A Doctoral Thesis on

GLOBAL OPTIMIZATION OF CHEMICAL PROCESSES

SUBMITTED TO THE

University of Pune

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

(In Chemical Engineering)

BY

SATISH R. INAMDAR

M.Tech. (Chemical Engineering) IIT Bombay; Ph.D. (Chemistry)
National Chemical Laboratory, Pune (Registered with University of Pune).

This research work was carried out with place of research
Chemical Engineering & Process Development Division
National Chemical Laboratory, Pune – 411 008, INDIA.

September 2012

DEDICATED TO MY

PARENTS, FAMILY AND TEACHERS

CERTIFICATE OF THE GUIDE

CERTIFIED that the work incorporated in the thesis **GLOBAL OPTIMIZATION OF CHEMICAL PROCESSES** submitted by Mr. Satish R. Inamdar was carried out by the candidate under my supervision/ guidance. Such material as has been obtained from other sources has been duly acknowledged in the thesis.

(B.D. Kulkarni)
(Supervisor/ Research Guide)

DECLARATION BY THE CANDIDATE

I declare that the thesis entitled GLOBAL OPTIMIZATION OF CHEMICAL PROCESSES submitted by me for the degree of Doctor of Philosophy is the record of work carried out by me during the period from July 2005 to September 2012 under the guidance of Dr. B.D. Kulkarni and has not formed the basis for the award of any degree, diploma, associateship, fellowship, titles in this or any other University or other institution of Higher learning.

I further declare that the material obtained from other sources has been duly acknowledged in the thesis.

(Satish R. Inamdar) Date: 20 September 2012

Acknowledgements

I remain indebted to two people for learning many things about conducting a good quality research in chemical engineering. I have learnt several things from Prof. Satish J. Parulekar, Professor and Acting Chair, Chemical & Biological Engineering Department, Illinois Institute of Technology, Chicago, IL, USA. And this is about the presentation values in writing a paper, adding strengths, and identifying weaknesses in work and strengthening the same. His active support and frequent supervision in doing computations using optimization algorithms has changed my views and has helped me in structuring the ideas in the right ways. I also remain indebted to Prof. Iftekar A. Karimi, Professor, Chemical & Biomolecular Engineering Department, National University of Singapore for almost training me to work on a research plan from scratch and to develop it up into a full-text paper. I learnt how to start with an idea having any newness to be made into a research paper from him. Lastly, I will always and ever remain indebted to my Resrach Guide, Dr. B.D. Kulkarni, Dean CSIR Academy, National Chemical Laboratory, Pune for his unstinted support while I worked with clear directions from him to find discrepancies in arguments to be made in expressing any ideas leading to derivation of analytical results. He corrected me at various stages in doing every single problem and encouraged me to take up fundamental concepts as topics of research than opting for conventional setting of engineering research. This work could not have been possible without his support and directions & advice given to me from time to time.

Before I conclude this thanksgiving, I will also remain grateful to my colleagues in the Chemical Engineering Department of Vishwakarma Institute of Technology, Pune for the cooperation I received from them. Special thanks go to Prof. Rajesh M. Jalnekar, Director, Vishwakarma Institute of Technology, who has been so kind to me and provided me all warmth and support that I could find time to do research.

Finally not to forget my father and family members (wife and two children), who have managed all my misgivings, general problems and supported me while giving me extra time to complete my work even till late night. I will remain grateful to them for all support, advice and care received during the past few years while I did my research work and wrote papers.

(Satish R. Inamdar)

Abstract

In this proposed research work, we focus upon the problem of global optimization of nonlinear process systems that arise in chemical engineering science. Our primary objective in this study is to devise new methods of global optimization (including those which are variants of existing optimization methods for global optimization).

Chapter I is an Introduction to research work being planned and done and tells about how the problem selected was identified as one which was solved as the research problem satisfactorily. The contents of Chapter I are summarized here, which will also give an idea about the topic of research and how identification of problems defines scope of work to be done.

Optimization of nonlinear chemical process systems has been a topic of interest for many years. The problems faced during optimization of such systems can be the presence of multiple optimum solutions, stability and convergence of algorithms. If an optimization program has highly nonlinear equations as constraints, we may experience slower convergence.

The problem due to nonlinearity of constraints also implies problems in obtaining accuracy of solution after computations. Also in search range of decision variables several possible local optimum solutions may exist and it will become difficult to obtain the required global optimum solution. Under such circumstances we divide the global optimization method into two parts, viz. global search method followed with a local search method.

Thus the idea is to devise a global search method which converges closer to the true optimum solution when a successive linear programming approximation is used. This converged solution serves as a guess in that a local search method such as trust region finds an accurate enough local optimum solution using nonlinear objective function and constraints. Again the global search method can be composed of two parts. First one gives a pattern search control in various search segments of total span of decision variables (box constraints); while the other uses an approximation as a box constrained linear program to come close to true optimum solution. In this work, we are interested in developing the latter part of global search method i.e. fast converging cutting plane modification.

In Chapter II of the thesis, we present a new variant of Kelley's cutting plane algorithm, which was found to have a higher rate of convergence than its predecessor. The observations made during numerical experience of applying the cutting plane algorithm to nonlinear optimization programs to standard examples in textbooks resulted in forming a concrete idea and identification of new directions for improvement in this existing method.

Thus, we are motivated to extend the basic result of Kelley's cutting planes to form a new cutting plane variant in order to improve convergence of numerical algorithm. The new alternative cut uses a strategy to achieve this objective and employs the new cut and we may term it here in this work as a variant of Kelley's cutting plane algorithm.

One possible approach for deriving cutting planes is to change the right hand sides of linear constraints. Our approach is conceptually different from this and results in a superlinear

rate of convergence vis-a-vis the sublinear convergence rate of the Kelley's cutting plane algorithm. We demonstrate the performance of these new cuts using several examples.

In our work, we have presented a new variant of Kelley's cutting plane method, which we termed as a sharp cut. This new variant is in reality an intersection of three cuts and is an outcome of experiments based on certain observations, which helped in forming clear objectives. The convergence improved almost becoming superlinear after a couple of iterations. Hence, there is a reason to believe that an improvement which is more than satisfactory was seen from numerical computations, when the algorithm was applied to a number of toy examples.

It was observed that in few iterations of the first sharp cut the signs of gradient vector reverses and the second sharp cut which is almost parallel to hyper tangent to unique global optimum solution in search range gives an extra cut and accelerates the rate of convergence to first super linear rate and then to super super linear rate and requires less number of iterations. The algorithm now reaches the stationary point as defined in theory and is at least twice as fast as than Kelley's algorithm.

In this chapter III, we discuss an application of a cutting plane variant presented by authors termed as a sharp cut algorithm. This cutting plane variant was found to be superior than existing methods in literature. When a few standard nonconvex functions were tested, the sharp cut algorithm gave out all optimum solutions in monotone order. From the limited literature we searched, we found that the sharp cut algorithm has a unique feature of finding all multiple optimum solutions in very few iterations, when standard nonconvex functions are most violated constraints. In order to illustrate and test its applicability, we have applied this cutting plane variant to solve a problem of control and optimization. The example we have chosen is that of a nonisothermal CSTR, where no stable nodes were found for a limited parameter search done. In addition, we considered the scenario, when a few system parameters shift during operation. When we wish to find a new optimum steady state after these shifts occurs, we apply the cut, obtain all optimum solutions, find one corresponding to maximum conversion, and, again apply the optimal control which is based on Pontryagin's maximum principle. The results of numerical computations are presented.

In Chapter IV, the sharp cut algorithm, which was found to be superior than it's predecessors and performed well when applied to the toy examples was used to extend its applicability to stochastic optimal control. The local neighborhood is formed by plant system parameters and we first consider a case where free system parameters are varied. At every iteration of optimal control, we check for a condition that either parameter shifts occur or that the system has reached its destination i.e. final state, and that we need to find out which are further available states in this neighborhood, and, apply the sharp cut algorithm to a linear naive objective function to find a new set point from multiple local optimum solutions obtained. A statement of dynamic optimization for a linear (linearized) control system is derived and an optimal controller equation is obtained. Now we consider a stochastic variation within plant system is considered, and, the efficacy of this new control system is checked to verify its

applicability. Several results from numerical computations are presented at the end to illustrate the theoretical results for control and optimization.

In the Chapter V, the previous application of stochastic optimal control in Chapter IV is re-visited and a problem of nonlinear control is solved. We assume no variation in the feed concentration being input to the nonisothermal CSTR and design an optimal controller using an external cooling jacket. The sharp cut algorithm can be executed at every iteration to find next final state for optimal control for the operation of external and internal cooling facility. This gives rise to a statement of optimal control and optimization problem. This is expected to eliminate the disturbance entering the CSTR dynamics. However, if a weak sinusoidal variation occurs in the feed concentration with random variation around a bias caused as added component, it is preferable that internal cooling coils are inserted into vessel to absorb this extra release of heat. Although the problem seems to be acceptable as an academic exercise, for a pilot plant facility, such an arrangement can be possible. So we derive the necessary dynamics for internal cooling coils, derive necessary transfer function relations after linearization (which is a routine textbook exercise) and carry out model block synthesis that will provide temperature balance equation for stochastic optimal control. Once the stochastic differential equation is obtained, we can apply theoretical results in literature and derive necessary analytical results that can be used to apply the stochastic optimal control. The Results and Discussion Section discusses how this can be implemented and the advantages we gain from application of the same.

The Chapter VI concludes the thesis by making a few recommendations as possible directions for future research.

Global optimization of chemical processes

Satish R. Inamdar

September 25, 2012

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

In this work, we will focus our attention on devising computationally more efficient optimization algorithm for optimization of nonlinear chemical processes. The objective is to come closer to possible local optimum solution that local search method can converge upon the true optimum solution.

1.2 Problem identification

The problem of finding a new alternative method which is more efficient than the existing computing algorithms remains a frontier topic in research

on optimization fundamentals even today and will continue later.

One alternative measure which is one view presented in conventional literature on optimization, is to devise global search methods which come very close to true local optimum solutions in very few iterations. This optimum value obtained can be used as a starting point by any local search method (e.g. trust region method) to obtain the true optimum again in a very few iterations. This guarantees a very high rate of convergence and can be a useful method to solve many practical nonlinear optimization problems, and more specifically so the nonconvex programming problems.

The focus in this work is upon the cutting plane methods for two reasons. Firstly, the guess values are not required for the cutting plane algorithms and its variants published in literature. Secondly, as the cutting plane algorithm serves as a global search method that a local search method can converge upon true optimum solution, it offers another advantage. The cutting plane method uses the box constraints specified as a starting point and finds the optimum solution; hence it eliminates the possibility of requiring any pattern search method during this search.

This our aim in this work is to find a new variant of the cutting plane method, which has a higher rate of convergence and will have an ability to

solve the nonconvex programming problems. This forms the scope of work being done in this doctoral thesis and forms a motivation to take up this research problem.

1.3 Importance of cutting plane methods

The cutting planes are best known alternative to branch and bound and are very commonly used as a starting routine before a nonlinear programming (NLP) solver is called. The cutting plane code thus comes very close to the true numerical solution of a nonlinear optimization program, which is subject to nonlinear equality and inequality constraints.

These cutting plane methods are applied to a general convex continuous optimization program. The commonly known variants of cutting plane methods are Kelley's method, Kelley- Cheney- Goldstein method, and bundle methods. These variants are very popular and used to solve the non-differentiable convex minimization programs. In these non-differentiable programs, a convex objective function and its subgradient are evaluated efficiently; however, the usual gradient methods for differentiable optimization can not be used. In general, the cutting plane methods are used in several

commercial solvers.

The successive linear programming (SLP) method is an extension of the technique of linear programming, which allows the optimization of nonlinear programming problems through a series of linear approximations. When we apply the SLP algorithm, we start with an initial estimate of the optimal solution and iterate further. This method thus solves successive first order approximations or linearizations of the nonlinear program.

These linear constraints which get added to the constraint set, as SLP iterates, are often non-bounded. Also, the optimum solution may lie in some cases within the interior of the feasible region. The SLP method comes near to the true optimum solution and next a commonly used local search method such as the trust region method which has a feature of putting a bound on step bounding.

1.4 Aim and scope

The primary aim of this research being done is to devise a new cutting plane algorithm which will overcome the weakness of having a sublinear rate of convergence, which is observed in case of Kelley's cutting plane method

when initial computing was done using toy examples. Later we can apply the sharp cut algorithm to an example of optimization of performance of a nonisothermal continuously stirred tank reactor (CSTR).

Chapter 2

The sharp cut algorithm

Abstract

In this chapter, we introduce a new cutting plane algorithm which is computationally less expensive and more efficient than Kelley's algorithm. This new cutting plane algorithm uses an intersection cut of three types of cutting planes. We find from numerical results that the global search method formed using successive linear programming and a new intersection set is at least twice as fast as than Kelley's cutting planes. The necessary mathematical analysis and convergence theorem are provided. The key findings are illustrated via optimization of three CSTRs.

2.1 Introduction

Optimization of nonlinear process systems (Hoza and Stadtherr, 1993; Terlaky, 1996; Harjunkoski, Jain, and Grossman, 2000; Rajesh et. al., 2001; Bertsekas and Nedic, 2003; Lucia and Yang, 2004; Floudas et. al., 2005; Georgiorgis et. al., 2006; Miri et. al., 2008; Srinivasan, Biegler and Bonvin, 2008; Varma, 2008) has been of interest for many years. Some problems that one may face in optimizing highly nonlinear systems are multiple local optima (Mangaserian, 1969; Ravindran, Ragsdell and Reklaitis, 1983; Boyd and Vandenberghe, 2004), stability (Higham, 1996; Borkar and Meyn, 2000; Bertsekas and Nedic, 2003), feasible solution, and algorithmic convergence (Kelley, 1960; Byrd et. al., 2005). Successive linear programming has been used to solve many problems of interest in practice, especially those in refinery planning and scheduling. The basic idea is to solve a series o linear program arising from a linear approximation of the feasible region. Kelley (1960) proposed a cutting plane algorithm to improve the convergence of such a strategy. Cheney and Goldstein (1959) also arrived at a similar algorithm to find an optimum solution. Our goal in this work is to extend Kelley's cutting planes (Kelley, 1960) to form new cutting planes that improve the convergence of an SLP algorithm.

Various cuts have been used in the literature on optimization theory and methods. Goffin and Vial (1997) proposed shallow, deep, and very deep cuts in solving non-differentiable convex programs. The techniques build an increasingly refined polyhedral approximation of the optimum solution set. An analytic center cutting plane method was used to solve optimization problems. Two cuts were used again by Goffin and Vial (1996) when a new cutting plane and a new upper bound were introduced for the objective function at the same time. It was observed that the updating directions depend on the cosine of the metric of Dikin's ellipsoids of the normals to the cut, where the acute angle in the cosine formula favors convergence. Balas (1971) presented another type of cutting planes that employs intersection cuts to solve problems in integer programming. Letchford (2002) presented totally tight Chvátal-Gomory cuts for mixed integer programming. Using a convergent Lagrangian, a contour cut method was presented by Li, Sun and Wang (2006) to solve nonlinear integer programming problems. Luoy (1997) provided the analysis of a cutting plane method that uses weighted analytic center and multiple cuts. Manfred Padberg (Grötschel, 2004) made fundamental contributions to theoretical as well as computational aspects of integer programming and combinatorial optimization.

One possible approach for deriving cutting planes is to change the right hand sides of linear constraints. Our approach is conceptually different from this and results in a superlinear rate of convergence vis-a-vis the sublinear convergence rate of the Kelley's cutting plane algorithm. We demonstrate the performance of these new cuts using several examples.

2.2 Problem statement

We consider a nonlinear program (NLP)

$$\begin{aligned} Min \qquad & \phi_0(\mathbf{x}) \\ s.t. \qquad & \phi_j(\mathbf{x}) \geq \mathbf{0}, \quad \mathrm{j} = 1, 2, \cdots, \mathrm{p} \end{aligned} \tag{2.1}$$

and the values which the decision variables can take are given by a box constraint set $\mathbf{B}$ as

$$\mathbf{B} = \{\forall i | x_i \in [a_i, b_i]\}, \qquad \mathbf{x} = \{x_1, x_2, \cdots, x_n\}^t \subset \Re^n \tag{2.2}$$

The commonly employed methods to solve a nonlinear program (NLP) are cutting plane algorithms. These algorithms are good candidates to write a code for commercial solvers. The reader may kindly refer to the Notation section to know what each symbol and term means.

We first apply Kelley's cutting plane algorithm to three examples of optimization program having strongly nonlinear constraints.

Example 1.

$$\begin{aligned} \textit{Min} \quad & \phi_0(\mathbf{x}) \equiv -3.0x_1 - 12.5x_2 - 20.0x_3 \\ \textit{s.t.} \quad & g_1(\mathbf{x}) \equiv \sqrt{x_1} - 8x_2^2 - 2x_3 \geq 0 \\ & g_2(\mathbf{x}) \equiv -1.2x_1^2 + 6.0x_3^2 - 3.0x_2x_3 \geq 0 \end{aligned}$$

Here, $\phi_0(\mathbf{x}) \in \Re^{\mathbf{1}}$ and $\mathbf{g}(\mathbf{x}) \equiv \{g_1(\mathbf{x}), g_2(\mathbf{x})\}^t \in \Re^2$. The box constraints within which the optimum solution must lie are

$$0.0 \leq x_1 \leq 2.5 \quad 0.0 \leq x_2 \leq 2.0 \quad 0.0 \leq x_3 \leq 14.0$$

Example 2.

$$\begin{aligned} \textit{Min} \quad & \phi_0(\mathbf{x}) \equiv -3.0x_1 + 2x_2 + 0.8x_3 \\ \textit{s.t.} \quad & g_1(\mathbf{x}) \equiv 1.0 - (x_1 - 2)^2 - (x_2 - 5)^2 + x_3^2 \geq 0 \\ & g_2(\mathbf{x}) \equiv x_1^2 + (x_2 - 2) - 5.0x_3^2 \geq 0 \end{aligned}$$

The box constraints within which the optimum solution must lie are

$$0.0 \leq x_1 \leq 4.5 \quad 0.0 \leq x_2 \leq 6.0 \quad 0.0 \leq x_3 \leq 8.0$$

Example 3 (Ravindran, Ragsdell and Reklaitis, 1983).

$$\textit{Min} \quad \phi_0(\mathbf{x}) \equiv -x_1 - x_2$$

$$s.t. \quad g_1(\mathbf{x}) \equiv 2x_1 - x_2^2 - 1.0 \geq 0$$

$$g_2(\mathbf{x}) \equiv 9.0 - 0.8x_1^2 - 2.0x_2 \geq 0$$

The box constraints within which the optimum solution must lie are

$$0.0 \leq x_1 \leq 5.0 \qquad 0.0 \leq x_2 \leq 4.0$$

Observations.

The numerical experience reveals some observations which lead to a problem statement and motivation (see Table 4). These observations (and subsequent motivation) are:

1. The convergence of algorithm based on function evaluation of most violated constraint showed a rate of convergence which is sub-linear.

2. The algorithm seems not to reach a fixed or stationary point of nonlinear program as would be reached by a global search method and this local optimum solution should have been very close to the true local optimum solution.

Problem motivation.

1. The sublinear rate of Kelley's cutting plane method clearly implies that an improvement is required to come nearer to a super linear rate of convergence. (The terms to measure rate of convergence of algorithm which are sublinear,

superlinear, super superlinear are explained later in Section 5.)

2. Especially in the last few iterations behavior of solution sequence suggesting occurrence akin to the cycling behavior of linear programming (LP) solvers and needs to be analyzed further to eliminate it.

Thus improving rate of convergence to be almost superlinear and elimination of SLP behavior which is akin to cycling behavior of LP solver form a motivation to take up this problem for further study.

