hat{x})$	$\hat{x}(P_{OA}, L_a, T_c, L_r, t_{1s}, t_{ct}, t_{v1}, t_{v2}, F^M_{CO_2}, F^M_{CH_4})$	N_{evals}	$\theta(x^0)$	$f(x^0)$
Case 1	\$109.57	(1, 0.8513, 1209.9, 0.5, 85.31, 95.31, 58.86, 91.3354, 5, 4.8)	681	2.17	3.32×10^5
Case 2	99.71%	(1, 2.49, 1223, 8.78, 12.19, 22.19, 9.69, 19.69, 4.48, 4.64)	332	1.38	0.88
Case 3	\$110.27	(1, 2.33, 1208.1, 0.5, 189.99, 199.99, 189.86, 199.95, 5, 4.90)	662	2.2	4.14×10^5
Case 4	99.75%	(1, 2.37, 1223, 4.96, 55.07, 65.07, 47,16, 65.07, 4.6, 5)	244	1.41	1.017

the the four cases. The algorithm, however, has increased the temperature of the reactor and the amounts of makeup CO_2 and CH_4 in order to increase the conversion of CO_2 and CH_4 to syngas while meeting the constraints on process metrics and improving the objective in each case. It is also observed that the temperature has reached the upper bound while maximizing CO_2 utilization in both the natural gas and biogas cases. This is because dry reforming is an endothermic reaction, hence, a high temperature is needed to increase the conversion of CO_2 and CH_4 to syngas. This helps in meeting constraints on loss of CO_2 and CH_4, while improving overall utilization and reducing the total cost per ton of syngas. Although increase in temperature results in more CO_2 conversion and more syngas being produced, there are increased utility costs involved in maintaining the reactor at high temperature. This is observed from the optimal reactor temperature for the cost minimization problem for both the natural gas and biogas cases not reaching the upper bound (1223 K). Instead it is maintained at a high enough value (around 1208 K) while balancing different trade-offs. Similarly it is observed that while moderate to high values of reactor lengths are obtained at optimum for the maximizing CO_2 utilization case, the optimal reactor length is at the lower bound in the cost minimization case for both gases. All the decision variables relating to the step and venting times in each case meet the unrelaxable constraints listed in Eq. (5.45)–(5.49). While the flue gas step durations and venting times at optimum are vary between the different cases, methane rich feed step duration however is at the lower bound of 10 s in all cases.

Table 5.7: Comparison of solution provided by the Two phase algorithm, COBYLA and NOMAD.

Case Studies	Two phase		COBYLA		NOMAD	
	$f(\hat{x})$	N_{evals}	$f(\hat{x})$	N_{evals}	$f(\hat{x})$	N_{evals}
Case 1	\$109.57	681	\$115.75	364	$\$2.79\times10^{5*}$	628
Case 2	99.71%	332	99.6%	206	97.59%	1248
Case 3	\$110.27	662	\$114.94	203	\$109.61	1473
Case 4	99.75%	244	99.25%	261	98.8%	1382

* A feasible solution could not be obtained.

The performance of the algorithm is compared on the case studies with COBYLA and NOMAD. For both the solvers, it is not possible to avoid simulations at points which violates Eq. (5.45) - Eq (5.49). The simulations fail if the design point do not satisfy these constraints. To avoid simulation failure, a very high arbitrary value (10^{15}) is assigned for the objective function and the constraints. All solvers are given the same initial guess listed in Table 5.5. Table 5.7 provides a summary of the results. COBYLA is economical in terms of function evaluations compared to the two phase algorithm but does not provide a better objective function value. NOMAD, on the other hand takes a lot of function evaluations. Furthermore, for case 1, a feasible point could not be obtained. However, NOMAD finds a point that gives the least objective function value compared to all other algorithms for case 3.

In summary, the two phase algorithm has been able to obtain a feasible point from a high initial constraint violation and then improve the objective function value in each case for a complex chemical process model involving both explicit and black-box constraints and inter-dependent decision variables. Thus, the use of this algorithm with such novel integrated process designs can greatly help in identifying the feasible and optimal operating regions which may otherwise be difficult to predict based on just parametric studies of individual variables. This can accelerate and pave the way for effective feasibility analyses of novel integrated process technologies.

5.4 Conclusions

After addressing global optimization of grey-box problems with bound constraints (GPBC) and grey-box constraints (GPGC) in Chapter 2 and 3, respectively, black-box problems with bound-constraints are optimized in Chapter 4. This chapter addresses general black-box problems that may have bound, known, unknown and hybrid constraints. A trust-region based two phase algorithm is presented. The first phase of the algorithm finds a feasible point if the initial point is infeasible while the second phase decreases the objective function. It is also observed that feasibility phase is critical for superior performance of the algorithm. A maximum of two samples are replaced at each iteration using model improvement algorithm to improve the geometry of the samples and make the model fully linear. The samples are economically used and this makes the algorithm suitable for problems with computationally expensive simulations. The efficiency of the algorithm is shown by comparing with two widely used solvers namely, NOMAD and COBYLA. The performance of the solvers are compared by constructing data and performance profile. Two convergence tests were used to plot data and performance profile. The two-phase algorithm not only finds the global minima for more number of problems compared to other solvers but achieves that in economical number of evaluations. The algorithm is also applied to optimize a cyclic and integrated carbon capture and conversion process. Four case studies are solved involving maximization of CO_2 utilization and cost minimization with biogas and natural gas used as methane source. Starting from a point with high infeasibility, the algorithm is able to obtain design and operation parameters that enables the process technology to be effective for CO_2 utilization. NOMAD and COBYLA were also applied to the integrated carbon capture and conversion case study. On comparing the performance of the three algorithms, it was observed that the proposed approach gives a high quality solution using economical number of evaluations.

6. CONCLUSIONS AND RECOMMENDATIONS

Black-box optimization problems are prevalent in several disciplines including engineering, science, finance, medicine and operations research. Effective black-box optimization strategies will enable more efficient processes and design, and contribute towards global energy sustainability. This dissertation proposed several novel methods and algorithms for optimization of classes of black-box and grey-box problems.