2.3 Cutting plane algorithm

In this work, we consider a nonlinear program (NLP)

$$\begin{aligned} & Min \quad \phi_0(\mathbf{x}) \\ s.t. \quad & \phi_j(\mathbf{x}) \geq \mathbf{0}, \quad \mathrm{j} = 1, 2, \cdots, \mathrm{p} \end{aligned} \tag{2.3}$$

where $\mathbf{x} = \{x_1, x_2, \cdots, x_n\}^t$ is the decision vector of optimization program. Here, we assume that the function $\phi_0 : \Re^n \rightarrow \Re$ is a continuous mapping and inequality constraints $\phi_j : \Re^n \rightarrow \Re; \quad j = 1, 2, \cdots, p$ are also continuous mappings. Before proceeding further, we give a brief description of successive

linear programming (SLP).

The basic algorithm for SLP, which is a global search method, is described below. Later, we will discuss other aspects and convergence theorem for this algorithm.

1. *Initial iteration.* The linear programming (LP) approximation begins with a linearized objective function $\hat{\phi}_0(\mathbf{x}) = \mathbf{c}^t\mathbf{x}$ and box constraints. Here, $\mathbf{c} = \{c_1, c_2, \cdots, c_n\}$ is a row vector of constants. In the initial iteration,

$$\begin{aligned} &Min \quad \hat{\phi}_0(\mathbf{x}) \\ s.t. \quad &\mathbf{Z}^o \equiv \{\forall i | a_i \leq x_i \leq b_i, i = 1, 2, \cdots, n\} \end{aligned} \tag{2.4}$$

   The call to LP solver gives us an optimum solution point denoted as $\mathbf{x} = \bar{\alpha}$ and for l^{th} iteration $\bar{\alpha}_l = \{\alpha_{1,l}, \alpha_{2,l}, \cdots, \alpha_{n,l}\}^t$, where subscript i denotes i^{th} decision variable and subscript l stands for l^{th} iteration. The LP solution is obtained by solving linear program in Eq. (2.4); this produces an optimum solution $\bar{\alpha}_1$, as this is used in iteration '1' to identify the most violated constraint as defined later.

2. *Cutting plane computation.*

Definition 1: Most violated constraint. According to Kelley's algorithm (1960), the most violated constraint is identified as

$$-g^o_{r,l} = \{\forall j \in \mathbf{J}; \exists r \in \mathbf{J}; \mathbf{J} = \{1, 2, \cdots, p\} | Max_{r\in\mathbf{J}}[-\phi_j(\bar{\alpha}_{l-1}), 0]\} \quad (2.5)$$

with r being the integer index, as returned by the *Max* function in Eq. (2.5). Here $\phi_j(\bar{\alpha}_l)$ is a numerical value of function $\phi_j(\mathbf{x})$ evaluated at $\mathbf{x} = \bar{\alpha}_l$, which is obtained as an optimum solution point by solving LP approximation for the l^{th} iteration of algorithm. $\diamondsuit$

The r^{th} constraint in Eq. (2.5) is identified as that corresponding to the maximum value of function $\phi_j(\bar{\alpha}_{l-1})$, $j \in \mathbf{J}$, $\mathbf{J} = \{1, \cdots, p\}$, and $r \in \mathbf{J}$. Using the most violated constraint, the function value $g^o_{r,l} = \phi_r(\bar{\alpha}_{l-1})$ and optimum solution point $\bar{\alpha}_{l-1}$ a cutting plane is obtained.

$$\mathbf{u}_l(\mathbf{x}; \bar{\alpha}_{l-1}) \equiv g^o_{r,l} + \sum_{i=1}^{n} \nabla_{x_i}\phi_r(\bar{\alpha}_{l-1})(x_i - \alpha_{i,l-1}) \geq 0 \qquad (2.6)$$

and is denoted as $\mathbf{u}_l$ and then added to the constraint set of LP program being solved every l^{th} iteration.

3. l^{th} *iteration.* Repeating the procedure above, for l^{th} iteration, we obtain an updated LP statement as

$$Min \quad \hat{\phi}_o(\mathbf{x}) = \mathbf{c}^t\mathbf{x}$$

$$s.t. \quad \{\forall l \in \{1, \cdots, k, \cdots, \infty\} | \mathbf{u}_l \geq 0\} \bigcap \mathbf{Z}^o$$

$$\mathbf{Z}^o \equiv \{\forall i | a_i \leq x_i \leq b_i, i = 1, 2, \cdots, n\} \tag{2.7}$$

where index l stands for current l^{th} iteration. From here on, using the following notation for indices of optimum solution point sets

$$\mathbf{P}_l = \{0, 1, 2, \cdots\} \tag{2.8}$$

$$\mathbf{P}_\infty = \{1, 2, \cdots, \infty\} \tag{2.9}$$

4. *Termination criteria.* The iterative procedure terminates when difference between successive values of optimum solution becomes negligible, i.e. when

$$\lim_{l \to \infty} |g^o_{r,l-1} - g^o_{r,l}| \to 0; \quad \text{or} \quad g^o_{r,l} > -\epsilon; \qquad \epsilon > 0 \tag{2.10}$$

ϵ being arbitrary. Here, $l \to \infty$ implies a large number of iterations. The algorithm terminates, when we are very close to the constraint hypersurface. Ideally, $g^o_{r,l} = 0$ as $l \to \infty$, unless in Eq. (2.10) we see that a limit point $g^o_{r,\infty} \neq 0$ is reached and algorithm cannot proceed further.

2.4 Stationary point of program

In this section, we give mathematical analysis leading to definition of a stationary point of a nonlinear optimization program.

Definition 2. We denote a sequence (Q_l) (Rudin, 1976) of numerical values of optimum solutions as

$$Q_l \equiv \{\forall l \in \mathbf{P}_l | \bar{\alpha}_l\} \tag{2.11}$$

The above is a set of solution vectors $\bar{\alpha}_l, \quad l = 0, 1, 2, \cdots$, put together as a set of solution points after execution of call to LP solver in l^{th} iteration of SLP algorithm. There is a sequence Q_l such that $\{\forall l \in \mathbf{P}_\infty | \bar{\alpha}_l\} \subset \Re^n$, which converges upon a solution point $\mathbf{x} = \bar{\alpha}^*$ as $l \to \infty$.

The linear approximation of NLP in Eq. (2.3) provided in Eq. (2.7) has a feasible region of local solutions $\mathbf{L}\Omega$, which is defined as

$$\mathbf{L}\Omega \equiv \{\forall l \in \mathbf{P}_l | \mathbf{u}_l \geq 0\} \tag{2.12}$$

with Kelley's cutting plane $\mathbf{u}_l$ being defined in Eq. (2.6). The feasible region for nonlinear program in Eq. (2.3) is given by nonempty Ω with

$$\Omega = \{\mathbf{x} | \phi_r(\mathbf{x}) \geq 0\}. \tag{2.13}$$

It follows that (Rudin, 1976)

$$\Omega \subset \mathbf{L}\Omega. \tag{2.14}$$

In the l^{th} iteration, an arbitrary point $\hat{\mu}_l$ lying in the domain $\mathbf{L}\Omega - \Omega$ gives us a cutting plane after obtaining optimum solution $\bar{\alpha}_l$ by calling LP solver. The update of point set $\{l \in \mathbf{P}_l | \mathbf{L}\Omega\}$ obeys the relation in Eq. (2.14). It follows that we have to iterate until a fixed or stationary point is reached and this yields a sequence Q_l [Eq. (2.11)]. To demonstrate the convergence of the algorithm, we need to show that

$$\lim_{l \to \infty} Q_l = \bar{\alpha}^*. \tag{2.15}$$

$\diamondsuit$

The theorem in Appendix I establishes that for an optimization program to which the sequence of optimum solutions converge is a unique stationary point.

Remark. We have proved that as $l \to \infty$, $diam \;\; Q_l \to 0$. This implies that the intersection is of sub-collections κ_l as sequence Q_l is appended with an optimum solution point $\bar{\alpha}_l$ reduces to a single numerical value $\bar{\alpha}^*$, which is the stationary point of nonlinear program.

Our aim in this analysis is to devise a sharp cut and provide a new convergence theorem. We wish to prove that the rate of convergence of this algorithm is significantly higher than that of Kelley's algorithm.

Remark. At this juncture, we will make a sequential modification in algorithm one step at a time as seen later. The idea is to modify the algorithm based on numerical observations and what we learn from them to seek further change in algorithm to improve the rate of convergence. The process of modification may continue till satisfactory rate of convergence of algorithm is obtained.

2.5 Rate of convergence

The sequence of optimum solutions converges to a limit point and yields the stationary point of program upon termination. The measure of the rate of convergence as given in termination critreria of SLP [Eq. (2.10)] is the value of $g^0_{r,l}$, which was found to be only sublinear for Kelley's algorithm. There is a need to improve the rate of convergence. The terms sub-linear, linear, super-linear, and super-super-linear are applied to the rate of convergence. The rate of convergence is defined as

$$\chi = |\vartheta(g^o_{r,l-1}) - \vartheta(g^o_{r,l})|$$

The order of magnitude is denoted as $\vartheta(\cdot)$ and is an integer exponent of 10.

The rate of convergence is

1. sublinear if $\chi < 1$,
2. linear if $\chi = 1$,
3. superlinear if $\chi = 2$,
4. super superlinear if $\chi > 2$ (nearing 4 or more).

Convergence of algorithm.

The SLP algorithm generates following sequences during iterative process

1. an optimum solution sequence

$$Q_l \equiv \{\bar{\alpha}_0, \bar{\alpha}_1, \cdots, \bar{\alpha}_\infty\}$$

2. a sequence of numerical values of linearized objective function $\hat{\phi}_0(\mathbf{x})$ as

$$\Gamma_l \equiv \{\hat{\phi}_0(\bar{\alpha}_0), \hat{\phi}_0(\bar{\alpha}_1), \cdots, \hat{\phi}_0(\bar{\alpha}_\infty)\}$$

3. a sequence of numerical values of constraint function $\phi_r(\mathbf{x})$

$$\Psi_l \equiv \{g^o_{r,0}|_{\mathbf{x}=\bar{\alpha}_0}, g^o_{r,1}|_{\mathbf{x}=\bar{\alpha}_1}, \cdots, g^o_{r,\infty}|_{\mathbf{x}=\bar{\alpha}_\infty}\}$$

If the converging algorithm reaches to a stationary point of program during the iterative process requires that

$$\lim_{l \to \infty} Q_l = \bar{\alpha}^*; \qquad \lim_{l \to \infty} \hat{\phi}_0(\mathbf{x} = \bar{\alpha}^*) = \phi_0^*; \qquad \lim_{l \to \infty} g^o_{r,l}|_{\mathbf{x}=\bar{\alpha}_l} = 0$$

We can verify the following facts by observation.

1. At the termination point of the algorithm, we should have

$$g^o_{r,\infty}|_{\mathbf{x}=\bar{\alpha}^*} = 0; \qquad \mathbf{L}\Omega - \Omega = \emptyset$$

2. The convexity assumption is expected to hold in the search range given by box constraints. The function $\phi_r(\mathbf{x})$ corresponding to the most violated constraint is convex. The local optimum solution obtained by global search method at the termination point is then a unique global optimum solution.

3. We define distance measure d as

$$d(\alpha_{i,l}, \alpha_{i,l+1}) = |\alpha_{i,l} - \alpha_{i,l+1}|; \quad l \in \mathbf{P}_k$$

and employing the distance measure to all the three sequences above, the convergence of algorithm can be expressed using condition for a Cauchy sequence.

The sequence Q_l is a Cauchy sequence, iff,

$$\lim_{l \to \infty} diam \;\; Q_l = 0.$$

The sequence Γ_l is a Cauchy sequence, iff,

$$\lim_{l \to \infty} diam \;\; \Gamma_l = 0.$$

The sequence Ψ_l is a Cauchy sequence, iff,

$$\lim_{l \to \infty} diam \;\; \Psi_l = 0.$$

$\diamond$

2.6 Construction of a new cut

We need to generate a new sequence $\tilde{Q}_l$, which will converge faster than that generated using Kelley's cut, Q_l. In order to achieve this, the sequence of vector points computed must cut steeper to descend upon hypersurface of identified constraint. In order to keep computational complexity to a minimum, we evaluate constraint function $[\phi_r(\mathbf{x})]$ and objective function $[\hat{\phi}_0(\mathbf{x})]$ (given a set of vector points as numerical data) and attempt to move steeply toward constraint hypersurface.

Definition 3. A new sequence of vector points.

We begin with a vector point $\bar{\alpha}_{l-1}$ which gives us $g^o_{r,l} = \phi_r(\bar{\alpha}_{l-1})$ for r^{th} identified constraint obtained from $(l-1)^{th}$ iteration. To descend steeply, we generate a vector data point set forming a set Φ for given search segment, as

$$\begin{aligned} \Phi &\equiv \{\forall \wp_{\pm} \in \{0, 1, 2, \cdots, \aleph_{\pm-1}\} \ \ |\bar{\nu}_{\wp_{\pm}}\} \\ \bar{\nu}_{\wp_{\pm}} &= \{\forall i = 1, \cdots, n, \wp_{\pm} = 1, \cdots, \aleph_{\pm} | \alpha_{i,l} \pm \wp_{\pm}\delta_{\pm,i}\} \end{aligned} \tag{2.16}$$

where

$$\forall i | \delta_{i,-} = (\alpha_{i,l} - a_i)/(\aleph_- - 1); i = 1, \cdots, n | a_i \leq \nu_{i,\wp_-} \leq b_i$$

$$\forall i | \delta_{i,+} = (b_i - \alpha_{i,l})/(\aleph_+ - 1); i = 1, \cdots, n | a_i \leq \nu_{i,\wp_+} \leq b_i$$

The vector point data set $E_{0,l} \subset \Phi$ is generated as

$$\begin{aligned} E_{0,l} \ \equiv \ & \{\forall \wp_{\pm} \in \{0, 1, 2, \cdots, \aleph_{\pm-1}\} | \bar{\nu}_{\wp_{\pm}}; \\ & 0 < \varsigma_{r,\wp_{\pm}}(\bar{\nu}_{\wp_{\pm}}) < g^o_{r,l-1}\} \\ & \varsigma_{r,\wp_{\pm}}(\bar{\nu}_{\wp_{\pm}}) = \phi_r(\bar{\nu}_{\wp_{\pm}}) < g^o_{r,l-1} \end{aligned} \tag{2.17}$$

Now, the new vector point is generated as

$$\psi^o_{r,l} = \{r | \sigma \in \{\wp_{\pm}\}; \ \ \bar{\nu}_{\wp_{\pm}} \in E_0 | \varsigma_{r,\sigma}(\bar{\nu}_{\sigma}) | Min[\hat{\phi}_0(\bar{\nu}_{\wp_{\pm}}); \wp_{\pm} = 0, ..\aleph_{\pm-1}]\} \tag{2.18}$$

The new vector point is denoted as $\bar{\beta}_l = \bar{\nu}_{\sigma}$. This new vector point is used to generate a new cut.

Definition 4: A new cut.

The vector point, $\bar{\beta}_l = \bar{\nu}_\sigma$ is used to form a new cut (or cutting plane steeply descending upon hyper surface of the identified constraint and closer to hypersurface than Kelley's cutting plane) as

$$\mathbf{v}_l(\mathbf{x}; \bar{\beta}_l) \equiv \psi^o_{r,l} + \sum_{i=1}^{n} \nabla_{x_i} \phi_r(\bar{\beta}_l)(x_i - \bar{\beta}_{i,l}) \geq 0 \tag{2.19}$$

Next, we introduce the first change in new algorithm, an initial "jump", which is expected to considerably reduce the required number of iterations.

2.7 Initial "jump".

Let the optimum solution obtained by calling LP solver be $\bar{\alpha}_0$. We introduce a "jump" from $g^o_{r,0} = \phi_r(\bar{\alpha}_0)$ to a new value $g_0 = w_0 \times g^o_{r,0}$, such that $g_0 << g^o_{r,0}$ (w_0 is an arbitrary positive number and a small fraction).

Definition 5. Initial "jump".

Let $\omega_0(\bar{\alpha}_0)$ be a convex polyhedral set given as

$$\omega_0 \equiv \{\forall i | a_i \leq x_i \leq b_i; \quad i = 1, 2, \cdots, n\}$$

which is the initial hypercube formed by box constraints. Let there be a set of vector points [using Eq. (2.17)] given as

$$g^o_{r,1} = \{\exists \hbar \in \mathbf{E}_{0,1} | \phi_r(\bar{\nu}_\ell) \in [0, g_0) | Min_{\hbar \in \mathbf{E}_{0,1}}(\hat{\phi}_0(\bar{\nu}_\ell))\} \tag{2.20}$$

Then the new convex polyhedral set ω_1 is generated as

$$\omega_1 \equiv \{\mathbf{u}_1(\mathbf{x};\bar{\alpha}_0) \geq 0 \bigcap \mathbf{v}_1(\mathbf{x};\bar{\beta}_1) \geq 0\} \qquad \text{and} \qquad \bar{\beta}_1 = \bar{\nu}_\hbar; \quad \omega_1 \subset \omega_0$$

It is expected that the "jump" will reduce the number of initial iterations while approaching the hyper surface of convex function $\phi_r(\mathbf{x})$.

Remark. The numerical experience shows that the initial "jump" is effected using Eq. (2.20) reduces initial iterations considerably, to see how the algorithm performs during further iterations with addition of new cut forming intersection set with Kelley's cutting plane. Next, we define the sharp cut.

2.8 First sharp cut

After defining the new cut $\mathbf{v}_l$ given in Eq. (2.19), our objective is to find another convex polyhedral set ω_l for l^{th} iteration to form a new intersection set, which satisfies a relation $\omega_l \subset \omega_{l-1}$. The intersection of the two cuts or cutting planes $\mathbf{u}_l$ and $\mathbf{v}_l$ forms a convex set $\hat{\omega}_l$.

The small reduction in convex region formed by $\mathbf{Z}^l$ during subsequent iteration can be increased by use of weights to alter the cutting plane equation. The weighted cut is given as

$$\mathbf{v}'_l \equiv w_{1,l} \times \psi^o_{r,l} + \sum_{i=1}^{n} \nabla_{x_i}\phi_r(\mathbf{x})|_{\mathbf{x}=\bar{\nu}'_\sigma}(x_i - \nu'_{i,\sigma}) \geq 0 \tag{2.21}$$

where,

$$\bar{\nu}'_{\sigma} = w'_{1,l} \times \bar{\nu}_{\sigma} \tag{2.22}$$

Here, $w_{1,l}$ is a weight, generated as a random number. For toy examples considered here, the weight $w_{1,l}$ usually lies the interval $[0.94, 0.98]$ and index '1' in subscript means weight used for first sharp cut. Similarly, the weight vector

$$w'_{1,l} = \{w'_{1,1,l}, w'_{1,2,l}, \cdots, w'_{1,n,l}\} \subset \Re^n$$

scales down the optimum solution point $\bar{\nu}_{\sigma}$. We do this with an expectation that a small added cut-off of remaining hypercube $\mathbf{Z}^l$ will accelerate the rate of convergence. The new cutting plane is termed as first sharp cut.

However, we learn from numerical experience and comparing results with those for Kelley's cut that successive decrement in vector point and optimum solution sequence Q_l shows that with every iteration the numerical values indicate a fall in the rate of convergence. This rate of convergence tends to be sublinear. To improve this rate of convergence (to nearly superlinear), we construct another cutting hyperplane in vicinity of the selected value of objective function $\hat{\phi}_0(\bar{\nu}_{\sigma})$, which is the new vector point. The final algorithm without weights did not perform well. This implies that randomized weights are necessary to move down steeply.

2.9 Second sharp cut

We find from numerical experience that the intersection set of Kelley's cutting plane and first sharp cut, expressed as $\{l \in \mathbf{P}_\infty | \{\mathbf{u}_l \geq 0 \cap \mathbf{v}_l \geq 0\}\}$ does not have the expected effect. And thus we need an extra cut within the remaining hypercube $\mathbf{Z}^l$.

In order to give an extra cut, we observe an important fact. When we extract a subset of vector points used in selecting a new vector point $\bar{\beta}_l$, we obtain a level set which is controlled by a single parameter $\hat{\delta}$. This set $\mathbf{E}_{1,l}$ is given as

$$\mathbf{E}_{1,l} \equiv \{\forall \ell \in \mathbf{E}_{0,l} | \forall \bar{\nu}_\ell | \phi_r(\bar{\nu}_\ell) \geq \hat{\delta} \times Max\left[\phi_r(\bar{\nu}_\ell), 0\right]\}. \tag{2.23}$$

When we obtain a multivariate linear fit to form a new cutting plane, the hyperplane formed is almost parallel to hypersurface of the most violated constraint. When this forms an intersection set with first cut the intersection point moves a substantial distance from previous Kelley's cutting plane.

$$\mathbf{w}'_l \equiv \hat{\delta} w_{1,l} \times \psi^o_{r,l} + \sum_{i=1}^{n} \hat{\theta}_i (x_i - \nu'_{i,\sigma}) \geq 0 \tag{2.24}$$

Here, $\psi^o_{r,l}$ is defined in Eq. (2.18). These random numbers are generated using a symbolic manipulator (MATHEMATICA software of Wolfram Research)

for second sharp cut. Upon application of this second cut, which accelerates convergence, we observe the following.

1. While selecting a new vector point $\bar{\beta}_l$ at some iteration of SLP, this vector point remains above the current local optimum solution of LP. This causes a reversal in sign of gradient set for most violated constraint used to form the cutting hyperplane.

2. The numerical values of gradient set of new cutting plane slowly begin to reduce to values less than unity.

3. The rate of convergence is initially linear and then becomes superlinear, until a stationary point of program is reached. The new algorithm is at least twice as fast as Kelley's algorithm.

Note that when the second sharp cut is effected with the parameter $\hat{\delta}$ controlling the "steepness" of descent, this cut intersects with the first sharp cut and is still intersecting the previous hypercube, but it does not cross boundary of $\mathbf{Z}^l$. The extra cut moves away from previous hypercube measured in terms of how far away the intersection point of the first and second sharp cuts has moved from the previous hypercube $\mathbf{Z}^{l-1}$.

2.10 The sharp cut algorithm

Here, we define the sharp cut or new intersection set in mathematical terms and show how cycling problem during successive calls to LP solver for iterations of SLP algorithm can be eliminated.

Definition 6. A sharp cut.

We define a sharp cut as an intersection set of the three cutting planes, one in Eq. (2.6), which is a Kelley's cutting plane (Kelley, 1960), a first sharp cut in Eq. (2.21), and a second sharp cut in Eq. (2.24). The set of points forming a sharp cut, lie within the set $\tilde{\omega}$, defined as

$$\tilde{\omega}(\bar{\alpha}_l, \bar{\beta}_l) \equiv \{\forall l \in \mathbf{P}_\infty | \mathbf{u}_l \geq 0 \bigcap \mathbf{v}_l \geq 0 \bigcap \mathbf{w}_l \geq 0\}. \tag{2.25}$$

The set $\tilde{\omega}$ is an intersection of three sets of cutting (hyper-)planes and in turn is convex (Mangaserian, 1969; Boyd and Vandenberghe, 2004). Any two points in this convex polyhedra obeys a relation

$$\{\forall l \in \mathbf{P}_l | \forall \mathbf{x}, \mathbf{y} \in \tilde{\omega}(\bar{\alpha}_l, \bar{\beta}_l) | \lambda \mathbf{x} + (1-\lambda)\mathbf{y} \in \tilde{\omega} | \lambda \in [0,1]\} \tag{2.26}$$

The convergence theorem for sharp cut algorithm is given in Appendix II. The basic SLP algorithm forms an intersection set of Kelley's cutting plane, and two sharp cuts. The convergence of Cauchy sequences to a limit point is explained with the Theorem given in Appendix III.

2.11 Cycling behavior of LP solver

When the cutting hyperplanes reach near the constraint ($\phi_r(\mathbf{x})$) hypersurface the gradient terms in Eq. (2.21) tend to become zero and the same holds true for $g^o_{r,l}$, $\psi^o_{r,l}$. We find from numerical experience that near termination of algorithm the cutting hyperplanes generated by algorithm in current and previous iterations are close and almost parallel to each other. The right hand sides of these cutting planes (Kelley's cut and first sharp cut) which are linear inequalities, can be re-arranged as

$$\mathbf{u}_l \equiv \sum_{i=1}^{n} \nabla_{x_i}\phi_r(\bar{\alpha}_{l-1})x_i \geq \hat{\xi}_1 \qquad \mathbf{v}'_l \equiv \sum_{i=1}^{n} \nabla_{x_i}\phi_r(\bar{\beta}_l)x_i \geq \hat{\xi}_2 \tag{2.27}$$

with the expressions for the two constants $\hat{\xi}_1$, $\hat{\xi}_2$ in Eq. (2.27), (see Eqs. (2.18), (2.8), where $w_{1,1,l}$ and $\psi^o_{r,l}$ are defined respectively), after some re-arrangement reduce to

$$\hat{\xi}_1 = g^o_{r,l} + \sum_{j=1}^{p} \nabla_{x_i}\phi_r(\bar{\alpha}_{l-1}) \times \alpha_{i,l-1}; \qquad \hat{\xi}_2 = w_{1,1,l}\psi^o_{r,l} + \sum_{j=1}^{p} \nabla_{x_i}\phi_r(\bar{\beta}_l) \times \beta_{i,l}$$

These constants $\hat{\xi}_1$, $\hat{\xi}_2$ become negligible that the linear system representing a polyhedra becomes inconsistent. At this point, it is difficult for a computing machine to maintain precision and since right hand side is nearly the same for at least two inequalities, during a call to LP solver, we observe cycling of LP (Gass, 1979).

When randomized weights are used,

1. The right hand sides in Eq. (2.27) become non-zero and unequal and,

2. In the neighborhood of the unique global optimum solution attainable by global search method, the Kelley's cut and first sharp cut become parallel to those in previous iteration. However, the second sharp cut keeps changing till it touches the constraint hypersurface within the limits of machine precision set by the user. This new added cut implies that it has made previous intersection set inactive at the corner point and a new optimum solution is formed.

Thus randomized weights and inclusion of third cut i.e. second sharp cut, prevent the cycling of LP. The converging sequence shows that a limit point has reached. This is the the required stationary point of program, and remains unaltered in further iterations. This is a significant improvement over Kelley's algorithm, which requires far more iterations before coming closer to local optimum solution and at times finds a degenerate point.

We need to elaborate here more on cycling as a theoretical concept and as a numerical experience in computing solutions while executing SLP as a global search method which uses a sharp cut. Beale (1955) has explained the

cycling in simplex algorithm using geometric meaning of re-appearance of basis and this issue will be analyzed further in detail in another publication elsewhere. This becomes a pertinent issue as MATHEMATICA (of Wolfram Research) flags an error saying that – "The specified constraints cannot be satisfied with tolerance 1.0×10^{-6}", (which is explained in Beale's article). We believe that any modification to overcome this difficulty will enhance the efficiency of the sharp cut algorithm and help us in obtaining the stationary point.

As Gass (1979) has explained, we can encounter two types of cycling, one is classical cycling, and the other is computer cycling. The classical cycling arises during call to LP solver in the following manner. If data in decimal notation, and can be expressed as rational fractions, then as an idealistic view computations can be performed without loss of accuracy and round-off error. Here the term loss of accuracy points to the number of decimal points for the solution can be correct. The term rational fraction denotes the machine precision limits during computations. If a version of LP as given by Dantzig or Gass is used, it calls for pricing out all vectors not in the current basis. Thus for a minimization problem, the ties are broken using a rule (arbitrary and consistent) that selects the vector with largest index. The vector to leave

is selected based on minimum-ratio test, and ties here are broken arbitrarily using largest pivot element (Gass, 1979). It is a well-known fact that if right hand side of one or few linear inequalities were to be zero, or there are ties for minimum ratio, then they can cause a zero to appear on right hand side. This leads to cycling that is after a number of vector changes the former basis reappears. As we continue calling LP solver, the basis keeps re-appearing, which means that value of objective function won't change. The finiteness of simplex method and its convergence do not apply as these conditions require non-zero positive right hand side coefficients for all feasible bases (which is referred to as a non-degeneracy). Gass (1979) refers to this as a case of classical cycling.

Cycling during computations is often denoted as computer cycling and refers to numerical solution of LP using a computer code run by a digital computer. These systems do not use rational fraction transformations, and use binary arithmetic (may be single or double precision), standardized rounding procedures or truncation. The developers in the past found that certain techniques can make LP converge faster and can prevent cycling. They used different criteria to choose incoming and outgoing variables, special inversion updating procedures, scaling of problem as required, small coefficients

being put to zero for numerical stability, tolerance being applied to selection of pivot element, and degenerate problems being automatically perturbed. When these precautions are taken, a mathematical programming problem never produces the conditions associated with classical cycling. Thus one may hope that computer cycling will not occur. Gass (1979) has concluded that classical cycling is limited to artificially constructed problems.

2.12 Examples

To illustrate the findings for the global search method which employs a sharp cut, we will solve examples in Section 2.2.

Example 1.

We find that Kelley's algorithm requires 26 iterations before settling upon an optimum solution $\bar{\alpha}^* = \{x_1^* = 1.23606, x_2^* = 0.0101739, x_3^* = 0.555478\}^t$. This solution point should be a stationary point. The generated sequence of solutions however oscillates for a large number of iterations. On the contrary, the sharp cut algorithm converges to a solution point $\bar{\alpha}^* \equiv \{x_1^* = 1.14024, x_2^* = 0.0363976, x_3^* = 0.519111\}^t$ after five iterations and the next

two iterations generate the same sequence of solutions implying that algorithm reached a limit point which is a fixed or stationary point of the nonlinear program after six iterations. Thus, the new sharp cut algorithm is at least twice as fast as Kelley's algorithm. The minimum function value reached for Kelley's algorithm is 14.9499, while that for sharp cut algorithm is 14.2579. The two values are very close. Since, this stationary solution is only a guess value to start local search method such as the trust region method, we find that computational complexity is reduced substantially. The two constraints become extremely close to each other at the end of algorithm, i.e. $\phi_1(\bar{\alpha}_l), \phi_2(\bar{\alpha}_l) \to 0$. If the constraints were approximated to be linear in a short range of decision variables, it would have generated the corner point solution. The solution obtained using global search method is therefore a good guess for a local search method in obtaining the true optimum solution. The index of most violated constraint keeps switching between 1 and 2 for Kelley's method and later settles on 1, while for sharp cut algorithm constraint 2 is the most violated constraint.

Following observations are made from the results reported in Tables 1-3.

1. The sequences of $\alpha_{1,l}$, $\alpha_{2,l}$, $\alpha_{3,l}$ are monotonic (Table 1).

2. In zeroth iteration, the generated vector point ($\bar{\beta}$) is below the one ob-

tained by calling LP solver. This iteration led to "initial jump". After second iteration, the generated point $\bar{\beta}_l$ lies above the local optimum solution point $\bar{\alpha}_{l-1}$ and the signs of the gradients are reversed.

3. After fourth iteration, as we come very close to the hypersurface of the most violated constraint, Kelley's cut and first sharp cut become almost parallel to those in previous iterations. From Table 3, we see that cutting planes $\mathbf{u_4}$-$\mathbf{u_7}$, $\mathbf{v_4}$-$\mathbf{v_7}$ remain parallel. The second sharp cut brings the LP solution further closer to the constraint hypersurface and a stationary point is reached in two iterations.

Example 2.

This is an interesting example. Kelley's cut yields an optimum solution after 25 iterations, $x_1 = 1.167951$, $x_2 = 4.445299$, $x_3 = 0$. Here the optimum value of one decision variable x_3 is zero with $g^0_{r,25} = O(10^{-12})$. This is a trivial solution and not acceptable. The intersection set of Kelley's cutting plane and two sharp cuts require 10 iterations and attains an optimum solution value, $x_1 = 0.705285$, $x_2 = 4.9178$, $x_3 = 0.826465$ and which remains constant for next two iterations. Thereafter the algorithm is terminated after reaching a value $g^0_{r,12} = O(10^{-16})$. While Kelley's method could not find a correct non-trivial optimum solution, the new intersection cut is able to find the

same.

A comparison of Kelley's cut and sharp cut algorithm is shown in Table 4. Since other algorithms in current literature are characteristically different, we have not compared these with the new sharp cut algorithm.

Since we have relaxed the assumption of convexity for most violated constraint, we are trying to solve a non-convex program and test the performance of this new algorithm. In Appendix IV, we have reported numerical experience of solving four more examples of non-convex program using this algorithm and have commented on the outcome of numerical experiments.

2.13 Optimization of cascade of CSTRs

In this section, we consider a problem of optimization of three continuously stirred tank reactors (CSTRs) in series. A first order consecutive exothermic reaction A $\rightarrow$ B $\rightarrow$ C occur and each CSTR is maintained at a favorable temperature by employing a cooling jacket. The reaction rate constant for step A $\rightarrow$ B is $k_{0,1}$ and for the consecutive reaction step B $\rightarrow$ C the rate constant is $k_{0,2}$. The dynamics of the three CSTRs is described by

Dynamics of CSTR I.

$$\begin{aligned}
V_1\frac{dC_{A,1}}{dt} &= F(\hat{R}C_{A,0}+(1-\hat{R})C_{A,2}-C_{A,1})-V_1k_{0,1}exp\left[\frac{-E_1}{RT_1}\right]C_{A,1}\\
V_1\frac{dC_{B,1}}{dt} &= F(\hat{R}C_{B,0}+(1-\hat{R})C_{B,2}-C_{B,1})+V_1k_{0,1}exp\left[\frac{-E_1}{RT_1}\right]C_{A,1}\\
&- V_1k_{0,2}exp\left[\frac{-E_2}{RT_1}\right]C_{B,1}\\
\rho c_pV_1\frac{dT_1}{dt} &= \rho c_pF(\hat{R}T_0+(1-\hat{R})T_2-T_1)-H_1V_1k_{0,1}exp\left[\frac{-E_1}{RT_1}\right]C_{A,1}\\
&- H_2V_1k_{0,2}exp\left[\frac{-E_2}{RT_1}\right]C_{B,1}-UA_{H,1}(T_1-T_{J,1})
\end{aligned}$$

Dynamics of CSTR II.

$$\begin{aligned}
V_2\frac{dC_{A,2}}{dt} &= F(C_{A,1}-C_{A,2})-V_2k_{0,1}exp\left[\frac{-E_1}{RT_2}\right]C_{A,2}\\
V_2\frac{dC_{B,2}}{dt} &= F(C_{B,1}-C_{B,2})+V_2k_{0,1}exp\left[\frac{-E_1}{RT_2}\right]C_{A,2}\\
&- V_2k_{0,2}exp\left[\frac{-E_2}{RT_2}\right]C_{B,2}\\
\rho c_pV_2\frac{dT_2}{dt} &= \rho c_pF(T_1-T_2)-H_1V_2k_{0,1}exp\left[\frac{-E_1}{RT_2}\right]C_{A,2}\\
&- H_2V_2k_{0,2}exp\left[\frac{-E_2}{RT_2}\right]C_{B,2}-UA_{H,2}(T_2-T_{J,2})
\end{aligned}$$

Dynamics of CSTR III.

$$\begin{aligned}
V_3\frac{dC_{A,3}}{dt} &= \hat{R}F(C_{A,2}-C_{A,3})-V_3k_{0,1}exp\left[\frac{-E_1}{RT_3}\right]C_{A,3}\\
V_3\frac{dC_{B,3}}{dt} &= \hat{R}F(C_{B,2}-C_{B,3})+V_3k_{0,1}exp\left[\frac{-E_1}{RT_3}\right]C_{A,3}\\
&- V_3k_{0,2}exp\left[\frac{-E_2}{RT_3}\right]C_{B,3}
\end{aligned}$$

$$\begin{aligned}\rho c_p V_3 \frac{dT_3}{dt} &= \hat{R}\rho c_p F(T_2 - T_3) - H_1 V_3 k_{0,1} exp\left[\frac{-E_1}{RT_3}\right] C_{A,3} \\ &- H_2 V_3 k_{0,2} exp\left[\frac{-E_2}{RT_3}\right] C_{B,3} - UA_{H,3}(T_3 - T_{J,3})\end{aligned}$$

In terms of the dimensionless variables listed below the mass and energy balances are re-stated.

$$\begin{array}{lll} x_1 = \frac{C_{A,0}-C_{A,1}}{C_{A,0}} & x_4 = \frac{C_{A,0}-C_{A,2}}{C_{A,0}} & x_7 = \frac{C_{A,0}-C_{A,3}}{C_{A,0}} \\ x_2 = \frac{C_{A,0}-C_{B,1}}{C_{A,0}} & x_5 = \frac{C_{A,0}-C_{B,2}}{C_{A,0}} & x_8 = \frac{C_{A,0}-C_{B,3}}{C_{A,0}} \\ x_3 = \frac{T_1}{T_0} & x_6 = \frac{T_2}{T_0} & x_9 = \frac{T_3}{T_0} \\ \tau_1 = \frac{V_1}{F} & \tau_2 = \frac{V_2}{F} & \tau_3 = \frac{V_3}{F} \\ \alpha_1 = \frac{UA_{H,1}}{\rho c_p F T_0} & \alpha_2 = \frac{UA_{H,2}}{\rho c_p F T_0} & \alpha_3 = \frac{UA_{H,3}}{\rho c_p F T_0} \\ \mu_1 = \frac{H_1}{\rho c_p} & \mu_2 = \frac{H_2}{\rho c_p} & \\ \gamma_1 = \frac{E_1}{RT_0} & \gamma_2 = \frac{E_2}{RT_0} & \\ \hat{R}_1 = 0 & \hat{R} = \hat{R}_2 \neq 0 & \hat{R}_3 = 0; \\ \widehat{h} = \ln[k_{0,1}] & \widehat{g} = \ln[k_{0,2}] & \end{array}$$

The dimensionless groups give rise to following equality constraints.

$$x_1 - x_4 + \hat{R}x_4 - \tau_1 exp\left[\hat{h} - \frac{\gamma_1}{x_3}\right](1 - x_1) = 0 \quad (2.28)$$

$$\begin{aligned}(1 - \widehat{R})(1 - x_5) - (1 - x_2) + \tau_1 exp\left[\widehat{h} - \frac{\gamma_1}{x_3}\right](1 - x_1) \\ -\tau_1(1 - x_2) exp\left[\widehat{g} - \frac{\gamma_2}{x_3}\right] = 0 \quad (2.29)\end{aligned}$$

$$
\begin{aligned}
(\hat{R}-x_3)+(1-\hat{R})x_6-\mu_1\tau_1(1-x_1)exp\left[\hat{h}-\frac{\gamma_1}{x_3}\right] & \\
-\mu_2\tau_1(1-x_2)exp\left[\hat{g}-\frac{\gamma_2}{x_3}\right] &= 0 \quad (2.30)\\
-x_1+x_4-\tau_2(1-x_4)exp\left[\hat{h}-\frac{\gamma_1}{x_6}\right] &= 0 \quad (2.31)\\
-x_2+x_5+\tau_2(1-x_4)exp\left[\widehat{h}-\frac{\gamma_1}{x_6}\right] & \\
-\tau_2(1-x_5)exp\left[\widehat{g}-\frac{\gamma_2}{x_6}\right] &= 0 \quad (2.32)\\
x_3-x_6-\delta_2\alpha_2-\mu_1\tau_2(1-x_4)exp\left[\hat{h}-\frac{\gamma_1}{x_6}\right] & \\
-\mu_2\tau_2(1-x_5)exp\left[\hat{g}-\frac{\gamma_2}{x_6}\right] &= 0 \quad (2.33)\\
\hat{R}(-x_4+x_7)-\tau_3(1-x_7)exp\left[\hat{h}-\frac{\gamma_1}{x_9}\right] &= 0 \quad (2.34)\\
\widehat{R}(-x_5+x_8)+\tau_3(1-x_7)exp\left[\widehat{h}-\frac{\gamma_1}{x_9}\right] & \\
-\tau_3(1-x_8)exp\left[\widehat{g}-\frac{\gamma_2}{x_9}\right] &= 0 \quad (2.35)\\
\hat{R}(x_6-x_9)-\delta_3\alpha_3-\mu_1\tau_3(1-x_7)exp\left[\hat{h}-\frac{\gamma_1}{x_9}\right] & \\
-\mu_2\tau_3(1-x_8)exp\left[\hat{g}-\frac{\gamma_2}{x_9}\right] &= 0 \quad (2.36)\\
\frac{x_2}{1-x_1}-f_1 &\geq 0 \quad (2.37)\\
\frac{x_5}{1-x_4}-f_2 &\geq 0 \quad (2.38)\\
\frac{x_8}{1-x_7}-f_3 &\geq 0 \quad (2.39)
\end{aligned}
$$

The lower and upper limits are expressed as box constraints which define

the initial hypercube $\mathbf{Z}^o$ as

$$\Delta_i \geq x_i \geq \nu_i; \quad i = \{2, 3, 5, 6, 8, 9\} \tag{2.40}$$

The objective function $\hat{\phi}_0$ is expressed as

$$\hat{\phi}_0 \equiv \sum_{i=1}^{9} C_i x_i; \quad i = \{2, 3, 5, 6, 8, 9\} \tag{2.41}$$

where the cost coefficients C_2, C_5, C_8 are pumping costs and C_3, C_6, C_9 are cooling system operation costs. In this example, we set the relative value of former cost coeffiencts as unity and add 40% extra cost in relative terms for cooling operation (for this generic example, we have used arbitrary values for the purpose of demonstrating the working and advantage of the sharp cut algorithm).