It was recognized from the beginning that finding guaranteed global optimum solution of a general black-box problem without using any information of the original problem. Therefore, the goal in Chapter 2 was to identify a class of grey-box problems for which it is possible to compute an ϵ-global optimum solution. Another important consideration is that the lower bound should be computed in finite number of black-box function evaluations. A broad class of grey-box problems with bound constraints was defined for which the maximum of the upper bounds of the diagonal elements of the Hessian are available. With this information, a novel underestimator called edge-concave underestimator (ECU) was then constructed. ECU exhibits a critical property that its minima lies at a vertex when box-constraints are involved and this allows computing lower bound in finite number of evaluations. A branch-and-bound algorithm is developed that enables to find ϵ-global optimal solution. The proposed methodology is also compared to other available solvers on several nonconvex problems and results indicate that the proposed method is computationally advantageous. In the literature, there are several instances that if a model is too complex, it is treated as black-box. This work presents a key step towards global optimization of many of these problems based on the argument that it is possible to construct a valid underestimator if only one parameter, i.e. the maximum of the upper bound of the diagonal elements of the Hessian, is known. Identification of practical applications that belong to this class of grey-box have yet to be fully explored. An important area that needs more research is exploring the possibility of estimating the maximum of the diagonal elements of the Hessian using simulation data.

The global optimization of problems with integral terms in the objective function and constraints with embedded system of nonlinear ODEs are addressed in Chapter 3. These problems also belong to a special case of grey-box problems defined in Chapter 2. However, it is possible to compute the upper bounds of all the diagonal elements of the Hessian and enables construction of tight relaxations. The terms in the objective function and constraints are separated into algebraic and dynamic parts and replaced by the corresponding ECUs or their convex envelopes to formulate the relaxed problem. The problems were solved to ϵ-global optimality using a branch-and-bound framework. The effectiveness of the method is illustrated through several case studies and compared to existing approaches. It is concluded that ECUs provide a promising avenue for relaxations of optimization problems with embedded ordinary differential equations. The challenge that is identified in this work is that while there are packages that enable symbolic differentiation of algebraic expressions, but software tools are needed that automatically provides expressions for the sensitivity equations by taking the derivatives of the differential equations. Furthermore, it is not always possible to compute finite and tight bounds on the state variables and its derivatives. A lot of advancements have been made to compute tight bounds but the problem is not completely solved. These challenges limit the applicability of the proposed approach to optimization problems with embedded large-scale system of ordinary differential equations. Until these issues are resolved, large-scale embedded system of ODE/PDEs can be treated as black-box and optimization approaches based on simulation data provides promising alternatives. Simulation/experimental data based optimization approaches also enable optimization of problems where the underlying model is not available.

Chapter 4 addresses the optimization of box-constrained black-box problems. The problem is completely black-box in the sense that only simulation data are needed. A novel algorithm based on univariate projection is developed. A univariate function referred as the lower envelope is identified in a new space defined by a linear combination of the decision variables. An algorithmic framework, UNIPOPT was proposed based on identifying the points on the lower envelope and subsequently using them for optimization. To identify the points, a predictor and corrector approach is used, where approximated Sensitivity theorem serves as the prediction step and a trust-region based algorithm is employed as the correction step. Through rigorous mathematical proof, it was shown that the algorithm converges to the local minima of the original black-box problem when certain assumptions are satisfied. The algorithm is thoroughly compared to existing model-based approaches using an extensive set of test problems. Computational experiments suggest that the proposed methodology explores the global space more efficiently but, it takes a lot of function evaluations. Specifically, the bottleneck is the correcting algorithm. Therefore, eliminating the correcting step could lead to significant improvement in the performance of the overall algorithm. Furthermore, the prediction step involves using approximated Sensitivity theorem that relies on the ability of the surrogate model to approximate the original black-box function. Of course, more accurate surrogate models will lead to a better prediction resulting in less function evaluations by the correcting algorithm. More advancements are needed in sampling and modeling to better capture the original black-box function. Currently, the idea of projection has been viewed from the point of view of f space. However, more research is needed towards identifying the optimal parametric map of x corresponding to the lower envelope. Application of several important results established in the parametric programming literature (see e.g., [193, 194, 195]) could be used in the context of black-box problems to potentially improve the performance of the approach. The idea of projection has only been investigated for black-box problems and can be extended to nonlinear programming problems.

The algorithm proposed in Chapter 5 provides an avenue for optimization of constrained black-box problems. In particular, inequality constraints are considered that can be of known or unknown form. A trust-region based two-phase algorithm is developed. The first phase finds a feasible point if the initial guess provided by a user is infeasible. The point given by the feasibility phase is then passed to the optimization phase so that the algorithm gets a clue of the feasible region. The optimization phase then proceeds to find the optimal point while maintaining feasibility. This algorithm is also guaranteed to converge to the local minima of the original problem. The two-phase methodology is compared to other solvers on an extensive set of problems from standard test libraries and a carbon capture and conversion case study. It is shown that this algorithm is computationally advantageous to other solvers and it is also able to handle unrelaxable constraints. The computational performance of the approach on black-box equality constraints is unexplored. Furthermore, it has been observed that the algorithm finds the optimum solution quickly but takes several evaluations/iterations to converge. In general nonlinear programming problem, criticality measure based on actual derivatives are used and that allows to converge faster even when the trust-region size is not very small. However, for black-box problems, criticality measure based on the derivatives of the surrogate model is used. When the trust-region size is large and the model fully-linear, there is a gap between the derivatives of the actual function and the surrogate model that is proportional to trust-region size. This leads to inexact estimation of the criticality measure and the algorithm converges only when the trust-region becomes small. One way to overcome the slow convergence is to strategically compute exact derivatives, wherever possible, so that the algorithm converges faster without taking a lot of expensive function evaluations. One application where computing exact derivative is possible is for the system governed by large-scale ODE/PDEs that can not be managed by methodologies given in Chapter 3.

REFERENCES

[1] S. S. Iyer, I. Bajaj, P. Balasubramanian, and M. M. F. Hasan, "Integrated carbon capture and conversion to produce syngas: Novel process design, intensification, and optimization," *Industrial & Engineering Chemistry Research*, vol. 56, no. 30, pp. 8622–8648, 2017.

[2] M. S. Khan and M. Lee, "Design optimization of single mixed refrigerant natural gas liquefaction process using the particle swarm paradigm with nonlinear constraints," *Energy*, vol. 49, pp. 146–155, 2013.