The lower and upper limits in box constraints are,

$$\nu_1 = 0.4 \quad \Delta_1 = 0.99 \quad \nu_2 = 0.2 \quad \Delta_2 = 0.99 \quad \nu_3 = 0.8 \quad \Delta_3 = 1.2$$

$$\nu_4 = 0.4 \quad \Delta_4 = 0.99 \quad \nu_5 = 0.2 \quad \Delta_5 = 0.99 \quad \nu_6 = 0.8 \quad \Delta_6 = 1.2$$

$$\nu_7 = 0.3 \quad \Delta_7 = 0.99 \quad \nu_8 = 0.1 \quad \Delta_8 = 0.99 \quad \nu_9 = 0.8 \quad \Delta_9 = 1.2$$

The cost coefficients are given as

$$C_2 = 1.0 \quad C_5 = 1.0 \quad C_8 = 1.0$$

$$C_3 = 1.4 \quad C_6 = 1.4 \quad C_9 = 1.4$$

The physico-chemical properties and parameters are given here.

After the successful run of the sharp cut algorithm, we find that seven

Table 2.1: Table of properties and parameters

$R = 8.33$(kJ/kmol K)	$E_1 = 56000$(kJ/kmol)	$E_2 = 50000$(kJ/kmol)
$C_{A,0} = 2.0$(kmol / m^3)	$C_{B,0} = 0$(kmol / m^3)	$T_0 = 315$(K)
$\rho = 200.0$(kmol/m^3)	$c_p = 2.0$(kJ/kmol K)	$F = 0.2(m^3$/hr)
$(-\Delta H_1) = -77400$(kJ/kmol)	$(\Delta H_2) = -62000$(kJ/kmol)	$U = 6000$(kJ/m^2 hr K)
$A_{H,1} = 0.4(m^2)$	$A_{H,2} = 0.44(m^2)$	$A_{H,3} = 0.46(m^2)$
$V_1 = 0.5(m^3)$	$V_2 = 0.7(m^3)$	$V_3 = 0.6(m^3)$
$\hat{R} = 0.4$	$k_{0,1} = 1.0 \times 10^{12}(hr^{-1})$	$k_{0,1} = 1.0 \times 10^{12}(hr^{-1})$
$\hat{f}_1 = 0.94$	$\hat{f}_2 = 0.96$	$\hat{f}_3 = 0.98$

decision variables reach the lower specified limit . Note that the lower limits were set with some trial and error. Now we substitute this solution obtained from global search method.

$$x_1^* = 0.4 \quad x_2^* = 0.2 \quad x_3^* = 0.8$$

$$x_4^* = 0.99 \quad x_5^* = 0.245622 \quad x_6^* = 0.8$$

$$x_7^* = 0.99 \quad x_8^* = 0.162969 \quad x_9^* = 0.8$$

into steady state equations [Equations(2.28)-(2.36)], we find that execution of the root finding routine of a symbolic manipulator yields negative concentration of species A for first and second reactors; hence we solve the equations again by keeping dimensionless temperature values fixed and accepting equality constraint to be satisfied is it numerical value when evaluates coming close to zero. At the end of these computations, we find the optimum solution as,

$$x_1^* = 0.00723 \qquad x_2^* = 0.999871 \quad x_3^* = 0.738382$$
$$x_4^* = 0.0100397 \quad x_5^* = 0.99987 \qquad x_6^* = 0.736031$$
$$x_7^* = 0.0159364 \quad x_8^* = 0.999867 \quad x_9^* = 0.741178$$

The numerical values of equality constraints come out as

$$\forall j; \qquad g_j^* = 0.0; \qquad j = \{1, 2, \cdots, 9\}$$

Note that we started with initial guess given as optimum solution obtained using global search method and solved the equalities using numerical solver for algebraic equations. Also for inequalities, we obtain

$$g_{10}^* \equiv 0.0671562 > 0; \qquad g_{11}^* \equiv 0.0500097 > 0; \qquad g_{12}^* \equiv 0.0360596 > 0$$

Lastly, we notice that the sharp cut algorithm required only four iterations to converge while fifth and sixth repeated the same solution confirming that

the limit point has been reached. While the solver for algebraic equations using symbolic manipulator (MATHEMATICA, Wolfram Research) needed merely nine iterations and converged to a local optimum solution.

Here superscript $*$ implies evaluation of constraint at local optimum solution. This demonstrates that even for relatively large sized nonlinear program, the method works well. We applied the Karush-Kuhn-Tucker (KKT) conditions to the program, which are

$$\begin{aligned}
\frac{\partial}{\partial X_i}\mathbf{L}(\mathbf{x},\vartheta,\xi) &= 0 \\
\frac{\partial}{\partial \xi}\mathbf{L}(\mathbf{x},\vartheta,\xi) &= 0 \\
\vartheta_j g_j(\mathbf{X}) &= 0; \qquad j = 1,2,\cdots,p \\
g_j(\mathbf{X}) &\geq 0 \\
\vartheta_j &\geq 0
\end{aligned} \tag{2.42}$$

Here ϑ is multiplier associated with inequalities and ξ are Lagrangian multipliers associated with equality constraints. We find that the KKT conditions are satisfied as all equality constraints are close to zero, and, the KKT condition for equalities is satisfied as all partial derivatives w.r.t. Lagragian multipliers are very close to or are zero. Thus, for the Lagrangian function

L, we have,

$$\begin{aligned} \mathbf{L} &= \hat{\phi}_0 + \sum_{j=1}^{9} \xi_j g_j + \sum_{j=10}^{12} \vartheta_j g_j; \\ \partial_{\xi_l} \mathbf{L}^* &\approx 0.0, \qquad l = 1, 2, \cdots, 9; \qquad \vartheta_j > 0, \quad j = 10, 11, 12. \end{aligned}$$

We accept this solution as an optimum solution. Here, the computational complexity was kept to a minimum dividing search interval of each decision variable into three parts. The modification we make is due to lesser number of divisions of search interval as for nine variables the divisions would yield $NDIV^9$ points. This means if we were to consider six divisions of each interval the number of points generated would be 6^9 (or 10077696), the symbolic manipulator (MATHEMATICA software of Wolfram Research) finds too large to handle. The lower limit is set to current LP solution after each execution of LP solver and re-divide the interval to a new vector point to generate the two sharp cuts.

2.14 Conclusions

In this chapter we presented the new type of cutting plane algorithm which has super linear rate of convergence. The new algorithm is an intersection

set of three cutting planes which are Kelley's cutting plane and the first and second sharp cuts. The first sharp cut reverses signs of gradient vector in few iterations and the second sharp cut, which is almost parallel to the hyper tangent to unique global optimum solution for convex program and local optimum solution for non-convex program gives an extra cut and accelerates the rate of convergence first to super linear rate and then to super super linear rate and requires less number of iterations. The algorithm now reaches the stationary point and is at least as fast as Kelley's algorithm.

Remarks. Here we give a few comments on this new method to be used in solving highly nonlinear and relatively large sized optimization programs. Here a large sized program is one having large number of decision variables and constraints.

To summarize, some more points are made in order to explain why the sharp cut algorithm is a better option to solve nonlinear (and non-convex) programs having box constraints for the following reasons.

1. The sharp cut usually requires less number of iterations (usually 2 to 4 times faster than Kelleys algorithm).

2. There can be cases where the sharp cut algorithm reaches a non-trivial solution, Kelleys algorithm yields a trivial solution due to a few decision

variables becoming zero when optimum solution is obtained.

3. In the end, the sharp cut algorithm becomes super superlinear and reduces the large number of iterations required by Kelleys algorithm.

4. The rate of convergence of Kelleys algorithm is usually sublinear for most problems while sharp cut algorithm is at least linear even at first iteration.

5. When numbers of constraints accumulate, the Kelleys cut algorithm experiences cycling of LP solver. The randomly generated weights used in the sharp cut algorithm eliminate the possibility of cycling.

6. The stationary solution is obtained as defined using the sharp cut algorithm; while Kelleys algorithm keeps cycling for a large number of iterations, and does not reach a stationary point.

7. When we employ the intersection set of sharp cut algorithm, we observe the following. It is the second sharp cut of the intersection set which forms a monotonic sequence of polyhedral set formed by the current set of linear inequality constraints, which continues its descent towards the local optimum point even when Kelleys cut and first sharp cut become parallel to those in previous iterations. and finally touches.

8. Thus, by relaxing the assumption of convexity for most violated constraint, we find that the sharp cut algorithm works well for non-convex programs. However, more research is required to analyze this in detail and will be taken up for further study in another publication.

2.15 Appendices

2.15.1 Appendix-I

We persent and prove a theorem to establish existence of a unique stationary point to an optimization program, provided sequence of optimum solutions genertated in successive iterations converge to it.

Theorem 1.

Let Q_l be a sequence of local optimum solutions [Eq. (2.11)] generated during iterations of SLP.

(i) For every sequence Q_l, $\bar{Q}_l$ is closure of set Q_l, $l \in \mathbf{P}_l$ in metric space $\mathbf{L}\Omega - \Omega$ and then,

$$diam \;\; \bar{Q}_l = diam \;\; Q_l \tag{2.43}$$

(ii) The sequence κ_l is a sequence of compact sets in $\mathbf{L}\Omega - \Omega$ consisting of a sequence of optimum solutions generated during successive iterations of SLP such that $\kappa_l \supset \kappa_{l+1}$ where $l \in \mathbf{P}_l$. If

$$\lim_{l \to \infty} diam \;\; Q_l = 0,$$

then $\cap_{l=0}^{\infty} \kappa_l$ consists of exactly one point. $\diamondsuit$

Proof. When the sequence of local optimum solutions

$$\{\forall i \in \{1, \cdots, n\} | \{\alpha_{i,0}, \alpha_{i,1}, \cdots, \alpha_{i,l}\}\}$$

until l^{th} (current) iteration is monotonically increasing or decreasing, the sequence $Q_l = \{l \in \mathbf{P}_l | q_l\}$ is called a subsequence of $\{l \in P_\infty | q_l\}$. And by definition, whenever Q_l converges after k iterations, the limit point reached is subsequential limit of $\{q_l | l \to \infty\}$.

By definition, for a metric space $\mathbf{L}\Omega - \Omega$ if $Q_l \subset \mathbf{L}\Omega - \Omega$ and if Q_l' denotes all limit points of subsequences of Q_l, $l \in \mathbf{P}_l$ then the closure of Q_l is the set $\bar{Q}_l = Q_l \cup Q_l'$.

Further, let $\hat{Q}_l$ be a subset of metric space $\mathbf{L}\Omega - \Omega$ and $\mathbf{S} = \{\alpha_{i,l}, \alpha_{i,l+1} \in \hat{Q}_l | d(\alpha_{i,l}, \alpha_{i,l+1})\}$, $d(\alpha_{i,l}, \alpha_{i,l+1}) = |\alpha_{i,l} - \alpha_{i,l+1}|$; $l \in \mathbf{P}_l$, then the supremum of $\mathbf{S}$ is the diameter (Rudin, 1976) of $\hat{Q}_l$.

The sequence Q_l is a Cauchy sequence, iff,

$$\lim_{l \to \infty} diam \;\; Q_l = 0$$

(i) By perturbing the solution points arbitrarily in the sequence Q_l, we obtain another point set $\{\hat{\alpha}_{i,l}, \hat{\alpha}_{i,l+1}\}$. Starting with a relation, $Q_l \subset \bar{Q}_l$, we observe that

$$d(\alpha_{i,l}, \alpha_{i,l+1}) \leq d(\alpha_{i,l}, \hat{\alpha}_{i,l}) + d(\hat{\alpha}_{i,l}, \hat{\alpha}_{i,l+1}) + d(\hat{\alpha}_{i,l+1}, \alpha_{i,l+1}).$$

For an arbitrary $\epsilon > 0$, we then have $d(\alpha_{i,l}, \hat{\alpha}_{i,l}) < \epsilon$ and $d(\alpha_{i,l+1}, \hat{\alpha}_{i,l+1}) < \epsilon$. This establishes Eq. (2.43) for $\epsilon \to 0$.

(ii) Let $\bar{\kappa} = \cap_{l=0}^{\infty} \kappa_l$. With $\{\kappa_l\}$ a collection of compact subsets that occur during successive calls to LP solver with an optimum solution generated every l^{th} iteration. Further, any intersection of this finite subcollection of $\{\kappa_l; l \in \mathbf{P}_l\} \subset \mathbf{L}\Omega - \Omega$ is non-empty. It then follows that $\bar{\kappa} = \cap_{l=0}^{\infty} \kappa_l$ is non-empty (Rudin, 1976).

By induction, if κ_l contains more than one point, then $diam \ \ \kappa_l > 0$. However, $\kappa_l \supset \bar{\kappa}$, so that $diam \ \ \kappa_l \geq diam \ \ \bar{\kappa}$. This contradicts the argument that $diam \ \ \kappa_l \to 0$. $\diamond$

2.15.2 Appendix-II

The convergence theorem for sharp cut algorithm is given here. The theorem states that the intersection set formed by Kelley's cut and two sharp cuts generates a converging Cauchy sequence and a data point set of each intersection set is contained in a convex polyhedral set for that iteration, which also forms a converging sequence.

Theorem. Convergence theorem.

The intersection set of sharp cut, whose point data set is contained in the convex polyhedral set $\tilde{\omega}(\bar{\alpha}_l, \bar{\beta}_l)$, defined in Eq. (2.25), gives rise to a monotone sequence

$$\{\forall l, l \in \mathbf{P}_\infty | \{\tilde{\omega}_1, \tilde{\omega}_2, \cdots, \tilde{\omega}_l\}\}$$

such that a data point set in $\{l \to \infty | \tilde{\omega}(\bar{\alpha}_l, \bar{\beta}_l)\}$ and $\bar{\alpha}_\infty = \bar{\alpha}^*$, $\bar{\beta}_l \in \emptyset$ as $l \to \infty$ minimizes $\hat{\phi}_0(\mathbf{x}) = \mathbf{c}^t\mathbf{x}$ on $\tilde{\omega}(\bar{\alpha}_l, \bar{\beta}_l)$. Then, this sequence converges onto a solution $\bar{\alpha}^*$, which is the local optimum solution. $\diamond$

Proof.

We assume that when we identify a constraint using a vector point in Eq. (2.18), this function $\phi_r(\mathbf{x})$ is convex in the search interval given by box constraints for decision variables. Using vector point in Eq. (2.5) and then

using the new vector point in Eq. (2.18), we generate two cutting hyperplanes given by $\mathbf{u}_l \geq 0$ [Eq. (2.6)], $\mathbf{v}_l \geq 0$ [Eq. 2.21] and $\mathbf{w}_l \geq 0$ [Eq. (2.24)]. The three cutting hyperplanes form an intersection set whose point set is contained in the convex polyhedral set $\tilde{\omega}(\bar{\alpha}_l, \bar{\beta}_l)$ at the end of computation for l^{th} iteration.

For function $\phi_r(\mathbf{x})$, there exists an arbitrary constant ρ_1 such that

$$||\sum_{i=1}^{n} \nabla_{x_i} \phi_r(\bar{\alpha}_l)(x_i - \alpha_{i,l})|| \geq \rho_1; \qquad \forall \mathbf{x} \in \bigcup_{l=1}^{l=\infty} \{\mathbf{u}_l \geq 0\}, \tag{2.44}$$

another constant ρ_2 such that

$$||\sum_{i=1}^{n} \nabla_{x_i} \phi_r(\bar{\beta}_l)(x_i - \beta_{i,l})|| \geq \rho_2; \qquad \forall \mathbf{x} \in \bigcup_{l=1}^{l=\infty} \{\mathbf{v}_l \geq 0\}, \tag{2.45}$$

and, a third constant ρ_3 such that

$$||\sum_{i=1}^{n} \nabla_{x_i} \phi_r(\bar{\psi}_l)(x_i - \psi_{i,l})|| \geq \rho_3; \qquad \forall \mathbf{x} \in \bigcup_{l=1}^{l=\infty} \{\mathbf{w}_l \geq 0\}. \tag{2.46}$$

Note that Eqs. (2.44)-(2.46) are the conditions for preventing occurrence of degeneracy in LP statement. The cutting hyperplanes in Eqs. (2.6) and (2.19) act as supporting hyperplanes (Boyd and Vandenberghe, 2004) to the identified constraint and the norms of the corresponding gradient terms satisfy the relations stated above. We have already defined a point set $\mathbf{L}\Omega$ in Eq. (2.12) and Ω in Eq. (2.13) with $\Omega \subset \tilde{\omega} \subset \mathbf{L}\Omega$.

The identified constraint $\phi_r(\mathbf{x}) \geq 0$ corresponds to a new vector point denoted as $\bar{\beta}_l = \bar{\nu}_\sigma$ [Eq. (2.18)] and is a row vector. The sequence of convex functions $\tilde{\omega}_l$ will obey the relation $\tilde{\omega}(\bar{\alpha}_l, \bar{\beta}_l) \subset \tilde{\omega}(\bar{\alpha}_{l-1}, \bar{\beta}_{l-1})$ for a call to constrained optimization procedure or LP solver, for every l^{th} iteration. Let the point sets in $\tilde{\omega}(\bar{\alpha}_l, \bar{\beta}_l)$ minimize the function $\hat{\phi}_0(\mathbf{x}) = \mathbf{c}^t\mathbf{x}$. Then

$$\tilde{\omega}(\bar{\alpha}_l, \bar{\beta}_l) = \tilde{\omega}(\bar{\alpha}_{l-1}, \bar{\beta}_{l-1}) \cap \{\mathbf{x} | \mathbf{u}_l \geq 0 \bigcap \mathbf{v}_l \geq 0 \bigcap \mathbf{w}_l \geq 0\}, \tag{2.47}$$

Every l^{th} point data set generates an optimum solution $\bar{\alpha}_l$, and we obtain a monotone sequence $\hat{Q}_l$ that minimizes $\hat{\phi}_0(\mathbf{x}) = \mathbf{c}^t\mathbf{x}$ on $\tilde{\omega}_k$. Thus, we obtain a Cauchy subsequence $\{\bar{\alpha}_l\}$ that converges on to $\bar{\alpha}^*$ near termination point, with $\{\bar{\alpha}^* \in \hat{Q}_l | l \in \mathbf{P}_\infty\}$, such that $\{\forall \mathbf{x} \in \Omega | \mathbf{c}^t\bar{\alpha}^* < \mathbf{c}^t\mathbf{x}\}$. The sequence $\hat{Q}_l$ thus converges onto a solution $\bar{\alpha}^*$ which is the optimum solution. $\diamondsuit$

2.15.3 Appendix-III

In this section, we establish that the Cauchy sequence of optimum solution points reaches a numerical limit when the cutting hyperplanes generated by sharp cut algorithm are in close vicinity of the constraint hypersurface.

Remark The initial "jump" causes a large or significant reduction in iterations of SLP and it is difficult at times to accumulate sufficient number of optimum solution points that can be used to verify properties of the generated sequence. However, even for a small number of points generated, we observe the following.

Definition A.1. Sequence of optimum solution points. Consider a sequence of points,

$$Q_l = \{\bar{\alpha}_0, \bar{\alpha}_1, \bar{\alpha}_2, \cdots, \bar{\alpha}_l\}, \quad l = 0, 1, \cdots, l, \cdots, K,$$

which for l^{th} iteration satisfy the relation

$$g^o_{r,l} < g^o_{r,(l-1)}; \qquad \text{where} \qquad \forall l \in \mathbf{P}_\infty | \mathbf{u}_l \geq 0.$$

The local minimum for a non-convex program is defined as

$$\psi = \{l \in \mathbf{P}_\infty | \mathbf{x} \in \tilde{\omega}(\bar{\alpha}_l, \bar{\beta}_l) | Min \hat{\phi}_0(\mathbf{x})\} \tag{2.48}$$

which occurs at

$$\tilde{\mathbf{x}}_l = \bar{\alpha}^* \in \mathbf{B} = \{\forall \mathbf{x}_l | a_i \leq \mathbf{x}_{l,i} \leq b_i\}$$

with **B** denoting box constraints for decision variables. Let

$$Q_0 \equiv \{\bar{\alpha}_l, \bar{\alpha}_{l+1}, \cdots, \bar{\alpha}_{l_\infty}, \cdots, \bar{\alpha}_K\}$$

be a Cauchy sequence of optimum solutions generated using sharp cut, where $l_\infty \in \mathbf{P}_\infty$. Let $\bar{\alpha}_0 = \{\forall j | b_j\}$ be the first point in the sequence which lies in convex polyhedral set $\tilde{\omega}(\bar{\alpha}_0, \bar{\beta}_0, \bar{\psi}_0)$ corresponding to Eq. (2.25). We call this sequence a sampled sequence, if a few random points are picked up from Q_0; else any subsequence from $l > 0$ to $l = l_\infty < K$ if K iterations were required for SLP to reach a limit point $\bar{\alpha}_\infty = \bar{\alpha}^*$. Note that $\bar{\alpha}_{l_\infty}$ is a solution point which is in the dense set near local optimum solution $\bar{\alpha}^*$. $\diamond$

Theorem. Convergence of a sequence to a limit point.

Extending the treatment given by Shubert (1972), we see that for a sequence

$$\mathbf{G} \equiv \{\bar{\alpha}_l, \bar{\alpha}_{l+1}, \cdots, \bar{\alpha}_{l_\infty}, \cdots, \bar{\alpha}_K\} \tag{2.49}$$

of optimum solutions from

$$\forall l \in \{1, 2, \cdots, l_\infty\} | \mathbf{x}_l \in \{\forall i | a_i \leq x_{i,l} \leq b_i; i = 1, 2, \cdots, n\}$$

the applicable recursive relation is

$$\Lambda_{l+1}(\mathbf{x}) = \eta_l, \quad l = 1, 2, \cdots, l_\infty \tag{2.50}$$

else arbitrary, with

$$\eta_l = Min\{\forall i | a_i \leq x_{l,i} \leq b_i\}\Lambda_l(\mathbf{x}); l \in \{1, 2, \cdots, l_\infty\} \tag{2.51}$$

and $\Lambda_l(\mathbf{x})$ being

$$\Lambda_l(\mathbf{x}) = Min_{l=1,2,\cdots,l_\infty}\{\hat{\phi}_0(\mathbf{x} - \mathbf{x}_l) + C\|\mathbf{x} - \mathbf{x}_l\|\} \tag{2.52}$$

In Eq. (2.52), the constant C is defined as

$$|\hat{\phi}_0(\mathbf{x}) - \hat{\phi}_0(\mathbf{x}_l)| \leq C|\mathbf{x} - \mathbf{x}_l|. \tag{2.53}$$

The sampled sequence in Eq. (2.49) converges to a limit point when $\hat{\phi}_0(\mathbf{x}_l) \rightarrow \hat{\phi}_0^*$ and $\eta_l \downarrow \bar{\alpha}^*$ as $l \rightarrow \infty$. $\diamond$

Proof.

Let $l \in \mathbf{G}|Q_l$ be a set considering all sampled points as given in Eq. (2.49), which are different. Hence we say that there is a limit point in $\{\forall i | a_i \leq x_i \leq b_i; i = 1, 2, \cdots, n\}$. For arbitrary and positive ϵ^+ and a limit point

$$\mathbf{z}_l = \bar{\alpha}_l | l \rightarrow \infty, \tag{2.54}$$

we say that $\hat{\phi}_0(\mathbf{z}_l) > \eta - \epsilon^+$ for $l \in \mathbf{G}$, where $\eta = \underset{l\rightarrow\infty}{lim}\eta_l$ for $l \in \mathbf{P}_\infty$. This can be proved by induction (Shubert, 1972). Further from Eqs. (2.51) and (2.52), and $\{\forall i | a_i \leq x_i \leq b_i; i = 1, 2, \cdots, n\}$, we have $l \in \mathbf{G}|\hat{\phi}_0(\bar{\alpha}_l) \leq \eta$ and

$l \in P_\infty | \hat{\phi}_0(\bar{\alpha}_l) < \eta_l$. We must then have $\hat{\phi}_0(\mathbf{z}_l) = \eta$ for all limit points $\mathbf{z}_l$ [Eq. (2.54)] of the sampling sequence. From Eq. (2.53), it follows that the function $\hat{\phi}_0(\bar{\alpha}_l)$ must be continuous on the interval $\{\forall i | a_i < x_i < b_i; i = 1, 2, \cdots, n\}$ and it is evident that $\hat{\phi}_0(\mathbf{x}) < \psi < \eta$. The proof is quite similar to that given by Shubert (1972) and is not provided here for the sake of brevity. $\diamond$

2.16 Notation

a_i lower limit on range of i^{th} decision variable.

b_i upper limit on range of i^{th} decision variable.

$\mathbf{B}$ box constraint defining search range and forming initial hypercube.

c_i i^{th} coefficient in row vector of constants in linearized objective function $\hat{\phi}_o(\mathbf{x})$.

$\mathbf{c}$ vector of coefficients in linearized objective function $\hat{\phi}_o(\mathbf{x})$.

C a Lipschitz constant given in Eq. (2.53).

$\mathbf{E}_{0,l}$ point set generated given by Eq. (2.17) in l^{th} iteration of SLP.

$\mathbf{E}_{1,l}$ point set to form a level hypersurface in l^{th} iteration as a second sharp cut [Eq. (2.23)].

$g^o_{r,l}$ function value of most violated constraint $\phi_r(\mathbf{x})$ evaluated at $\bar{\alpha}_{l-1}$.

$\mathbf{G}$ a dense set of optimum solution points which is a subsequence [Eq. (2.49)].

l_∞ a limit point nearing termination of sharp cut algorithm.

K number of iterations after which termination of sharp cut algorithm occurs

$\mathbf{L}\Omega$ feasible region of local optimum solutions formed by cutting planes.

n integer giving number of decision variables of program in Eq. (2.3).

p integer giving total number of inequality constraints as given in Eq. (2.3).

$\mathbf{P}_l$ index set given in Eq. (2.8).

$\mathbf{P}_\infty$ index set given in Eq. (2.9).

Q_l sequence of optimum solution values formed in l^{th} iteration of SLP.

$\bar{Q}_l$ closure of sequence Q_l.

$\mathbf{S}$	a set of distance metrics used in defining diameter of a sequence.
$\mathbf{u}_l$	Kelley's cut in l^{th} iteration given by Eq. (2.6).
$\mathbf{v}_l$	first sharp (weighted) cut in l^{th} iteration given by Eq. (2.21).
$\mathbf{w}_l$	second sharp cut in l^{th} iteration given by Eq. (2.24).
w_0	weight used in causing "initial jump".
$w_{1,l}$	weight (random number) used to scale $\psi^o_{r,l}$ [Eq. (2.21)].
$w'_{1,l}$	weight vector (random numbers) used to scale $\bar{\nu}_\sigma$ [Eq. (2.21)].
x_i	i^{th} decision of optimization program
$\mathbf{x}$	vector of decision variables in $\Re^n$.
$\mathbf{y}$	vector of decision variables in $\Re^n$ lying in convex set $\tilde{\omega}$ [Eq.(2.26)].
$\mathbf{z}_l$	limit point as defined in Eq. (2.54).
$\mathbf{Z}^o$	box constraint denoted as constraint implying a hypercube.
$\mathbf{Z}^l$	feasible region as cut hypercube remaining in l^{th} iteration.

Greek symbols.

$\alpha_{i,l}$	optimum solution value of i^{th} decision variable in l^{th} iteration of SLP.
$\bar{\alpha}_l$	vector of optimum solutions as obtained in l^{th} iteration of SLP.
$\bar{\alpha}^*$	optimum solution obtained by global search using sharp cut and stationary point of program as defined by Eq. (1)
$\beta_{i,l}$	vector point obtained to devise first sharp cut as defined in Eq. (1)
$\bar{\beta}_l$	vector point used after multiplication with weights to form first and second sharp cuts.
$\hat{\delta}$	parameter in sharp cut algorithm to control steep descent for the second sharp cut as given in Eq. (2.23).
$\delta_{i,-}$, $\delta_{i,+}$	divisions of search interval given by box constraint set $\mathbf{B}$ on either side of optimum solution $\bar{\alpha}_l$ in l^{th} iteration of SLP.
ϵ	an arbitrary positive real number to define convergence limit as given in Eq. (2.10).
$\hat{\theta}_i$	i^{th} multi-nomial coefficient fitted for a linear multi-variate hyperplane.

η_l	deviational value defined in Eq. (2.52) to prove convergence to a limit point.
ϑ	integer exponent of 10 as order of magnitude of $g^o_{r,l}$.
κ_l	sequence of compact sets for l^{th} iteration of SLP in $\mathbf{L}\Omega - \Omega$.
$\bar{\kappa}$	limit point of sequence κ_l.
λ	a scalar lying in interval $[0, 1]$ to express convexity relation (see Eq.(2.26)).
μ	arbitrary vector point lying in domain of $\mathbf{L}\Omega - \Omega$.
ρ_1-ρ_3	real numbers as constants forming right hand side coefficients in Eqs. (2.44)-(2.46).
$\bar{\nu}$	vector points lying in box constraint $\mathbf{B}$ on either side of $\bar{\alpha}_{l-1}$.
ς	numerical values of the most violated constraint at generated vector point set [Eq. (2.17)].
$\phi_0(\mathbf{x})$	objective function of program in Eq. (2.3).
$\phi_j(\mathbf{x})$	j^{th} inequality costraint in program statement in Eq. (2.3).
$\hat{\phi}_0(\mathbf{x})$	linearized objective function.
χ	variable to measure rate of convergence and equal to $\lvert g^o_{r,l} - g^o_{r,l-1}\rvert$.
$\psi^o_{r,l}$	minimum function value of the most violated constraint corresponding to minimum linearized objective function value using generated vector points to form a new cut.

$\omega_l(\bar{\alpha}_l)$	convex polyhedral set used in proving convergence result.
$\hat{\omega}_l$	convex polyhedral set formed for intersection of Kelley's and first sharp cuts.
$\omega_0(\bar{\alpha}_0)$	convex polyhedral set used in iteration '1', generated in '0' iteration.
$\omega_1(\bar{\alpha}_1)$	convex polyhedral set used in iteration '2'.
$\tilde{\omega}_l(\bar{\alpha}_l, \beta_l)$	convex polyhedral set formed in l^{th} iteration of SLP.
Ω	constraint set corresponding to the most violated constraint.
Φ	set of equally spaced divisions on either side of $\bar{\alpha}_{l-1}$ within box constraint.
$\tilde{\Phi}$	deviational value used to analyze a limit point of sequence of optimum solution points [Eq. (2.50)].

Symbols.

$\aleph_{\pm}$ number of divisions on either side of $\bar{\alpha}_{l-1}$ in box $\mathbf{B}$.

$\hbar$ i^{th} index variable in selecting a point for "initial jump" as in Eq. (1).

ℓ index variable for vector point set $\mathbf{E}_{0,1}$ in Eq. (2.20).

$\wp_{\pm}$ number of divisions lying on either side of box constraint ($\mathbf{B}$) of a vector point $\bar{\alpha}_l$ in l^{th} iteration as given in Eq. (2.16).

$\Re$ set of real numbers.

$\emptyset$ an empty set (containing no elements).

∇ operator to obtain gradient set of the most violated constraint $\phi_r(\mathbf{x})$.

Superscripts.

o indicates evaluation of $\phi_r(\mathbf{x})$ at a vector point.

Subscripts.

i i^{th} decision variable of optimization program

j j^{th} inequality constraint of program as defined in Eq. (1)

l integer index giving l^{th} iteration of SLP algorithm.

r integer index corresponding to r^{th} constraint which is the most violated constraint.

2.17 References

Balas E., 1971, Intersection Cuts-A New Type of Cutting Planes for Integer Programming, Operations Research, **19**(1), 19-39.

Beale, E.M.L., 1955, Cycling in the dual simplex algorithm, Naval Research Logistics Quarterly, **2**, 269 - 275.

Bertsekas D. and Nedic A., 2003, Convex Analysis and Optimization Athena Scientific.

Borkar, V.S.; Meyn, S.P., 1998, Stability and convergence of stochastic approximation using the ODE method, Proceedings of the 37th IEEE Conference on Decision and Control, Volume 1, 277 - 282.

Borkar V.S. and Meyn, S.P., 2000, The O.D.E. method for convergence of stochastic approximation and reinforcement learning, SIAM J. Control Optim., **38**, 447–469

Boyd S., Vandenberghe L., 2004, Convex Optimization, Cambridge University Press.

Byrd, R.H., Gould, N., Nocedal, J., Waltz R.A., 2005, On the convergence of successive linear programming algorithms, SIAM J. Opt., **16** (2),

471-489.

Cheney and Goldstein (1959) Newton's method of convex programming and Tchebycheff approximation, Numer. Math., **1**, 253-268.

Edgar, T.F. and Himmelablau, D.M. (1988), Optimization of Chemical Processes, McGraw-Hill Inc.

Floudas, C.A., Akrotirianakis, I.G., Caratzoulas, S., Meyer, C.A., Kallrath, J., 2005, Global optimization in the 21st century: Advances and challenges, Computers & Chemical Engineering, **29**, 1185-1202.

Gass, S.I., 1979, Comments on the possibility of cycling with the simplex method, Operations Research **27**, 848-852.

Georgiorgis D.I., Georgiadis M., Bowen G., Pantelides C.C., Pistikopoulos E.N., 2006, Dynamic oil and gas production optimization via explicit reservoir simulation, Computer Aided Chemical Engineering, **21**, 179-184.

Goffin J.-L. and Vial J.-P., A two cut approach in the analytic center cutting plane method, LogiLab Technical Report, 96.1, May 15, 1996 (Revised June 30, 1997).

Goffin J.-L. and Vial J.-P., Shallow, deep and very deep cuts in the analytic center cutting plane method, LogiLab Technical Report, 97.6, August 1997, (Revised April 29, 1998).

Grötschel, M., 2004, The Sharpest Cut: The Impact of Manfred Padberg and His Work, Series on Optimization, Vol. 4, MPS-SIAM.

Harjunkoski I., Jain V., and Grossman, I.E., 2000, Hybrid mixed– integer / constraint logic programming strategies for solving scheduling and combinatorial optimization problems, Computers & Chemical Engineering, **24**, 337-343.

Higham N.J., 1996, Accuracy and Stability of Numerical Algorithms, 1st Edition, SIAM, Philadelphia.

Hoza, M., Stadtherr, M. A., 1993, An improved watchdog line search for successive quadratic programming, Computers & Chemical Engineering, **17**, 943-947.

Kelley Jr. J.E., 1960, The cutting plane method for solving convex programs, S.I.A.M. J., **8**, 703-712.

Letchford A.N., 2002, Totally tight Chvtal-Gomory cuts, Operations Re-

search Letters, **30**, 71-73.

Li, D., Sun X.-L., Wang F. L., 2006, Convergent Lagrangian and Contour Cut Method for Nonlinear Integer Programming with a Quadratic Objective Function, S.I.A.M. J. Opt., **17**,(2), 372 - 400.

Lucia A., Yang F., 2004, Solving distillation problems by terrain methods, Computers & Chemical Engineering, **28**, 2541-2545.

Luoy Z-Q., 1997, Analysis of a cutting plane method that uses weighted analytic center and multiple cuts, S.I.A.M. J. Optim., **7** (3), 697-716.

Mangaserian, O.L., 1969, Nonlinear Programming, McGraw-Hill Company, NY.

Miri T., Tsoukalas A., Bakalis S., Pistikopoulos E.N., Rustem B., Fryer P.J., 2008, Global optimization of process conditions in batch thermal sterilization of food, Journal of Food Engineering, **87**, 485-494.

Rajesh J. K., Gupta S. K., Rangaiah G. P., Ray A. K., 2001, Multi-objective optimization of industrial hydrogen plants, Chemical Engineering Science, **56**, 999-1010.

Ravindran,A., Ragsdell, K. M., Reklaitis G. V., 1983, Engineering Optimization, I Edn., Wiley-Interscience.

Rudin, Walter, Principles of Mathematical Analysis, III Edition, McGraw-Hill International Edition, (1976).

Shubert B.O., 1972, A sequential method seeking the global maximum of a function, S.I.A.M. J. Numer. Anal., **9**, 379-388.

Srinivasan, B., Biegler, L.T., Bonvin, D., 2008, Tracking the necessary conditions of optimality with changing set of active constraints using a barrier-penalty function, Computers & Chemical Engineering, **32**, 572-579.

Terlaky T. (Ed.), 1996, Interior Point Methods of Mathematical Programming (Applied Optimization), 1st Edition, Springer, NY.

Varma V. A., Pekny J.F., Blau G. E., Reklaitis G. V., 2008, A framework for addressing stochastic and combinatorial aspects of scheduling and resource allocation in pharmaceutical R&D pipelines Computers & Chemical Engineering, **32**, 1000-1015.

MATHEMATICA software, Wolfram Research.

Zhang J., Kim N.-H., and Lasdon L., 1985, An Improved Successive Linear Programming Algorithm, Management Science, **31**, 1312-1331.

2.18 Tables

Table 2.2: Numerical experience to solve Example 1 using sharp cut algorithm.

Iter No	$\alpha_{1,l}$	$\alpha_{2,l}$	$\alpha_{3,l}$	$-g^o_{r,l}$	$\hat{\phi}_0(\bar{\alpha}_l)$
1	2.5	0.0	0.738295	4.22952	21.9559
2*	1.56756	0.0226037	0.589861	0.907893	-16.6638
3	1.21341	0.0340357	0.531225	0.127868	-14.6902
4	1.14323	0.0363009	0.519607	0.00502017	-14.2756
5	1.14025	0.0363974	0.519112	9.11×10^{-6}	-14.2579
6	1.14024	0.0363976	0.519111	3.02×10^{-11}	-14.2579
7	1.14024	0.0363976	0.519111	4.44×10^{-16}	-14.2579

Table 2.3: Numerical experience to solve Example 1 using sharp cut algorithm.

Iter No	$\beta_{1,l}$	$\beta_{2,l}$	$\beta_{3,l}$	$-\psi^o_{r,l}$	$-g^o_{r,l}$	$\hat{\phi}_0(\bar{\alpha}_l)$	$\hat{\phi}_0(\bar{\beta}_l)$
1	2.28	0.0	0.7	0.0142136	4.22952	-22.2659	-20.0
2*	2.85	0.9	1.4	0.48	0.907893	-16.6638	-46.75
3	2.85	0.9	1.4	0.48	0.127868	-14.6902	-46.75
4	2.675	0.9	1.456	0.48	0.00502017	-14.2756	-46.75
5	2.675	0.9	1.456	0.48	9.11×10^{-6}	-14.2579	-46.75
6	2.625	1.0	1.428	0.06	0	-14.2579	-48.0
7	2.625	1.0	1.428	0.06	0	-14.2579	-48.0

**Note.* After second iteration of SLP algorithm, the gradient vector has reversed sign.

Table 2.4: Cutting planes generated for Example 1.