[3] A. Dutta, I. A. Karimi, and S. Farooq, "Heating value reduction of LNG (liquefied natural gas) by recovering heavy hydrocarbons: Technoeconomic analyses using simulation-based optimization," *Industrial & Engineering Chemistry Research*, vol. 57, no. 17, pp. 5924–5932, 2018.

[4] H. Rahmanifard, R. Vakili, T. Plaksina, M. R. Rahimpour, M. Babaei, and X. Fan, "On improving the hydrogen and methanol production using an auto-thermal double-membrane reactor: Model prediction and optimisation," *Computers & Chemical Engineering*, 2018.

[5] A. Alarifi, Z. Liu, F. S. Erenay, A. Elkamel, and E. Croiset, "Dynamic optimization of lurgi type methanol reactor using hybrid ga-gps algorithm: the optimal shell temperature trajectory and carbon dioxide utilization," *Industrial & Engineering Chemistry Research*, vol. 55, no. 5, pp. 1164–1173, 2016.

[6] Y. K. Salkuyeh and T. A. Adams II, "A novel polygeneration process to co-produce ethylene and electricity from shale gas with zero CO_2 emissions via methane oxidative coupling," *Energy Conversion and Management*, vol. 92, pp. 406–420, 2015.

[7] M. M. F. Hasan, E. L. First, and C. A. Floudas, "Cost-effective CO_2 capture based on in silico screening of zeolites and process optimization," *Physical Chemistry Chemical Physics*, vol. 15, no. 40, pp. 17601–17618, 2013.

[8] E. L. First, M. M. F. Hasan, and C. A. Floudas, "Discovery of novel zeolites for natural gas purification through combined material screening and process optimization," *AIChE Journal*, vol. 60, no. 5, pp. 1767–1785, 2014.

[9] M. M. F. Hasan, E. L. First, F. Boukouvala, and C. A. Floudas, "A multi-scale framework for CO_2 capture, utilization, and sequestration: CCUS and CCU," *Computers & Chemical Engineering*, vol. 81, pp. 2–21, 2015.

[10] A. Nuchitprasittichai and S. Cremaschi, "Optimization of CO_2 capture process with aqueous amines using response surface methodology," *Computers & Chemical Engineering*, vol. 35, no. 8, pp. 1521–1531, 2011.

[11] L. T. Biegler, Y.-d. Lang, and W. Lin, "Multi-scale optimization for process systems engineering," *Computers & Chemical Engineering*, vol. 60, pp. 17–30, 2014.

[12] B. Beykal, F. Boukouvala, C. A. Floudas, N. Sorek, H. Zalavadia, and E. Gildin, "Global optimization of grey-box computational systems using surrogate functions and application to highly constrained oil-field operations," *Computers & Chemical Engineering*, vol. 114, pp. 99–110, 2018.

[13] N. Ploskas, C. Laughman, A. U. Raghunathan, and N. V. Sahinidis, "Optimization of circuitry arrangements for heat exchangers using derivative-free optimization," *Chemical Engineering Research and Design*, 2017.

[14] K. Palmer and M. Realff, "Optimization and validation of steady-state flowsheet simulation metamodels," *Chemical Engineering Research and Design*, vol. 80, no. 7, pp. 773–782, 2002.

[15] R. Lima, G. François, B. Srinivasan, and R. Salcedo, "Dynamic optimization of batch emulsion polymerization using msimpsa, a simulated-annealing-based algorithm," *Industrial & engineering chemistry research*, vol. 43, no. 24, pp. 7796–7806, 2004.

[16] I. M. Mujtaba, N. Aziz, and M. A. Hussain, “Neural network based modelling and control in batch reactor,” *Chemical Engineering Research and Design*, vol. 84, no. 8, pp. 635–644, 2006.

[17] J. A. Egea, D. Vries, A. A. Alonso, and J. R. Banga, “Global optimization for integrated design and control of computationally expensive process models,” *Industrial & Engineering Chemistry Research*, vol. 46, no. 26, pp. 9148–9157, 2007.

[18] J. A. Caballero and I. E. Grossmann, “An algorithm for the use of surrogate models in modular flowsheet optimization,” *AIChE Journal*, vol. 54, no. 10, pp. 2633–2650, 2008.

[19] F. Boukouvala and M. G. Ierapetritou, “Surrogate-based optimization of expensive flowsheet modeling for continuous pharmaceutical manufacturing,” *Journal of Pharmaceutical Innovation*, vol. 8, no. 2, pp. 131–145, 2013.

[20] S. Yang, S. Kiang, P. Farzan, and M. Ierapetritou, “Optimization of reaction selectivity using cfd-based compartmental modeling and surrogate-based optimization,” *Processes*, vol. 7, no. 1, p. 9, 2019.

[21] J. Müller, “Miso: mixed-integer surrogate optimization framework,” *Optimization and Engineering*, vol. 17, no. 1, pp. 177–203, 2016.

[22] L. S. Dias, R. C. Pattison, C. Tsay, M. Baldea, and M. G. Ierapetritou, “A simulation-based optimization framework for integrating scheduling and model predictive control, and its application to air separation units,” *Computers & Chemical Engineering*, vol. 113, pp. 139–151, 2018.

[23] C. Audet and D. Orban, “Finding optimal algorithmic parameters using derivative-free optimization,” *SIAM Journal on Optimization*, vol. 17, no. 3, pp. 642–664, 2006.

[24] J. Liu, N. Ploskas, and N. V. Sahinidis, “Tuning BARON using derivative-free optimization algorithms,” *Journal of Global Optimization*, pp. 1–27, 2018.

[25] J.-H. Yoon and C. A. Shoemaker, "Comparison of optimization methods for ground-water bioremediation," *Journal of Water Resources Planning and Management*, vol. 125, no. 1, pp. 54–63, 1999.

[26] G. A. Gray, T. G. Kolda, K. Sale, and M. M. Young, "Optimizing an empirical scoring function for transmembrane protein structure determination," *INFORMS Journal on Computing*, vol. 16, no. 4, pp. 406–418, 2004.

[27] A. L. Marsden, J. A. Feinstein, and C. A. Taylor, "A computational framework for derivative-free optimization of cardiovascular geometries," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 21, pp. 1890–1905, 2008.

[28] M. C. Bartholomew-Biggs, S. C. Parkhurst, and S. P. Wilson, "Using DIRECT to solve an aircraft routing problem," *Computational Optimization and Applications*, vol. 21, no. 3, pp. 311–323, 2002.