Iter	Cutting planes	χ_l	Convergence*
1	$\mathbf{u}_1 \equiv 4.22952 - 6.0x_1 - 2.21489x_2 + 8.85954x_3 \geq 0$	–	
	$\mathbf{v}_1 \equiv 0.660651 + 0.331133x_1 + 0.0x_2 - 2.0x_3 \geq 0$		
	$\mathbf{w}_1 \equiv 0.66204 + 0.238555x_1 - 1.35236x_2 - 1.7045x_3 \geq 0$		
2	$\mathbf{u}_2 \equiv 0.901073 - 3.76214x_1 - 1.76958x_2 + 7.01052x_3 \geq 0$	~ 1	L
	$\mathbf{v}_2 \equiv 4.062 - 6.84x_1 - 4.2x_2 + 14.1x_3 \geq 0$		
	$\mathbf{w}_2 \equiv 0.428758 - 0.380649x_1 - 0.342705x_2 + 0.997519x_3 \geq 0$		
3	$\mathbf{u}_3 \equiv 0.127868 - 2.91218x_1 - 1.59368x_2 + 6.2726x_3 \geq 0$	~ 1	L
	$\mathbf{v}_3 \equiv 4.062 - 6.84x_1 - 4.2x_2 + 14.1x_3 \geq 0$		
	$\mathbf{w}_3 \equiv 0.380264 - 0.308106x_1 - 0.267439x_2 + 0.836095x_3 \geq 0$		
4	$\mathbf{u}_4 \equiv 0.00502017 - 2.74376x_1 - 1.55882x_2 + 6.12638x_3 \geq 0$	~ 2	SL
	$\mathbf{v}_4 \equiv 0.124668 - 6.42x_1 - 4.368x_2 + 14.772x_3 \geq 0$		
	$\mathbf{w}_4 \equiv 0.510507 - 0.0130997x_1 - 0.00559788x_2 - 0.0263927x_3 \geq 0$		
5	$\mathbf{u}_5 \equiv 9.112 \times 10^{-6} - 2.73659x_1 - 1.55734x_2 + 6.12015x_3 \geq 0$	~ 3	SL
	$\mathbf{v}_5 \equiv 0.124668 - 6.42x_1 - 4.368x_2 + 14.772x_3 \geq 0$		
	$\mathbf{w}_5 \equiv 0.3512 - 0.0628926x_1 - 0.068668x_2 + 0.213488x_3 \geq 0$		
6	$\mathbf{u}_6 \equiv 3.024 \times 10^{-11} - 2.73658x_1 - 1.55733x_2 + 6.12014x_3 \geq 0$	~ 4	SSL
	$\mathbf{v}_6 \equiv 0.701292 - 6.3x_1 - 4.284x_2 + 14.136x_3 \geq 0$		
	$\mathbf{w}_6 \equiv 0.566676 - 0.0868851x_1 + 0.068824x_2 - 0.247498x_3 \geq 0$		
7	$\mathbf{u}_7 \equiv 4.440 \times 10^{-16} - 2.73658x_1 - 1.55733x_2 + 6.12014x_3 \geq 0$	~ 5	SSL
	$\mathbf{v}_7 \equiv 0.701292 - 6.3x_1 - 4.284x_2 + 14.136x_3 \geq 0$		
	$\mathbf{w}_7 \equiv 0.43678 - 0.001253x_1 + 0.212109x_2 - 0.414286x_3 \geq 0$		

**Note.* L-linear, SL- super linear, SSL - super super linear.

Table 2.5: Comparison of Kelley's (1) and sharp (2) cuts

Example No.	Optimum (1) solution	Iter- (1) ations	Optimum (2) solution	Iter- (2) ations	Factor
1	14.9449	25	14.2579	7	3.57
2	12.3944	25	12.6126	9	2.44
3	4.5	9	4.5	7	1.29

Chapter 3

Nonconvex functions and application of the sharp cut algorithm

Abstract

Earlier we presented a variant of Kelley's cutting plane algorithm, termed as a *sharp cut*, that was found to be superior than existing methods in the literature. In this work, we show that the sharp cut when applied to a few standard nonconvex functions, gives out all optimum solutions in monotone order. From the limited literature searched, it was found that among cutting plane algorithms only the sharp cut algorithm finds all multiple optimum solutions in very few iterations, when the standard nonconvex functions were used as most violated constraints. This feature of collecting multiple optimum solutions, which is a unique feature of its kind, has been observed for the first time. Later, when the sharp cut was applied to control system design of the nonisothermal CSTR, it could find optimum solutions which were present over an uneven (or discernible rough) terrain. This is another interesting feature of the sharp cut as observed when applied to a problem of the control and optimization of nonisothermal CSTR.

3.1 Notation

Alphabetic.

A	Jacobian matrix of linearized evolution equations of reactor dynamics.
A_H	Surface area for heat transfer for cooling jacket, m^2.
a_i	Lower limit in box constraints for sharp cut algorithm.
B	Square matrix in state space form for evolution equations for control of reactor dynamics.
b_i	Upper limit in box constraints for sharp cut algorithm.
C_A	Concentration of chemical species A which is a reactant, kg/m^3.
$C_{A,0}$	Concentration of chemical species A in feed stream, kg/m^3.
c_p	Heat capacity of reaction mixture in reactor vessel, $kJ/kg \cdot (K)$.
c_J	Heat capacity of coolant fluid in cooling jacket, $kJ/kg \cdot (K)$.
E	Activation energy in Arrhenius rate form for chemical reaction, kJ/kg.
F	Generic variable representing the flowrate into and out of CSTR, m^3/hr.
F_0	Flowrate of feed stream entering the reactor vessel, m^3/hr.
ΔF_0	Variation in feed flow rate from set point value as a parameter shift.

F_J	Flowrate of coolant in cooling jacket, m^3/hr.
$g^o_{r,l}$	Numerical value of function in most violated constraint used by the sharp cut algorithm.
$H^{(l)}$	The convex polyhedra formed by linear constraint set of cutting planes in the sharp cut algorithm.
$(-\Delta H_R)$	The heat of exothermic reaction, kJ/kg.
k_0	The absorption or pre-exponential factor used in Arrhenius rate form.
n	The dimensionality index for number of decision variables in the sharp cut algorithm.
$\aleph_\pm$	The total number of vector points generated at different steps in the sharp cut algorithm.
P_l	The set of optimum solutions generated by the sharp cut algorithm.
R	The ideal gas law constant.
T	Temperature variable associated with the reactor dynamics, (K).
t	Time variable used in CSTR dynamics.

T_0	The inlet temperature of feed entering the CSTR, (K).
T_J	Temperature of coolant in cooling jacket, (K).
$T_{J,0}$	The feed temperature of coolant entering the cooling jacket, (K).
U	Overall heat transfer coefficient, $kJ/m^2(K)$.
$u(t)$	The variable representing action of stochastic optimal control in HJB equation.
$\mathbf{u}_l$	The Kelley's cutting plane computed in sharp cut algorithm in l^{th} iteration.
$\mathbf{v}_l$	The first sharp cut, which is computed in sharp cut algorithm in l^{th} iteration.
V	Volume of CSTR, m^3.
V_{Min}	The minimum head equivalent volume in reactor tank, m^3.
V_J	The volume used by coolant in the cooling jacket to compute surface area for heat transfer, m^3.
$\mathbf{w}_l$	The second sharp cut, which is computed in sharp cut algorithm in l^{th} iteration.
w_i, $w_{i,j}$	The random numbers of appropriate magnitude used to scale the cutting plane *r.h.s.*and gradient set in the sharp cut algorithm.

X_s The steady state vector for reactor dynamics.

Z^o The initial hypercube formed by the box constraints.

$Z^{(l)}$ The hypercube formed by active constraints in l^{th} iteration of the sharp cut algorithm.

Greek.

$\bar{\alpha}_l$ The optimum solution point as a vector of decision variables appearing in the sharp cut algorithm.

$\alpha^1(\cdot)$ The fraction of cost borne for randomly varying variable in stochastic optimal control.

$\alpha^2(\cdot)$ The loss function used in the theory of stochastic optimal control.

$\bar{\beta}_l$ The vector of selected vector point used to form the first sharp cut in the sharp cut algorithm.

$\delta_{\pm}$ Increment variable in the sharp cut algorithm.

$\bar{\nu}$ The vector point selected from generated vector point set in the sharp cut algorithm.

ρ_R The density of reaction mixture in CSTR, kg/m^3.

$\tilde{\phi}(\cdot)$ A function.

$\psi^o_{r,l}$ The function used to generate the second sharp cut in the optimization algorithm.

Symbols.

$\wp_{\pm}$ A variable to represent index increment in point generation in the sharp cut algorithm.

Superscripts.

o indicates evaluation of $\phi_r(\mathbf{x})$ at a vector point.

Subscripts.

i The integer index for decision variable in optimization algorithm.

j An integer index.

k The current iteration of the optimization algorithm.

l The iteration index for the sharp cut algorithm.

Min A condition expressing the minimum level head for CSTR.

3.2 Introduction

Optimization of nonlinear process systems (Bertsekas & Nedic, 2003; Floudas, Akrotirianakis, Caratzoulas, Meyer, & Kallrath, 2005; Georgiorgis, Georgiadis, Bowen, Pantelides, & Pistikopoulos, 2006; Harjunkoski, Jain, & Grossman, 2000; Lucia & Yang, 2004; Srinivasan, Biegler & Bonvin, 2008; Varma, Pekny, Blau & Reklaitis, 2008) has remained a topic of interest over past few decades and continues to be so as new methods of obtaining multiple solutions are being introduced and researched further. The problem of nonconvex programming is at the centerstage that needs to be solved effectively and faster, and developing new methods having ability to find all multiple optimum solutions remains the focus of all such efforts. Multiple optimum solutions can occur due to presence of a number of equality / inequality type of nonconvex constraints. The presence of multiple local optimum solutions (Boyd & Vandenberghe, 2004; Edgar & Himmelablau, 1988; Mangaserian, 1969; Ravindran, Ragsdell & Reklaitis, 1983) poses the most significant difficulties.

The nonconvex functions as equality or inequality constraints, when plotted as a 3D portrait is seen to possess several convex and concave subsets. The union of all these subdomains form the mappings of the standard non-

convex functions.

Consider a standard minimization program statement

$$Min \quad \hat{\phi}_0(\mathbf{x}) \equiv \sum_{i=1}^{n} c_i x_i$$
$$s.t. \phi_j(\mathbf{x}) \geq 0; \quad j = 1, 2 \cdots, p$$

where $\hat{\phi}_0(\mathbf{x})$ is the linear objective function and $\{\forall j \in \{1, 2, \cdots, p\} | \phi_j(\mathbf{x}) \geq 0\}$ are p inequality constraints. Here the inequality constraints are assumed be highly nonlinear. Let there be multiple local optimum solutions available. It is clear that there are several locally existing subdomains, which correspond to convex and concave forms. Hence, the cutting plane algorithm upon execution of solver should yield these multiple optimum solutions.Thus, the task before us is to obtain all existing multiple optimum solutions.

Towards this end, we first present a brief review of the methods of obtaining multiple optimum solutions to nonconvex program. The homotopy search method (Floudas, 1999; Sun, Liu and Wang, 2009; Bonilla, 2010) is most commonly used to go near to existing local optimum solutions. The standard nonlinear programming solvers include a variant of Newton method (Ren and Argyros, 2010; Xu and Li, 2008) and require that the initial guess

given lies within a radius of convergence. If this condition is met, then the local search method will converge upon the optimum solution. A number of cutting plane algorithms apply the method if successive linear programming (SLP) and any other such alternative to solve nonlinear programs. The cutting plane algorithms have been used to solve especially the optimization problems in refinery planning and scheduling. In addition to Kelley's (1960) cutting plane algorithm various cuts have been used in the literature on optimization theory and methods. Goffin and Vial (1997) proposed shallow, deep, and very deep cuts in solving non-differentiable convex programs. The techniques build an increasingly refined polyhedral approximation of the optimum solution set. An analytic center cutting plane method was used to solve optimization problems. Two cuts were used by Goffin and Vial (1996) when a new cutting plane and a new upper bound were introduced for the objective function at the same time. It was observed that the updating directions depend on the cosine of the metric of Dikins ellipsoids of the normals to the cut, where the acute angle in the cosine formula favors convergence. Luoy (1997) provided the analysis of a cutting plane method that uses weighted analytic center and multiple cuts. These cutting plane algorithms have not shown any ability to collect multiple optimum solutions as seen from avail-

able open literature. We wish to try to see if the sharp cut algorithm will be able to collect all optimum solutions.

As an alternative, we already know that the sharp cut algorithm can also come very close to a unique global optimum solution for a single convex constraint assuming the convexity condition is obeyed by the constraint. Hence we need to explore how the sharp cut algorithm can be used to collect all the optimum solutions after executing the optimization code. In order to find out if the sharp cut algorithm can solve the nonconvex programming problem, we selected a few standard nonconvex functions as a single inequality constraint, and, assumed a linear objective function to prepare toy examples. When the sharp cut was applied to these nonconvex inequalities it revealed a new and unique feature of the cutting plane algorithm. This can happen when the constraints of the program are nonlinear and form a nonconvex hypersurface.

In what follows we give a sequential computational procedure, and then apply the sharp cut algorithm to our toy examples having standard nonconvex functions as inequality constraints. Later we show the application example of the nonisothermal CSTR dynamics to obtain optimum steady states. Our primary objective is to carry out the numerical computations

and to show how the sharp cut algorithm has the unique feature of collecting multiple optimum solutions.

3.3 The sharp cut algorithm

In this section, we give a computational stepwise procedure to use the sharp cut to obtain local optimum solutions of a nonlinear program.

1. *Initialization.* Begin with a linear objective function $\hat{\phi}_o(\mathbf{x})$ (this is a limitation of this algorithm).

2. Write the box constraints as set $\mathbf{Z}^o$, which forms the constraint set for zeroth iteration.

$$\mathbf{Z}^o = \{\forall i | a_i < x_i < b_i; \quad i = 1, 2, \cdots, n\}$$

   The call to LP solver gives an optimum solution $\bar{\alpha}_0$.

3. *First iteration.* Set SLP iteration index $l = 1$ and compute the function value $g^o_{r,1}$ (here r^{th} constraint is the most violated constraint).

4. To descend steeply upon hypersurface of most violated constraint, first generate a number of point data sets forming a set Φ for given search

segment,

$$\begin{aligned}\Phi &\equiv \{\forall \wp_{\pm} \in \{0,1,2,\cdots,\aleph_{\pm-1}\} \ \ |\bar{\nu}_{\wp\pm}\} \\ \bar{\nu}_{\wp\pm} &= \{\forall i = 1,\cdots,n, \wp_{\pm} = 1,\cdots,\aleph_{\pm}|\alpha_{i,l} \pm \wp_{\pm}\delta_{\pm,i}\}\end{aligned}$$

where

$$\begin{aligned}\forall i|\delta_{i,-} &= (\alpha_{i,l} - a_i)/(\aleph_{-} - 1); \\ i = 1,\cdots,n|a_i &< \nu_{i,\wp_-} < b_i\end{aligned}$$

$$\begin{aligned}\forall i|\delta_{i,+} &= (b_i - \alpha_{i,l})/(\aleph_{+} - 1); \\ i = 1,\cdots,n|a_i &< \nu_{i,\wp_+} < b_i\end{aligned}$$

The point data set E_0 is generated as,

$$\begin{aligned}E_0 &\equiv \{\forall \wp_{\pm} \in \{0,1,2,\cdots,\aleph_{\pm-1}\}|\bar{\nu}_{\wp\pm}\} \\ 0 &< \varsigma_{r,\wp\pm}(\bar{\nu}_{\wp\pm}) \leq g^o_{r,l} \\ \varsigma_{r,\wp\pm}(\bar{\nu}_{\wp\pm}) &= \phi_r(\bar{\nu}_{\wp\pm}) \leq g^o_{r,l}\end{aligned}$$

5. Now, the new vector point can be generated as,

$$\psi^o_{r,l} = \{r|\sigma \in \{\wp_{\pm}\}; \ \ \bar{\nu}_{\wp\pm} \in E_0|\varsigma_{r,\sigma}(\bar{\nu}_{\sigma})|Min\{\hat{\phi}_0(\bar{\nu}_{\wp\pm})\}\} \tag{3.1}$$

where,

$$\wp_{\pm} = \{0,\cdots,\aleph_{\pm-1}\}$$

The new vector point is denoted as $\bar{\beta}_l = \bar{\nu}_\sigma$.

6. Generate the first sharp cut (this is closer to hypersurface than Kelly's cutting plane).

$$\mathbf{v}'_l(\mathbf{x}; \bar{\beta}_l) \equiv w_{1,l}\psi^o_{r,l} + \sum_{i=1}^{n} \nabla_{x_i}\phi_r(\bar{\beta}_l)(x_i - \beta_{i,l}) \geq 0 \tag{3.2}$$

The vector point $\bar{\beta}_l$ is scaled using weights.

7. In order to construct the second sharp cut, use a relation

$$\{r|\forall\sigma \in \{\wp_\pm\}|\ \bar{\nu}_\sigma|\{\varsigma_{r,\sigma}(\bar{\nu}_\sigma) \leq \delta \times \psi^o_{r,l}\} \subset E_0\}$$

and a symbolic manipulator to obtain a multi-variate fit for these points. The gradient set from data fitting comes out as

$$\forall i|1 \leq i \leq n|\rho_i$$

and then the second sharp cut can be formed as

$$\mathbf{w}_l(\mathbf{x}; \bar{\beta}_l) \equiv w_{2,l}\psi^o_{r,l} + \sum_{i=1}^{n} \rho_i(x_i - \beta_{i,l}) \geq 0 \tag{3.3}$$

Here $w_{2,l}$ is a random number between $[0.96, 0.98]$ generated for every l^{th} iteration.

8. *Formation of constraint set.* Add this to constraint set using a mathematical expression as,

$$\mathbf{Z}^1 = \mathbf{Z}^o \cap \mathbf{H}^1$$

where $\mathbf{H}^1$ is given as, $\mathbf{H}^1 = \{l = 1 | \mathbf{u}_1 \geq 0 \cap \mathbf{v}_1 \geq 0 \cap \mathbf{w}_1 \geq 0\}$. Then, a call to LP solver will give out an optimum solution, $\bar{\alpha}_1$.

9. l^{th} *SLP iteration.* Now, we present computational step for l^{th} iteration of SLP. The minimization program can be written as,

$$Min \quad \hat{\phi}_0(\mathbf{x}) = \mathbf{c^t x}$$

$$s.t. \qquad \mathbf{x} \in \mathbf{Z}^{l-1} \cap \mathbf{H}^l$$

where,

$$\mathbf{H}^l = \{l \in \mathbf{P}_l | \, (\mathbf{u}_l \geq 0 \cap \mathbf{v}_l \geq 0 \cap \mathbf{w}_l \geq 0)\}$$

The call to LP solver gives an optimum solution, $\bar{\alpha}_l$.

10. *Termination criteria.* If $g^o_{r,l}$ as $l \to \infty$ becomes zero (idealistic condition that can occur) or remains constant at some infinitesimally small value closer to machine precision limit, the program will be terminated.

Remarks.

1. For weights used in algorithm, if we generate random numbers in that interval, there will not remain any possibility of cycling behavior.

2. Setting a numerical value for constant δ needs a little trial and error. But our numerical experience shows that this is not very sensitive to

small incremental changes in its value.

3. The constants in first and second sharp cut will have to be chosen carefully that when other decision variables are put to zero, for i^{th} decision variable, the value does not cross the limits set by respective box constraint.

3.4 New revelations

The sharp cut algorithm is applied for the purpose of illustration to two variable nonconvex functions, so that the results can be seen visually.

Here, we will solve the optimization program P_1, which is given as

$$\begin{aligned} &\text{Min} \quad \phi_0(\mathbf{x}) \equiv c_1 x_1 + c_2 x_2 \\ &\text{s.t.} \quad \phi_1(\mathbf{x}) \leq 0 \end{aligned} \tag{3.4}$$

where $\mathbf{x} = \{x_1, x_2\}^T \in \Re^2$, is a two variable nonconvex function. The scalar nonconvex function is the sole inequality constraint. Since this function is nonconvex, the basic assumption used in Kelley's cutting plane algorithm is violated. Hence Kelley's algorithm is not applicable and will not give us the multiple local optimum solutions of the selected nonconvex functions. A few standard nonconvex functions are listed below.

Function 1. Ackleys function in 2D. The program statement is

$$\begin{aligned} \text{Min} \quad \hat{\phi}_0(\mathbf{x}) &= -2.0 * x_1 - x_2 \\ s.t. \quad g_1 &\equiv 0.2 - x_1 * \text{Sin}[2. * \pi * x_1] \\ &- \quad x_2 * \text{Cos}[2. * \pi * x_2] \end{aligned}$$

The box constraints are given as

$$0 \leq x_1 \leq 4.0; \qquad 0 \leq x_2 \leq 4.0$$

The sharp cut algorithm finds several solutions as function value of most violated constraint shifts between ± 0.0595865 in eighteen iterations, after which it is terminated. The function is shown in Fig. 3.1 and the computed solutions are listed in Table 3.1.

Function 2. Rastrigins function in 2D. Consider a nonlinear program

$$\begin{aligned} \textit{Min} \quad \hat{\phi}_0(\mathbf{x}) &= -x_1 - x_2 \\ s.t.\ g_1 &\equiv 0.3 + 0.1((x_1^2 + 1.2\text{Cos}[2.\pi x_1]) \\ &- \quad (x_2^2 - 0.2\text{Cos}[2\pi x_2])) \end{aligned}$$

The box constraints are given as

$$0 \leq x_1 \leq 4.0; \qquad 0 \leq x_2 \leq 1.6$$

The output from first sixteen iterations of the sharp cut algorithm is tabulated (see Table 3.2) to reach various stationary points corresponding to the level hypersurface (see Fig. 3.2) formed after arriving at the accumulation point $g^o_{r,l} = \pm 0.211828$, $l = 1, 2, \cdots, 16$. Thus the sharp cut algorithm has succeeded in emitting multiple optimizers. Note that the negative sign of $g^o_{r,l}$ implies a local maximum value, while the positive sign refers to the nonconvex function, which has a local minimum value.

Function 3. Schwefels function in 2D. Consider a nonlinear program

$$\begin{aligned} Min\hat{\phi}_0(\mathbf{x}) &= 2.0x_1 - 0.4x_2 \\ s.t.\ g_1 &\equiv (8.0x_1 - 0.5x_2 + 1.2x_1x_2 + 0.6)/90.0 \\ &+ 8.0 * \mathrm{Cos}[3.0x_1 + 0.4\sqrt{x_2} + 0.4x_1x_2] \\ &\times \mathrm{Cos}[(1.4x_2 + 0.4)/\sqrt{2}] - 0.2) \end{aligned}$$

The function profile is shown in Fig. 3.3. The box constraints are given as

$$0 \leq x_1 \leq 6.0; \qquad 0 \leq x_2 \leq 4.0$$

Thus, at a level given by $g^o_{r,l} = \pm 2.37271$, we obtain several local optimum points. The algorithm could not proceed beyond eight iterations. Its output is again given in Table 3.3.

Function 4. Griewangks function in 2D. Consider a nonlinear program

$$\begin{aligned} Min\hat{\phi}_0(\mathbf{x}) &= 2x_1 - 0.4x_2 \\ s.t. \qquad & g_1 \equiv ((x_1^2 + x_2^2)/200.0 \\ & \quad - 3.7 * \mathrm{Cos}[x_1]\mathrm{Cos}[x_2/\sqrt{2}] - 0.4) \end{aligned}$$

The function profile is shown in Fig. 3.4. The box constraints are given as

$$0 \leq x_1 \leq 8.0; \qquad 0 \leq x_2 \leq 4.0$$

The sharp cut algorithm immediately reaches to an accumulation point given by $g^o_{r,l} = \pm 0.0808988$, $l = 1, 2, \cdots, 21$ (see Table 3.4).

Now, we will try to apply this cutting plane variant to an optimal control and optimization problem.

3.5 Nonlinear dynamics of CSTR

The nonisothermal CSTR considered here carries out a first order exothermic reaction, $A \rightarrow R$, and has nonlinear dynamics.

T,he CSTR is maintained at a favorable temperature by employing a cooling jacket. The reaction rate form is of Arrhenius type and the rate

constant for step A $\rightarrow$ R is $k(T) = k_0 exp\left[\frac{-E}{RT}\right]$. The governing equations are given here.

$$\begin{aligned} V\frac{dC_A}{dt} &= F_0C_{A,0} - FC_A - Vk_0 exp\frac{-E}{RT}C_A \\ \rho c_p V\frac{dT}{dt} &= \rho c_p(F_0T_0 - FT) \\ &+ (-\Delta H)Vk_0 exp\frac{-E}{RT}C_A - UA_H(T - T_J) \\ \rho_J c_J V_J\frac{dT}{dt} &= F_J\rho_J c_J(T_{J,0} - T_J) + UA_H(T - T_J) \end{aligned} \tag{3.5}$$

The output flowrate from CSTR is set to a constant value F such that

$$F = F_0 + K_c(V - V_{\min}) \tag{3.6}$$

Now we will see how multiple optimum solutions for nonisothermal CSTR can be computed, when small shifts in input flowrates to reactor and cooling jacket (i.e. F_0 and F_J) occur. The results of numerical computations done are discussed in next section.

3.6 Numerical computations

First we derive the steady state equations and compute a steady state to show how the optimum solution with a little slack added give different reference

states at that parameter point. We can use these reference states to apply the optimal control law. This part of control and optimization application will be presented later elsewhere in detail giving rigorous theory to investigate strength of the new algorithm.

Table 3.5 shows the numerical values of physical parameters appearing in system dynamics. In the numerical computations done here, we vary the reactor input flowrate in the range 0.25 m^3/hr to 0.45 m^3/hr, while the coolant flow rate in cooling jacket is varied in the range 2.2 m^3/hr to 4.4 m^3/hr. The output steady state values are fitted using Ackley's function of a form

$$T_{J,s} = \mu_0 + x_1 \text{Sin}(2 * \pi * x_1) - x_2 \text{Cos}(2 * \pi * x_2) \tag{3.7}$$

where $T_{J,s}$ is steady state water temperature in cooling jacket.

The symbolic manipulator used in computing the constant in the above functional form after data fitting was found to be $\mu_0 = -19.1744$. We further scaled the functional form by multiplying it by 10^4, re-scaled the temperatures of reactor feed and cooling jacket stream (denoted as $\hat{x}_1$ and $\hat{x}_2$) between 0 and 1 and generated the steady state values and this data was used for data fitting. The values of optimum temperatures can be scaled using relations

$$\hat{x}_1 = 64.53(6.1332 + x_1); \qquad \hat{x}_2 = 4.72816(64.669 + x_2);$$

This is plotted in Fig. 3.5. Here x_1 refers to steady state reactor temperature and x_2 refers to steady state cooling jacket temperature. The output from the sharp cut algorithm used for nonconvex programming is presented in Table 3.6. The multiple optimum solutions obtained by the sharp cut algorithm are given in Table 3.7.

3.7 Results and discussion

The application of the algorithm to a select few standard nonconvex functions shows that the algorithm reaches an accumulation point, and along the way has generated a sequence of solutions existing across a level hypersurface and are obtained in a monotone order. The Section 3.4 and accompanying Tables 3.1, 3.2, 3.3 and 3.4 clearly show how $g_r^{(l)} = \text{Eval}[g_r]$ attains a constant value and that it reaches the accumulation point. It is observed that the 2–tuple $(\forall l|(g_{r,l}^o, \hat{\phi}_0(\bar{\alpha}_l)))$ selects only a few of apex points seen in Figures 3.1, 3.2, 3.3, 3.4 that form a monotone sequence, when the sharp cut iterates. Thus, it showed the ability to capture all possible optimum solutions available at the level hypersurface defined by the accumulation point $(g_{r,\infty}^o)$ of Cauchy sequence $(\{\forall l \in \{0, 1, 2, \cdots, \infty\}|g_{r,l}^o = \hat{\phi}_0(\bar{\alpha}_l)\})$. It is observed that as multiple

solutions are obtained by algorithm, the first sharp cut and second sharp cut remain the same. This is the reason that in succeeding iterates for the evaluated numerical value of the most violated constraint, the algorithm keeps finding all possible optimum solutions. Thus as this evaluated value $g^o_{r,l}$ does not change an accumulation point is reached. It is also observed that the number of iterations range from zero to two before this accumulation point is reached (see Tables 3.1–3.4).

In the application example used for illustration, the nonlinear dynamics as depicted in the $3D$ plane ($C_{A,s}$, T_s, $T_{J,s}$) forms a rough terrain with several peaks across the $2D$ plane ($T_{J,s}$, T_s) and although $C_{A,s}$ remains a value indicating higher conversion for all such local optimum point, the reactor temperature vaies and hence the cooling water temperature varies in proportion. Our objective is to utilize less energy from utility providiong cooling water and hence it implies opt for lower reaction temperature lower output jacket temperature. This will ensure minimum exothermicity at that reaction temperature which in turn implies less amount of heat to be absorbed from reactor by coolant. Table 3.6 shows that the value of convergence parameter $g^o_{r,l}$ moves through a rough terrain to collect all optimum points. In this case the first and second sharp cuts remain invariant and collect all points which

are not in monotone order, but the accumulation point of Cauchy sequence formed as *weighted* $g^o_{r,l} = 474900.0$ gives a point sequence of convergence parameter $\{-52595.0, -94623.2, -140247.0, -81477.9\}$ (see Section 3.3 giving algorithm).

3.8 Conclusions

In this chapter, we have applied the sharp cut to various nonconvex functions and obtained multiple optimum solutions. This is a unique feature of the variant of Kelley's cutting plane algorithm. When it was applied to a highly nonlinear nonconvex function to obtain successive final states for optimal control of nonisothermal CSTR. It was found that the sharp cut could move across the uneven terrain and obtain all local optimum solutions.

3.9 References

Bertsekas D. and Nedic A., 2003, Convex Analysis and Optimization Athena Scientific.

Bonilla, J., Diehl, M., Logist, F., De Moor, B. , Van Impe, J., 2010, An automatic initialization procedure in parameter estimation problems with parameter-affine dynamic models, Computers & Chemical Engineering, **34**, (6), 953–964.

Boyd S., Vandenberghe L., 2004, Convex Optimization, Cambridge University Press.

Edgar, T.F. and Himmelablau, D.M. (1988), Optimization of Chemical Processes, McGraw-Hill Inc.

Floudas, C.A. 1999, Recent advances in global optimization for process synthesis, design and control: Enclosure of all solutions, Computers & Chemical Engineering, **23**, Supplement, S963–S973.

Floudas, C.A., Akrotirianakis, I.G., Caratzoulas, S., Meyer, C.A., Kallrath, J., 2005, Global optimization in the 21st century: Advances and challenges, Computers & Chemical Engineering, **29**, 1185–1202.

Georgiorgis D.I., Georgiadis M., Bowen G., Pantelides C.C., Pistikopoulos E.N., 2006, Dynamic oil and gas production optimization via explicit reservoir simulation, Computer Aided Chemical Engineering, **21**, 179–184.

Harjunkoski I., Jain V., and Grossman, I.E., 2000, Hybrid mixed-integer/constraint logic programming strategies for solving scheduling and combinatorial optimization problems, Computers & Chemical Engineering, **24**, 337–343.

Inamdar, S.R., Karimi I.A., Parulekar S.J., Kulkarni B.D., (2011), Computers & Chemical Engineering **35**, 2716-2728.

Mangaserian, O.L., 1969, Nonlinear Programming, McGraw-Hill Company, NY.

Ravindran,A., Ragsdell, K. M., Reklaitis G. V., 1983, Engineering Optimization, I Edn., Wiley-Interscience.

Ren H., and Argyros I.K., 2010, Convergence radius of the modified Newton method for multiple zeros under Hlder continuous derivative, Applied Mathematics and Computation, **217**, (2), 612–621.

Srinivasan, B., Biegler, L.T., Bonvin, D., 2008, Tracking the necessary conditions of optimality with changing set of active constraints using a barrier-penalty function, Computers & Chemical Engineering, **32**, 572–579.

Sun W., Liu, Q. and Wang C., 2009, A homotopy method for getting a local minimum of constrained nonconvex programming, Nonlinear Analysis: Theory, Methods & Applications, **71**, (10), 4725–4731.

Varma V. A., Pekny J.F., Blau G. E., Reklaitis G. V., 2008, A framework for addressing stochastic and combinatorial aspects of scheduling and resource allocation in pharmaceutical R&D pipelines, Computers & Chemical Engineering, **32**, 1000–1015.

Xu X., Li C., 2008, Convergence criterion of Newton's method for singular systems with constant rank derivatives, Journal of Mathematical Analysis and Applications, **345**, (2), 689–701.

Figures

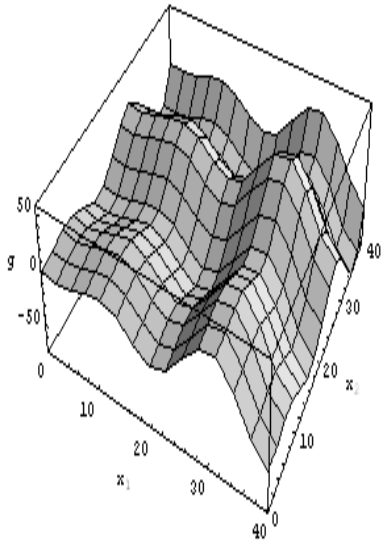

Figure 3.1: The Ackley's function.

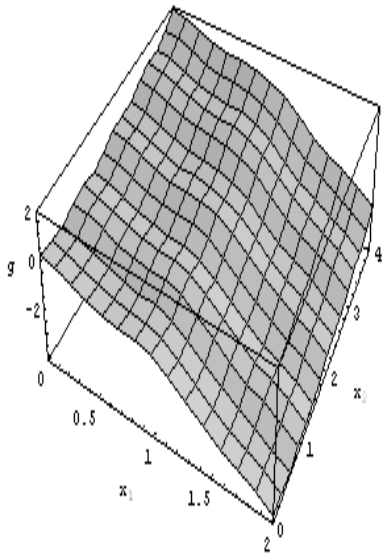

Figure 3.2: The Rastrigin's function.

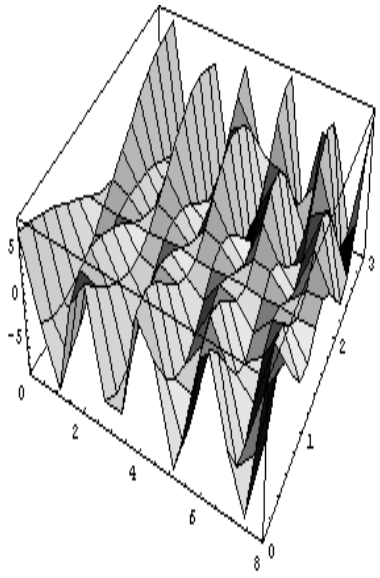

Figure 3.3: The Schwefel's function.

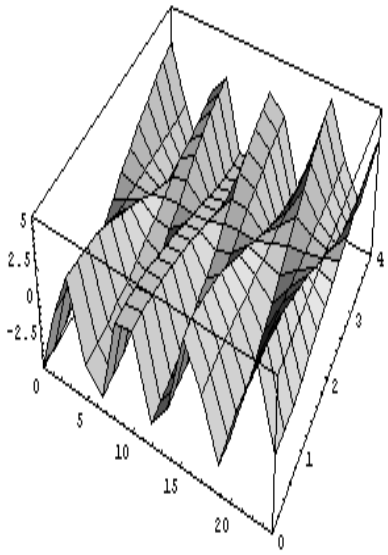

Figure 3.4: The Griewangk's function.

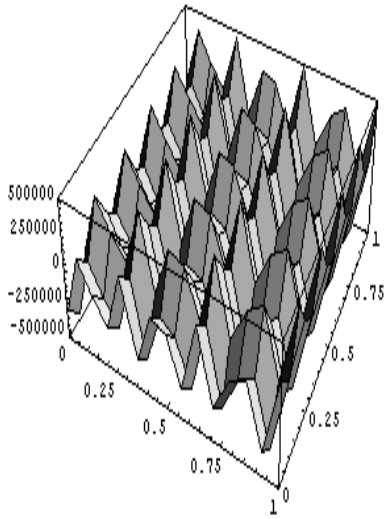

Figure 3.5: Nonconvex function by fitting steady state values for nonisothermal CSTR.

Tables

Table 3.1: Output of sharp cut algorithm for Ackleys function in 2D as a nonconvex constraint.

Iter	Eval$[g_r]$	$\hat{\phi}_{0,l}(\mathbf{x})$	$\alpha_{1,l}$	$\alpha_{2,l}$
1	-0.669324	-11.6976	3.8488	4.0
2	-0.175273	-11.5966	3.7983	4.0
3	-0.0595865	-11.5399	3.76993	4.0
4	0.0595865	-11.4808	3.78408	4.0
5	-0.0595865	-11.4808	3.78408	4.0
6	0.0595865	-11.4777	3.78482	3.90802
7	-0.0595865	-11.4777	3.78482	3.90802
8	0.0595865	-11.4747	3.78553	3.90363
9	-0.0595865	-11.4747	3.78553	3.90363
10	0.0595865	-11.4718	3.78621	3.8994
11	-0.0595865	-11.4718	3.78621	3.8994

12	0.0595865	-11.4691	3.78687	3.89531
13	0.0595865	-11.4664	3.78752	3.89135
14	-0.0595865	-11.4664	3.78752	3.89135
15	0.0595865	-11.4638	3.78814	3.88749
16	-0.0595865	-11.4638	3.78814	3.88749
17	-0.0595865	-11.4638	3.78814	3.88749
18	0.0595865	-11.4612	3.78875	3.88374

*After 18^{th} iteration algorithm terminated as LP solution could not be found.

Table 3.2: Output of sharp cut algorithm for Rastrigin's function in 2D as a non-convex constraint.

Iter	Eval$[g_r]$	$\hat{\phi}_{0,l}(\mathbf{x})$	$\alpha_{1,l}$	$\alpha_{2,l}$
1	0.211828	-5.31487	4.0	1.31487
2	-0.211828	-5.31487	4.0	1.31487
3	0.211828	-5.25888	4.0	1.25888
4	-0.211828	-5.25888	4.0	1.25888
5	0.211828	-5.20273	4.0	1.20273
6	-0.211828	-5.20273	4.0	1.20273
7	0.211828	-5.014401	4.0	1.14401
8	-0.211828	-5.014401	4.0	1.14401
9	0.211828	-5.07935	4.0	1.07935
10	-0.211828	-5.07935	4.0	1.07935
11	0.211828	-5.00259	4.0	1.00259
12	-0.211828	-5.00259	4.0	1.00259

13	0.211828	-4.89801	4.0	0.898015
14	-0.211828	-4.89801	4.0	0.898015
15	0.211828	-4.69257	4.0	0.695265
16	-0.211828	-4.69257	4.0	0.695265

*After 16^{th} iteration algorithm terminated as LP solution could not be found.

Table 3.3: Output of sharp cut algorithm for Schwefels function in 2D as a non-convex constraint.