[29] T. Levina, Y. Levin, J. McGill, and M. Nediak, "Dynamic pricing with online learning and strategic consumers: An application of the aggregating algorithm," *Operations Research*, vol. 57, no. 2, pp. 327–341, 2009.

[30] S. Nilchan and C. Pantelides, "On the optimisation of periodic adsorption processes," *Adsorption*, vol. 4, no. 2, pp. 113–147, 1998.

[31] Y. Kawajiri and L. T. Biegler, "Optimization strategies for simulated moving bed and powerfeed processes," *AIChE Journal*, vol. 52, no. 4, pp. 1343–1350, 2006.

[32] A. Agarwal, L. T. Biegler, and S. E. Zitney, "A superstructure-based optimal synthesis of PSA cycles for post-combustion co2 capture," *AIChE journal*, vol. 56, no. 7, pp. 1813–1828, 2010.

[33] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[34] A. Arora, I. Bajaj, S. S. Iyer, and M. M. F. Hasan, "Optimal synthesis of periodic sorption enhanced reaction processes with application to hydrogen production," *Computers & Chemical Engineering*, vol. 115, pp. 89–111, 2018.

[35] A. Arora, S. S. Iyer, I. Bajaj, and M. M. F. Hasan, "Optimal methanol production via sorption enhanced reaction process," *Industrial & Engineering Chemistry Research*, 2018.

[36] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to derivative-free optimization*, vol. 8. SIAM, 2009.

[37] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," *Journal of Global Optimization*, vol. 56, no. 3, pp. 1247–1293, 2013.

[38] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[39] V. Torczon, "On the convergence of pattern search algorithms," *SIAM Journal on Optimization*, vol. 7, no. 1, pp. 1–25, 1997.

[40] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.

[41] W. Huyer and A. Neumaier, "Global optimization by multilevel coordinate search," *Journal of Global Optimization*, vol. 14, no. 4, pp. 331–355, 1999.

[42] R. L. Smith, "Efficient monte carlo procedures for generating points uniformly distributed over bounded regions," *Operations Research*, vol. 32, no. 6, pp. 1296–1308, 1984.

[43] C. J. Bélisle, H. E. Romeijn, and R. L. Smith, "Hit-and-run algorithms for generating multivariate distributions," *Mathematics of Operations Research*, vol. 18, no. 2, pp. 255–266, 1993.

[44] N. Hansen, "The CMA evolution strategy: A tutorial," *arXiv preprint arXiv:1604.00772*, 2016.

[45] S.-K. S. Fan and E. Zahara, "A hybrid simplex search and particle swarm optimization for unconstrained optimization," *European Journal of Operational Research*, vol. 181, no. 2, pp. 527–548, 2007.

[46] E. H. Aarts and P. J. Van Laarhoven, "Statistical cooling: A general approach to combinatorial optimization problems.," *Philips J. Res.*, vol. 40, no. 4, pp. 193–226, 1985.

[47] A. D. Bethke, *Genetic algorithms as function optimizers*. PhD thesis, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, 1978.

[48] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pp. 39–43, IEEE, 1995.

[49] B. O. Shubert, "A sequential method seeking the global maximum of a function," *SIAM Journal on Numerical Analysis*, vol. 9, no. 3, pp. 379–388, 1972.

[50] J. D. Pintér, *Global optimization in action: continuous and Lipschitz optimization: algorithms, implementations and applications*, vol. 6. Springer Science & Business Media, 2013.

[51] P. Gilmore and C. T. Kelley, "An implicit filtering algorithm for optimization of functions with many local minima," *SIAM Journal on Optimization*, vol. 5, no. 2, pp. 269–285, 1995.

[52] S. M. Wild, R. G. Regis, and C. A. Shoemaker, "ORBIT: Optimization by radial basis function interpolation in trust-regions," *SIAM Journal on Scientific Computing*, vol. 30, no. 6, pp. 3197–3219, 2008.

[53] M. J. D. Powell, "The BOBYQA algorithm for bound constrained optimization without derivatives," *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, 2009.

[54] A. R. Conn, K. Scheinberg, and L. N. Vicente, "Global convergence of general derivative-free trust-region algorithms to first-and second-order critical points," *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 387–415, 2009.

[55] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[56] A. J. Booker, J. Dennis Jr, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset, "A rigorous framework for optimization of expensive functions by surrogates," *Structural optimization*, vol. 17, no. 1, pp. 1–13, 1999.

[57] W. Huyer and A. Neumaier, "SNOBFIT–stable noisy optimization by branch and fit," *ACM Transactions on Mathematical Software (TOMS)*, vol. 35, no. 2, p. 9, 2008.

[58] R. Oeuvray and M. Bierlaire, "A new derivative-free algorithm for the medical image registration problem," *International Journal of Modelling and Simulation*, vol. 27, no. 2, pp. 115–124, 2007. cited By 14.

[59] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Statistical science*, pp. 409–423, 1989.

[60] H.-M. Gutmann, "A radial basis function method for global optimization," *Journal of Global Optimization*, vol. 19, no. 3, pp. 201–227, 2001.

[61] R. R. Barton, "Metamodeling: a state of the art review," in *Simulation Conference Proceedings, 1994. Winter*, pp. 237–244, IEEE, 1994.

[62] A. L. Custódio and L. N. Vicente, "Using sampling and simplex derivatives in pattern search methods," *SIAM Journal on Optimization*, vol. 18, no. 2, pp. 537–555, 2007.

[63] G. Liuzzi and S. Lucidi, "A derivative-free algorithm for inequality constrained nonlinear programming via smoothing of an ℓ_∞ penalty function," *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 1–29, 2009.

[64] G. Liuzzi, S. Lucidi, and M. Sciandrone, “Sequential penalty derivative-free methods for nonlinear constrained optimization,” *SIAM Journal on Optimization*, vol. 20, no. 5, pp. 2614–2635, 2010.

[65] M. Diniz-Ehrhardt, J. Martínez, and L. G. Pedroso, “Derivative-free methods for nonlinear programming with general lower-level constraints,” *Computational & Applied Mathematics*, vol. 30, no. 1, pp. 19–52, 2011.

[66] C. Audet and J. E. Dennis Jr, “Mesh adaptive direct search algorithms for constrained optimization,” *SIAM Journal on Optimization*, vol. 17, no. 1, pp. 188–217, 2006.

[67] C. Audet and J. E. Dennis Jr, “A progressive barrier for derivative-free nonlinear programming,” *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 445–472, 2009.

[68] C. Audet and J. E. Dennis Jr, “A pattern search filter method for nonlinear programming without derivatives,” *SIAM Journal on Optimization*, vol. 14, no. 4, pp. 980–1010, 2004.

[69] M. J. Powell, “A direct search optimization method that models the objective and constraint functions by linear interpolation,” in *Advances in optimization and numerical analysis*, pp. 51–67, Springer, 1994.

[70] F. Augustin and Y. Marzouk, “NOWPAC: a provably convergent derivative-free nonlinear optimizer with path-augmented constraints,” *arXiv preprint arXiv:1403.1931*, 2014.

[71] M. B. Arouxét, N. E. Echebest, and E. A. Pilotta, “Inexact restoration method for nonlinear optimization without derivatives,” *Journal of Computational and Applied Mathematics*, vol. 290, pp. 26–43, 2015.

[72] C. Audet, A. R. Conn, S. Le Digabel, and M. Peyrega, “A progressive barrier derivative-free trust-region algorithm for constrained optimization,” tech. rep., Technical Report G-2016-49, Les cahiers du GERAD, 2016.

[73] P. R. Sampaio and P. L. Toint, “A derivative-free trust-funnel method for equality-constrained nonlinear optimization,” *Computational Optimization and Applications*, vol. 61, no. 1, pp. 25–49, 2015.

[74] N. I. Gould and P. L. Toint, "Nonlinear programming without a penalty function or a filter," *Mathematical Programming*, vol. 122, no. 1, pp. 155–196, 2010.

[75] R. G. Regis and S. M. Wild, "CONORBIT: constrained optimization by radial basis function interpolation in trust regions," *Optimization Methods and Software*, vol. 32, no. 3, pp. 552–580, 2017.

[76] J. P. Eason and L. T. Biegler, "A trust region filter method for glass box/black box optimization," *AIChE Journal*, 2016.

[77] J. P. Eason and L. T. Biegler, "Advanced trust region optimization strategies for glass box/black box models," *AIChE Journal*, vol. 64, no. 11, pp. 3934–3943, 2018.

[78] F. Boukouvala and C. A. Floudas, "ARGONAUT: Algorithms for global optimization of constrained grey-box computational problems," *Optimization Letters*, vol. 11, no. 5, pp. 895–913, 2017.

[79] Z. Wang and M. Ierapetritou, "A novel feasibility analysis method for black-box processes using a radial basis function adaptive sampling approach," *AIChE Journal*, vol. 63, no. 2, pp. 532–550, 2017.

[80] C. A. Kieslich, F. Boukouvala, and C. A. Floudas, "Optimization of black-box problems using smolyak grids and polynomial approximations," *Journal of Global Optimization*, pp. 1–25, 2018.

[81] C. A. Henao and C. T. Maravelias, "Surrogate-based superstructure optimization framework," *AIChE Journal*, vol. 57, no. 5, pp. 1216–1232, 2011.

[82] P. Balasubramanian, I. Bajaj, and M. M. F. Hasan, "Simulation and optimization of reforming reactors for carbon dioxide utilization using both rigorous and reduced models," *Journal of CO2 Utilization*, vol. 23, pp. 80–104, 2018.

[83] A. Cozad, N. V. Sahinidis, and D. C. Miller, "A combined first-principles and data-driven approach to model building," *Computers & Chemical Engineering*, vol. 73, pp. 116–127, 2015.

[84] S. S. Garud, I. A. Karimi, and M. Kraft, "LEAPS2: Learning based evolutionary assistive paradigm for surrogate selection," *Computers & Chemical Engineering*, vol. 119, pp. 352–370, 2018.

[85] A. Nuchitprasittichai and S. Cremaschi, "An algorithm to determine sample sizes for optimization with artificial neural networks," *AIChE Journal*, vol. 59, no. 3, pp. 805–812, 2013.

[86] J. Straus and S. Skogestad, "Surrogate model generation using self-optimizing variables," *Computers & Chemical Engineering*, vol. 119, pp. 143–151, 2018.

[87] A. P. Tran and C. Georgakis, "On the estimation of high-dimensional surrogate models of steady-state of plant-wide processes characteristics," *Computers & Chemical Engineering*, vol. 116, pp. 56–68, 2018.

[88] S. Le Digabel, "Algorithm 909: Nomad: Nonlinear optimization with the mads algorithm," *ACM Transactions on Mathematical Software (TOMS)*, vol. 37, no. 4, p. 44, 2011.

[89] K. Scheinberg, "Manual for fortran software package DFO v2. 0," 2003.

[90] M. J. D. Powell, "The NEWUOA software for unconstrained optimization without derivatives," in *Large-scale nonlinear optimization*, pp. 255–297, Springer, 2006.

[91] A. I. F. Vaz and L. N. Vicente, "A particle swarm pattern search method for bound constrained global optimization," *Journal of Global Optimization*, vol. 39, no. 2, pp. 197–219, 2007.

[92] L. Ingber, A. Petraglia, M. R. Petraglia, and M. A. S. Machado, "Adaptive simulated annealing," in *Stochastic global optimization and its applications with fuzzy adaptive simulated annealing*, pp. 33–62, Springer, 2012.

[93] A. Dakota, "Multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis," *Sandia National Laboratories, SAND2010-2183*, 2009.

[94] T. Csendes, L. Pál, J. O. H. Sendin, and J. R. Banga, “The global optimization method revisited,” *Optimization Letters*, vol. 2, no. 4, pp. 445–454, 2008.

[95] G. Di Pillo, G. Liuzzi, S. Lucidi, V. Piccialli, and F. Rinaldi, “A direct-type approach for derivative-free constrained global optimization,” *Computational Optimization and Applications*, vol. 65, no. 2, pp. 361–397, 2016.

[96] G. Liuzzi, S. Lucidi, V. Piccialli, and A. Sotgiu, “A magnetic resonance device designed via global optimization techniques,” *Mathematical Programming*, vol. 101, no. 2, pp. 339–364, 2004.

[97] C. Audet and W. Hare, *Derivative-free and blackbox optimization*. Springer, 2017.

[98] F. Boukouvala, R. Misener, and C. A. Floudas, “Global optimization advances in mixed-integer nonlinear programming, minlp, and constrained derivative-free optimization, cdfo,” *European Journal of Operational Research*, vol. 252, no. 3, pp. 701–727, 2016.

[99] A. Bhosekar and M. Ierapetritou, “Advances in surrogate based modeling, feasibility analysis, and optimization: A review,” *Computers & Chemical Engineering*, vol. 108, pp. 250–267, 2018.

[100] S. S. Garud, I. A. Karimi, and M. Kraft, “Design of Computer Experiments: A Review,” *Computers & Chemical Engineering*, vol. 106, 2017.

[101] B. K. Olsen, F. Castellino, and A. D. Jensen, “Modeling deactivation of catalysts for selective catalytic reduction of no x by kcl aerosols,” *Industrial & Engineering Chemistry Research*, vol. 56, no. 45, pp. 13020–13033, 2017.

[102] R. Horst and H. Tuy, *Global optimization: Deterministic approaches*. Springer Science & Business Media, 2013.

[103] R. Misener and C. A. Floudas, “ANTIGONE: Algorithms for coNTinuous/Integer Global Optimization of Nonlinear Equations,” *Journal of Global Optimization*, vol. 59, no. 2-3, pp. 503–526, 2014.

[104] M. Tawarmalani and N. V. Sahinidis, “Global optimization of mixed-integer nonlinear programs: A theoretical and computational study,” *Mathematical programming*, vol. 99, no. 3, pp. 563–591, 2004.

[105] M. M. F. Hasan, “An edge-concave underestimator for the global optimization of twice-differentiable nonconvex problems,” *Journal of Global Optimization*, vol. 71, no. 4, pp. 735–752, 2018.

[106] F. Tardella, “On the existence of polyhedral convex envelopes,” in *Frontiers in global optimization*, pp. 563–573, Springer, 2004.

[107] Y. A. Guzman, M. F. Hasan, and C. A. Floudas, “Performance of convex underestimators in a branch-and-bound framework,” *Optimization Letters*, vol. 10, no. 2, pp. 283–308, 2016.

[108] W. Hofschuster and W. Krämer, “C-xsc 2.0–a c++ library for extended scientific computing,” in *Numerical software with result verification*, pp. 15–35, Springer, 2004.

[109] C. G. Moles, P. Mendes, and J. R. Banga, “Parameter estimation in biochemical pathways: a comparison of global optimization methods,” *Genome Research*, vol. 13, no. 11, pp. 2467–2474, 2003.

[110] K. G. Gadkar, R. Gunawan, and F. J. Doyle, “Iterative approach to model identification of biological networks,” *BMC Bioinformatics*, vol. 6, no. 1, p. 155, 2005.

[111] S. Katare, A. Bhan, J. M. Caruthers, W. N. Delgass, and V. Venkatasubramanian, “A hybrid genetic algorithm for efficient parameter estimation of large kinetic models,” *Computers & Chemical Engineering*, vol. 28, no. 12, pp. 2569–2581, 2004.

[112] V. Ramadesigan, V. Boovaragavan, J. C. Pirkle, and V. R. Subramanian, “Efficient reformulation of solid-phase diffusion in physics-based lithium-ion battery models,” *Journal of The Electrochemical Society*, vol. 157, no. 7, pp. A854–A860, 2010.

[113] W. Hu, B. Lowry, and A. Varma, “Kinetic study of glycerol oxidation network over Pt–Bi/C catalyst,” *Applied Catalysis B: Environmental*, vol. 106, no. 1-2, pp. 123–132, 2011.

[114] K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control*, vol. 40. Prentice Hall New Jersey, 1996.

[115] D. E. Kirk, *Optimal control theory: An introduction*. Courier Corporation, 2012.

[116] C. Wu and K. Teo, "Global impulsive optimal control computation," *Journal of Industrial & Management Optimization*, vol. 2, no. 4, pp. 435–450, 2006.

[117] C. G. Moles, J. R. Banga, and K. Keller, "Solving nonconvex climate control problems: pitfalls and algorithm performances," *Applied Soft Computing*, vol. 5, no. 1, pp. 35–44, 2004.

[118] T. Miri, A. Tsoukalas, S. Bakalis, E. Pistikopoulos, B. Rustem, and P. Fryer, "Global optimization of process conditions in batch thermal sterilization of food," *Journal of Food Engineering*, vol. 87, no. 4, pp. 485–494, 2008.

[119] R. Luus and D. Cormack, "Multiplicity of solutions resulting from the use of variational methods in optimal control problems," *The Canadian Journal of Chemical Engineering*, vol. 50, no. 2, pp. 309–311, 1972.

[120] W. R. Esposito and C. A. Floudas, "Deterministic global optimization in nonlinear optimal control problems," *Journal of Global Optimization*, vol. 17, no. 1-4, pp. 97–126, 2000.

[121] W. R. Esposito and C. A. Floudas, "Global optimization for the parameter estimation of differential-algebraic systems," *Industrial & Engineering Chemistry Research*, vol. 39, no. 5, pp. 1291–1310, 2000.

[122] C. S. Adjiman, S. Dallwig, C. A. Floudas, and A. Neumaier, "A global optimization method, αBB, for general twice-differentiable constrained NLPs—I. Theoretical advances," *Computers & Chemical Engineering*, vol. 22, no. 9, pp. 1137–1158, 1998.

[123] I. Papamichail and C. S. Adjiman, "A rigorous global optimization algorithm for problems with ordinary differential equations," *Journal of Global Optimization*, vol. 24, no. 1, pp. 1–33, 2002.

[124] I. Papamichail and C. S. Adjiman, “Global optimization of dynamic systems,” *Computers & Chemical Engineering*, vol. 28, no. 3, pp. 403–415, 2004.

[125] B. Chachuat and M. Latifi, “A new approach in deterministic global optimisation of problems with ordinary differential equations,” in *Frontiers in Global Optimization*, pp. 83–108, Springer, 2004.

[126] A. B. Singer and P. I. Barton, “Global solution of optimization problems with parameter-embedded linear dynamic systems,” *Journal of Optimization Theory and Applications*, vol. 121, no. 3, pp. 613–646, 2004.

[127] A. B. Singer and P. I. Barton, “Bounding the solutions of parameter dependent nonlinear ordinary differential equations,” *SIAM Journal on Scientific Computing*, vol. 27, no. 6, pp. 2167–2182, 2006.

[128] A. B. Singer and P. I. Barton, “Global optimization with nonlinear ordinary differential equations,” *Journal of Global Optimization*, vol. 34, no. 2, pp. 159–190, 2006.

[129] J. K. Scott, B. Chachuat, and P. I. Barton, “Nonlinear convex and concave relaxations for the solutions of parametric odes,” *Optimal Control Applications and Methods*, vol. 34, no. 2, pp. 145–163, 2013.

[130] J. K. Scott and P. I. Barton, “Bounds on the reachable sets of nonlinear control systems,” *Automatica*, vol. 49, no. 1, pp. 93–100, 2013.

[131] J. K. Scott and P. I. Barton, “Interval bounds on the solutions of semi-explicit index-one DAEs. Part 2: computation,” *Numerische Mathematik*, vol. 125, no. 1, pp. 27–60, 2013.

[132] M. E. Villanueva, B. Houska, and B. Chachuat, “Unified framework for the propagation of continuous-time enclosures for parametric nonlinear ODEs,” *Journal of Global Optimization*, vol. 62, no. 3, pp. 575–613, 2015.

[133] B. Houska, M. E. Villanueva, and B. Chachuat, “Stable set-valued integration of nonlinear dynamic systems using affine set-parameterizations,” *SIAM Journal on Numerical Analysis*, vol. 53, no. 5, pp. 2307–2328, 2015.

[134] S. M. Harwood, J. K. Scott, and P. I. Barton, "Bounds on reachable sets using ordinary differential equations with linear programs embedded," *IMA Journal of Mathematical Control and Information*, vol. 33, no. 2, pp. 519–541, 2015.

[135] S. M. Harwood and P. I. Barton, "Efficient polyhedral enclosures for the reachable set of nonlinear control systems," *Mathematics of Control, Signals, and Systems*, vol. 28, no. 1, p. 8, 2016.

[136] M. E. Villanueva, R. Quirynen, M. Diehl, B. Chachuat, and B. Houska, "Robust MPC via min–max differential inequalities," *Automatica*, vol. 77, pp. 311–321, 2017.

[137] S. M. Harwood and P. I. Barton, "Affine relaxations for the solutions of constrained parametric ordinary differential equations," *Optimal Control Applications and Methods*, vol. 39, no. 2, pp. 427–448, 2018.

[138] Y. Lin and M. A. Stadtherr, "Deterministic global optimization for parameter estimation of dynamic systems," *Industrial & Engineering Chemistry Research*, vol. 45, no. 25, pp. 8438–8448, 2006.

[139] Y. Lin and M. A. Stadtherr, "Deterministic global optimization of nonlinear dynamic systems," *AIChE Journal*, vol. 53, no. 4, pp. 866–875, 2007.

[140] Y. Zhao and M. A. Stadtherr, "Rigorous global optimization for dynamic systems subject to inequality path constraints," *Industrial & Engineering Chemistry Research*, vol. 50, no. 22, pp. 12678–12693, 2011.

[141] F. Tardella, "On a class of functions attaining their maximum at the vertices of a polyhedron," *Discrete applied mathematics*, vol. 22, no. 2, pp. 191–195, 1988.

[142] C. A. Meyer and C. A. Floudas, "Convex envelopes for edge-concave functions," *Mathematical Programming*, vol. 103, no. 2, pp. 207–224, 2005.

[143] W. Rudin, *Principles of mathematical analysis*, vol. 3. McGraw-Hill New York, 1976.

[144] L. H. Loomis and S. Sternberg, *Advanced Calculus: Revised*. World Scientific Publishing Company, 2014.

[145] R. E. Moore, *Methods and applications of interval analysis*, vol. 2. SIAM, 1979.

[146] L. Pontryagin, "Chapter 4 - Existence Theorems," in *Ordinary Differential Equations* (L. Pontryagin, ed.), pp. 150 – 199, Pergamon, 1962.

[147] V. Lakshmikantham and S. Leela, *Differential and Integral Inequalities: Theory and Applications: Volume I: Ordinary Differential Equations*. Academic press, 1969.

[148] H. S. Ryoo and N. V. Sahinidis, "A branch-and-reduce approach to global optimization," *Journal of Global Optimization*, vol. 8, no. 2, pp. 107–138, 1996.

[149] Y. Puranik and N. V. Sahinidis, "Domain reduction techniques for global NLP and MINLP optimization," *Constraints*, vol. 22, no. 3, pp. 338–376, 2017.

[150] M. A. Abramson, C. Audet, G. Couture, J. E. Dennis Jr, S. Le Digabel, and C. Tribes, "The NOMAD project," 2011.

[151] I. Bajaj, S. S. Iyer, and M. M. F. Hasan, "A trust region-based two phase algorithm for constrained black-box and grey-box optimization with infeasible initial point," *Computers & Chemical Engineering*, vol. 116, pp. 306–321, 2018.

[152] S. M. Rump, "INTLAB—INTerval LABoratory," in *Developments in Reliable Computing*, pp. 77–104, Springer, 1999.

[153] I. B. Tjoa and L. T. Biegler, "Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems," *Industrial & Engineering Chemistry Research*, vol. 30, no. 2, pp. 376–385, 1991.

[154] W. W. Hogan, "Point-to-set maps in mathematical programming," *SIAM Review*, vol. 15, no. 3, pp. 591–603, 1973.

[155] A. M. Geoffrion, "Primal resource-directive approaches for optimizing nonlinear decomposable systems," *Operations Research*, vol. 18, no. 3, pp. 375–403, 1970.

[156] W. I. Zangwill, *Nonlinear programming: a unified approach.* Prentice-Hall, 1969.

[157] G. B. Dantzig, J. Folkman, and N. Shapiro, “On the continuity of the minimum set of a continuous function,” *Journal of Mathematical Analysis and Applications*, vol. 17, no. 3, pp. 519–548, 1967.

[158] J. P. Evans and F. J. Gould, “Stability in nonlinear programming,” *Operations Research*, vol. 18, no. 1, pp. 107–118, 1970.

[159] R. Meyer, “The validity of a family of optimization methods,” *SIAM Journal on Control*, vol. 8, no. 1, pp. 41–54, 1970.

[160] B. Gollan, “On the marginal function in nonlinear programming,” *Mathematics of Operations Research*, vol. 9, no. 2, pp. 208–221, 1984.

[161] A. V. Fiacco and J. Kyparisis, “Convexity and concavity properties of the optimal value function in parametric nonlinear programming,” *Journal of optimization theory and applications*, vol. 48, no. 1, pp. 95–126, 1986.

[162] A. V. Fiacco, *Introduction to sensitivity and stability analysis in nonlinear programming.* Academic Press, New York, 1984.

[163] A. V. Fiacco, “Sensitivity analysis for nonlinear programming using penalty methods,” *Mathematical Programming*, vol. 10, no. 1, pp. 287–311, 1976.

[164] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[165] F. Borrelli, M. Baotić, A. Bemporad, and M. Morari, “Dynamic programming for constrained optimal control of discrete-time linear hybrid systems,” *Automatica*, vol. 41, no. 10, pp. 1709–1721, 2005.

[166] V. Dua, N. A. Bozinis, and E. N. Pistikopoulos, “A multiparametric programming approach for mixed-integer quadratic engineering problems,” *Computers & Chemical Engineering*, vol. 26, no. 4, pp. 715–733, 2002.

[167] J. Kyparisis, “Sensitivity analysis for nonlinear programs and variational inequalities with nonunique multipliers,” *Mathematics of Operations Research*, vol. 15, no. 2, pp. 286–298, 1990.

[168] K. Jittorntrum, “Solution point differentiability without strict complementarity in nonlinear programming,” *Sensitivity, Stability and Parametric Analysis*, pp. 127–138, 1984.

[169] A. Shapiro, “Sensitivity analysis of nonlinear programs and differentiability properties of metric projections,” *SIAM Journal on Control and Optimization*, vol. 26, no. 3, pp. 628–645, 1988.

[170] J. Gauvin and R. Janin, “Directional behaviour of optimal solutions in nonlinear mathematical programming,” *Mathematics of Operations Research*, vol. 13, no. 4, pp. 629–649, 1988.

[171] A. Auslender and R. Cominetti, “First and second order sensitivity analysis of nonlinear programs under directional constraint qualification conditions,” *Optimization*, vol. 21, no. 3, pp. 351–363, 1990.

[172] P. Conejo, E. W. Karas, and L. Pedroso, “A trust-region derivative-free algorithm for constrained optimization,” *Optimization Methods and Software*, vol. 30, no. 6, pp. 1126–1145, 2015.

[173] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust region methods*, vol. 1. SIAM, 2000.

[174] R. G. Regis, “Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points,” *Engineering Optimization*, vol. 46, no. 2, pp. 218–243, 2014.

[175] A. R. Conn, K. Scheinberg, and L. N. Vicente, “Geometry of interpolation sets in derivative free optimization,” *Mathematical Programming*, vol. 111, no. 1-2, pp. 141–172, 2008.

[176] M. J. D. Powell, “UOBYQA: unconstrained optimization by quadratic approximation,” *Mathematical Programming*, vol. 92, no. 3, pp. 555–582, 2002.

[177] M. Björkman and K. Holmström, "Global optimization of costly nonconvex functions using radial basis functions," *Optimization and Engineering*, vol. 1, no. 4, pp. 373–397, 2000.

[178] A. S. Drud, "CONOPT—a large-scale grg code," *ORSA Journal on computing*, vol. 6, no. 2, pp. 207–216, 1994.

[179] M. J. D. Powell, "On the convergence of trust region algorithms for unconstrained minimization without derivatives," *Computational Optimization and Applications*, vol. 53, no. 2, pp. 527–555, 2012.

[180] P. S. Ferreira, E. W. Karas, and M. Sachine, "A globally convergent trust-region algorithm for unconstrained derivative-free optimization," *Computational and Applied Mathematics*, vol. 34, no. 3, pp. 1075–1103, 2015.

[181] A. R. Conn, N. Gould, A. Sartenaer, and P. L. Toint, "Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints," *SIAM Journal on Optimization*, vol. 3, no. 1, pp. 164–221, 1993.

[182] M. Tawarmalani and N. V. Sahinidis, "A polyhedral branch-and-cut approach to global optimization," *Mathematical Programming*, vol. 103, pp. 225–249, 2005.

[183] Y. Lin and L. Schrage, "The global solver in the LINDO API," *Optimization Methods & Software*, vol. 24, no. 4-5, pp. 657–668, 2009.

[184] S. Le Digabel, "Algorithm 909: NOMAD: Nonlinear optimization with the mads algorithm," *ACM Trans. Math. Softw.*, vol. 37, pp. 44:1–44:15, Feb. 2011.

[185] K. Scheinberg and P. L. Toint, "Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization," *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3512–3532, 2010.

[186] M. J. D. Powell, *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation*, pp. 51–67. Dordrecht: Springer Netherlands, 1994.

[187] I. Bajaj and M. M. F. Hasan, "Effective sampling, modeling and optimization of constrained black-box problems," *Computer Aided Process Engineering*, vol. 38, pp. 553–558, 2016.

[188] F. J. Hickernell, *Lattice rules: how well do they measure up?* Springer, 1998.

[189] GlobalLib, "Global library," 2015. http://www.gamsworld.org/global/globallib.htm.

[190] J. J. Moré and S. M. Wild, "Benchmarking derivative-free optimization algorithms," *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 172–191, 2009.

[191] A. I. Stankiewicz, J. A. Moulijn, *et al.*, "Process intensification: transforming chemical engineering," *Chemical engineering progress*, vol. 96, no. 1, pp. 22–34, 2000.

[192] S. E. Demirel, J. Li, and M. M. F. Hasan, "Systematic process intensification using building blocks," *Computers & Chemical Engineering*, vol. 105, pp. 2–38, 2017.

[193] A. Gupta, S. Bhartiya, and P. Nataraj, "A novel approach to multiparametric quadratic programming," *Automatica*, vol. 47, no. 9, pp. 2112–2117, 2011.

[194] J. SpjøTvold, E. C. Kerrigan, C. N. Jones, P. TøNdel, and T. A. Johansen, "On the facet-to-facet property of solutions to convex parametric quadratic programs," *Automatica*, vol. 42, no. 12, pp. 2209–2214, 2006.

[195] R. Oberdieck and E. N. Pistikopoulos, "Explicit hybrid model-predictive control: The exact solution," *Automatica*, vol. 58, pp. 152–159, 2015.