Iter	Eval$[g_r]$	$\hat{\phi}_{0,l}(\mathbf{x})$	$\alpha_{1,l}$	$\alpha_{2,l}$
1	1.78907	-1.01577	0.292114	4.0
2	-1.78907	-1.01577	0.292114	4.0
3	1.78907	-0.777757	0.348952	3.689154
4	-1.78907	-0.777757	0.348952	3.689154
5	1.78907	-0.609136	0.495432	4.0
6	-1.78907	-0.609136	0.495432	4.0
7	1.78907	-0.586843	0.453901	3.73661
8	-1.78907	-0.586843	0.453901	3.73661
9	1.78907	-0.429321	0.466212	3.40436
10	-1.78907	-0.429321	0.466212	3.40436
11	1.78907	-0.135724	0.565017	3.16439
12	-1.78907	-0.135724	0.565017	3.16439
13	1.78907	0.772647	0.870713	2.42195
14	-1.78907	0.772647	0.870713	2.42195

*After 14^{th} iteration algorithm terminated as LP solution could not be found.

Table 3.4: Output of sharp cut algorithm for Griewangks function in 2D as a non-convex constraint.

Iter	Eval$[g_r]$	$\hat{\phi}_{0,l}(\mathbf{x})$	$\alpha_{1,l}$	$\alpha_{2,l}$
1	-1.51111	7.07726	4.33863	4.0
2	0.0808988	7.98746	4.79373	4.0
3	-0.0808988	7.98746	4.79373	4.0
4	0.0808988	8.03295	4.81648	4.0
5	-0.0808988	8.03295	4.81648	4.0
6	0.0808988	8.07854	4.83927	4.0
7	-0.0808988	8.07854	4.83927	4.0
8	0.0808988	8.12424	4.86212	4.0
9	-0.0808988	8.12424	4.86212	4.0
10	0.0808988	8.17009	4.88504	4.0
11	-0.0808988	8.17009	4.88504	4.0
12	0.0808988	8.2161	4.90805	4.0
13	-0.0808988	8.2161	4.90805	4.0
14	0.0808988	8.2623	4.93115	4.0

15	-0.0808988	8.2623	4.93115	4.0
16	0.0808988	8.30872	4.95436	4.0
17	-0.0808988	8.30872	4.95436	4.0
18	0.0808988	8.35539	4.97769	4.0
19	-0.0808988	8.35539	4.97769	4.0
20	0.0808988	8.40233	5.00116	4.0
21	-0.0808988	8.40233	5.00116	4.0

*After 21^{st} iteration algorithm terminated as LP solution could not be found.

Table 3.5: Parameter values used to compute steady states.

$R = 8.33\ kJ/kmol(K)$	$F_0 = 0.4\ m^3/hr$
$E = 13{,}900.0\ kJ/kmol$	$k_0 = 7.0 \times 10^4\ hr^{-1}$
$(-\Delta H_R) = 25{,}900.0\ kJ/kmol$	$C_{A,0} = 2.06\ kmol/m^3$
$V = 2.0\ m^3$	$V_{\min} = 0.7V,\ m^3$
$\rho_R = 600.0\ kmol/m^3$	$c_p = 2.5\ kJ/kmol(K)$
$T_0 = 410\ (K)$	$U = 8000.0\ kJ/m^2 \cdot hr \cdot K$
$A_H = 0.02\ m^2$	$\rho_J = 700.0\ kmol/m^3$
$V_J = 1.4\ m^3$	$V_{J,\min} = 0.7\ \ V_J$
$c_J = 3.0\ kJ/kmol(K)$	$F_{J,0} = F_0\ \ \eta$
$T_{J,0} = 300.0\ (K)$	

Table 3.6: Performance of the cutting plane variant for CSTR example.

Iteration l	Number of data	Convergence parameter $g^o_{r,l}$	Weighted max value of $g^o_{r,l}$	$\phi_0(\bar{\alpha}_l)$
1	10	-52595.	474900	-0.981331
2	10	-94623.2	474900	-0.980581
3	10	-140247.	474900.	-0.923408
4	10	-81477.9	474900.	-0.989544
5	10	-81477.9	474900.	-0.989544

Table 3.7: Multiple optimum solutions obtained by the cutting plane variant for CSTR example.

Iteration l	$\phi_0(\bar{\alpha}_l)$	x_1^*	x_2^*
1	-0.981331	0.00133106	0.98
2	-0.980581	0.000580935	0.98
3	-0.923408	0.00383961	0.919569
4	-0.989544	0.0000700541	0.989474

Chapter 4

Optimal control and optimization of a nonisothermal CSTR. Application of sharp cut algorithm

Abstract

The authors presented a new variant of the Kelley's cutting plane method, which they termed as a sharp cut algorithm. This algorithm was found to be superior than its predecessors. Recent numerical computations revealed that the algorithm collects all local optimum solutions in a monotone order. Here, we apply this algorithm to solve a control and optimization problem for a nonisothermal CSTR.

4.1 Introduction

The topic of control of a chemical reactor having nonlinear dynamics still remains open for further research and innovation. In this chapter, we focus upon the problem of control system design for a nonisothermal continuously stirred tank reactor (CSTR).

4.1.1 Motivation for this work

To find a control strategy for a nonisothermal CSTR, we observed from computed results that most of the steady states were unstable for select parameter set. Hence, we opted to apply an optimal controller to move from one set point to another in succession in a small local neighborhood. At the end of iterations of optimal control, we consider a scenario that parameter shifts occur. To begin next move of optimal controller, we need to identify admissible steady states in the neighborhood of this final state. As multiple steady states are possible, we define a linear objective function, fit steady states as a nonconvex function of two parameters, viz, feed temperature and feed or coolant flow rate. Now we apply the sharp cut algorithm to find multiple optimum solutions, select a new set point, re-set the plant transfer function and controller, and again apply the optimal controller. Further,

when the plant experiences a stochastic variation in inlet parameters, we use feedback control for level, add extra internal cooling coils, and, derive a stochastic optimal control to save costs.

4.1.2 Identification of problem

The idea of designing a neighborhood of local optimum solutions for design and control of a CSTR stems from a revelation. The authors presented a new variant of the cutting plane algorithm, a sharp cut algorithm (Inamdar et. al., 2012), that has a high rate of convergence. The variant of Kelleys cutting planes, as the best option amongst commonly used cutting plane methods, and suggested to explore if it works for a well-defined problem of nonconvex program. The new variant showed a unique ability to capture all possible optimum solutions at an accumulation point of Cauchy sequence. Next, we present the sharp cut algorithm and later show how it is applicable to the control problem.

4.2 The sharp cut algorithm

In this section, we present the sharp cut algorithm to obtain local optimum solutions of a nonconvex program. Later it will be applied to an application example.

4.2.1 The algorithm for nonconvex programming

1. **Initialization.** We begin with a linear objective function $\phi_0(x)$. This is a limitation of this algorithm. We write the box constraint

$$Z^0 = \{\forall i | a_i \leq x_i \leq b_i | i = 1, 2, , n\}.$$

The call to LP solver for zeroth iteration gives an optimum solution $\bar{\alpha}_0^*$.

2. **First iteration.** Now, we set SLP iteration index $l = 1$ and compute the function value $g_(r, 1)^0$. (Here r^{th} constraint is the most violated constraint (MVC)). To descend steeply upon the hypersurface of MVC, we first generate a number of vector point for $l = 1$ and the point set Φ defined as

$$\Phi = \{\forall \wp \in \{0, 1, 2, \cdots, \aleph_{\pm -1}\} | \bar{\nu}_\wp\}$$

and

$$\bar{\nu}_{\wp\pm} = \{\forall i = 1, \cdots, n | \wp\pm = 1, \cdots, \aleph_{\pm} | \alpha_{i,l} \pm \wp\pm \delta_{\pm,i}\}$$

where

$$\{\forall i | \delta_(i,-) = (\alpha_(i,l) - a_i)/(\aleph_- - 1), \qquad \forall i = 1, \cdots, n | a_i \leq \bar{\nu}_{i,\wp-} \leq b_i\},$$

and

$$\forall i | \delta_{i,+} = (b_i - \alpha_{i,l})/(\aleph_+ - 1), \quad \forall i = 1, \cdots, n | a_i \leq \bar{\nu}_(i, \wp+ \leq b_i.$$

Then, the point set E_0 is generated as

$$E_0 = \left\{\forall \wp\pm \in \{0, 1, 2, \cdots, \aleph_{\pm} - 1\} | \bar{\nu}_{\wp\pm}\right\}$$

and

$$0 < \varsigma_{r,\wp\pm}(\bar{\nu}_{\wp\pm}) \leq g^o_{r,l}, \quad \varsigma_{r,\wp\pm}(\bar{\nu}_{\wp\pm}) = \phi_r(\bar{\nu}_{\wp\pm}) \leq g^o_{r,l}.$$

Now, the new vector point $\psi_(r,l)^o$ can be generated as

$$\psi^o_{r,l} = \{r | \sigma \in \{\wp\pm\}; \bar{\nu}_{\wp\pm} \in E_0 | \varsigma_{r,\sigma}(\bar{\nu}_\sigma) | Min \left\{\hat{\phi}_0(\bar{\nu}_{\wp\pm})\right\}\right\}$$

where, $\wp\pm = \{0, 1, \cdots, \aleph_{\pm-1}\}$. The new vector point obtained is denoted as, $\bar{\beta}_l = \bar{\nu}_\sigma$.

3. **First sharp cut.**

$$v'_l(x; \bar{\beta}_l) = w_{1,l}\psi^o_{r,l} + \sum_{i=1}^{n} \nabla_{x_i}\phi_r(\bar{\beta}_l)(x_i - \beta_{i,l}) \geq 0;$$

the vector point $\bar{\beta}_l$ is scaled using weights.

4. **Second sharp cut.** Using a relation

$$\{r|\sigma \in \{\wp_{\pm}\}|\bar{\nu}_{\sigma}|\{\varsigma_(r,\sigma)(\bar{\nu}_{\sigma}) \leq \delta \times \psi^{o}_{r,l}\} \subset E_0\},$$

a linear multi-variate fit gives a gradient set, $\{\forall i|1 \leq i \leq n|\rho_i\}$, and, the second sharp cut is obtained as

$$w_l(x;\bar{\beta}_l) = w_{2,l}\psi^{o}_{r,l} + \sum_{i=1}^{n} \rho_l(x_i - \beta_{i,l}) \geq 0;$$

and $w_{2,l}$ is a random number used as the weight.

5. **Formation of constraint set.** Now to descend steeply upon the MVC to form the next hypercube as

$$Z^1 = Z^0 \cap H^1, \qquad \text{where,} \qquad H^1 = \{l = 1|u_l \geq 0 \cap v'_l \geq 0 \cap w_l \geq 0\}.$$

The call to LP solver gives us an optimum solution, $\bar{\alpha}^{*}_1$.

6. **The l^{th} SLP iteration.** The statement of optimization program for SLP iteration becomes,

$$Min\bar{\phi}_0(x) = c^t x; \text{s.t. } \; x \in Z^{l-1} \cap H^l.$$

This gives an optimum solution in Cauchy sequence, $\bar{\alpha}_l$.

7. **Termination criteria.** If $g_(r,l)^0$ becomes zero, as $l \to \infty$, (an idealistic condition that can occur), or an accumulation point is reached for the Cauchy sequence, the algorithm terminates.

4.3 Optimal control

Let us continue with an example in chemical reactor theory. A continuously stirred tank reactor (CSTR) is carrying out a first order exothermic reaction, $A \to R$. The reactor temperature is controlled by employing a cooling jacket. The reaction rate form is of the Arrhenius type with the rate constant k being, $k = k_0 exp(-E/RT)$. The governing model equations are

$$\begin{aligned}
\frac{d(VC_A)}{dt} &= F_0 C_{A0} - FC_A - Vk_0 exp(-E/RT)C_A \\
\rho c_p \frac{d(VT)}{dt} &= \rho c_p (F_0 T_0 - FT) \\
&+ (-\Delta H_R) Vk_0 exp(-E/RT)C_A - UA_H(T - T_J) \\
\rho_J c_J V_J \frac{dT_J}{dt} &= F_J \rho_J c_J (T_{J0} - T_J) + UA_H(T - T_J)
\end{aligned}$$

The controller, $F = F_0 + K_C(V - V_{\min})$ is added to have a zero degree of freedom. Note that multiple optimum solutions for reactor appear, as small shifts in input flow rates to reactor and cooling jacket (i.e. F_0 and F_J) occur. This results in an optimization step at end of optimal control move.

Defining a deviation $\mathbf{X} = \mathbf{X}_s + \mathbf{x}$, we find that, linearizing around a steady state X_s, and, neglecting higher order terms, we obtain a state space system, $\mathbf{Mx} = \mathbf{Ax} + \mathbf{Bu}$. The operation of control system is devised to move from initial state to final state by applying an optimal control and repeat control move as reactor steady states are unstable. This sequential computation of optimal control step and then a subsequent optimization step by applying sharp cut to seek available locally optimum states gives rise to a control and optimization problem. Another reason for optimization step at end of optimal control move is shifts experienced by flow rates of feed and coolant from their initial values.

Now, the objective function for optimal control is

$$J(X(t), u(t), t) = \frac{((1 - F(t)T(t))}{(F_0T_0)^2} + \frac{(\rho_J c_J F_J)}{(\rho c_p F_0 T_0)}(T(t) - T_J(t))^2$$

The Hamiltonian function is

$$\begin{aligned} H(\mathbf{X}(t), \mathbf{u}(t), t) &= \frac{(1 - F(t)T(t))}{(F_0T_0))^2} + \frac{(\rho_J c_J F_J)}{(\rho c_p F_0 T_0)}(T(t) - T_J(t))^2 \\ &+ \bar{\lambda}(\mathbf{Mx} - (\mathbf{Ax} + \mathbf{Bu})) \end{aligned}$$

In the next section, we show how in real systems this leads to a derivation of Hamilton-Jacobi-Bellman (HJB) equation for stochastic optimal control.

4.4 Stochastic optimal control

In this section, we consider a small amplitude variation in the feed concentration being pumped into the CSTR. This variation causes equivalent variation in heat being added to the CSTR contents. Assuming that basic instrumentation is used to measure temperature of reaction mixture, feed flow rate, and liquid level in the reactor; and the effluent flow rate leaving CSTR, deviations from set point or steady state satisfy the relations, defining

$$\varphi_1 \equiv \frac{(2exp(-E/(RT_s))\alpha H_0 A_R)}{(RT_s^2)}$$
$$\varphi_2 \equiv \frac{(2exp(-E/(RT_s))\alpha(-\Delta H_R)H_0 A_R)}{(RT_s^2)}$$

we can write relations

$$F_0(C_{As} - C_{A0}) + F_0(\Delta C_{As} - \Delta C_{A0}) + \varphi_1(E\Delta T_s C_{As} + RT_s^2 \Delta C_{As}) = 0 \quad (A)$$

$$F_0(-T_0 + T_s) + (UA_H + F_0)\Delta T_s + \varphi_2(E\Delta T_s C_{As} + RT_s^2 \Delta C_{As}) = 0 \quad (B)$$

Here the cooling jacket heat load reduces to a simple term, $UA_H\Delta T_s$. This will be excess heat removed by internal cooling coils. Now, we consider the variation caused by spikes, and apply the stochastic optimal control to make additional profits. The modified steady state equation for internal cooling coils (ICC) is written as, $\Delta Q_{c,\text{icc}} = c_p \rho_R F_0 \Delta T_{\text{icc}}$. The ICC dynamics

for removal of heat in the presence of fluctuations is

$$\ddot{X} = \hat{f}(X(t), \varsigma(t)) + \xi(t); \quad t \leq t \leq T; \quad X(\tau) = x^0$$

The control $\varsigma(\tau)$ depends upon ? and sampled data $X(\tau)$ and $\xi(\tau)$ is Gaussian white noise. The fluctuations drive the system to a new final state as per the relation

$$dX(t) = (1 - \alpha^1(t))X(t)rdt + \alpha^1(t)X(t)(Rdt + \sigma dW) - \alpha^2(t)dt$$

The pay-off functional $P_{x,t}|\varsigma()$ as a probability density function to be minimized and is given as, $P_(x,t)|\varsigma(\cdot) = E(\int_t^\tau r(X(\tau), \xi(\tau))d\tau + g(X(\tau))$; where r is generic running cost function and g is terminal cost function. Let $W = \xi$ and we set, $X(t) = x^0 + \int_0^\tau \hat{f}(X(\tau))d\tau + \sigma dW(\tau)$; and then to design a stochastic optimal control, we write, $dX(\tau) = \hat{f}(X(\tau), \varsigma(\tau)) + \sigma dW(\tau)$. Define a value function, $v(x,t) = \underset{\varsigma(\tau)}{Max} \in \Xi P_(x,t)[\varsigma(\cdot)]$, now apply a stochastic optimal control and obtain the Hamilton–Jacobi–Bellman (HJB) equation, which solves for the value function (v) as

$$v_t(x,t) + \frac{\sigma^2}{2}\Delta v(x,t) + \begin{pmatrix} Max \\ \varsigma \in \Xi \end{pmatrix} \left\{ \hat{f}(x,a) \cdot \nabla_x v(x,t) + r(x,a) \right\}; \quad v(x,T) = g(x)$$

and then stochastic optimal control is designed as

$$dX^*(\tau) = \hat{f}(X^*(\tau), \varsigma(X^*(\tau), \tau), \tau)d\tau + \sigma dW(\tau); X^*(\tau) = x,$$

and then, $\xi^*(\tau) = \varsigma(X^*(\tau), \tau)$ is an optimal feedback control. The bias term is calculated using Eqs. (A) and (B). Now, we carry out numerical computations.

4.5 Numerical computations

We study the application of optimal control and later that of stochastic optimal control.

4.5.1 Steady state analysis

In this section, we compute the multiple steady states (Inamdar et.al. 2012a). For $F_0 = 0.39(m^3/hr)$, and, $F_J = 3.5F_0(m^3/hr)$, the steady state conditions are, $C_{As} = 0.0001655(kg/m^3)$, $T_s = 437.151(K)$, $T_{Js} = 307.251(K)$. There is near complete conversion in reactor at steady state ($x_A = 0.999834$). To apply the optimal control, we need to compute next final state to move to. Therefore, we vary flow rates in the intervals $F_0 \in [0.25, 0.45]$, $F_J \in [2.2, 4.4]$. The data is fitted into Ackley's 2D function (see Fig. 1). The sharp cut is applied to minimization of sum of scaled temperatures of reaction and coolant. For same values of $g^o_{r,l}$ and first and second sharp cuts at

accumulation point, solutions show varied depths (see Ref. Inamdar et. al. 2012a).

4.5.2 Optimal control and optimization.

For a selected unstable reactor state, we begin our moves between initial and final states (see Ref. Inamdar et. al. 2012b). The results of optimization steps before selecting next final state and loci of optimal control paths can be computed using standard theory. Now, we consider stochastic variation of feed reactant concentration.

4.5.3 Stochastic optimal control

Let the feed concentration C_{A0} vary as a small amplitude sinusoidal oscillation to which we add a stochastic or random disturbance. Let the parameter values be, $\hat{a}$=0.2, $\omega_e = 1.92$, R=1.2, $r = 1.08$, $\hat{\rho} = 0.4$, $\gamma = 0.06$, $\hat{\sigma} = 0.5$, $T = 2.0$. We use a relation $X(t) = \hat{a}\sin(\omega_e t) + \xi(t)$, where $\xi(t)$ is a white Gaussian noise and $0 \leq \xi(t) \leq 1$. The plot of $\alpha_2(t)$ versus t (see Fig. 2) shows a logarithmic trend and the output response (not shown) gradually approaches the final state. Further details will be given provided in a manuscript under preparation (Inamdar et. al., 2012b).

4.6 Conclusions

In this chapter, we presented an optimal control and optimization of a non-isothermal CSTR. As the feed concentration varies stochastically, a stochastic optimal control using extra internal coils is applied. Numerical computations illustrate the findings well.

4.7 References

Akesson, J. Introduction to Optimal control theory,

See URL: https://www.control.lth.se/Staff/JohanAkesson.html.

Evans, L.C., (1983), An Introduction to Mathematical optimal control theory, Version 2.0, URL: http://math.berkeley.edu/evans/control.course.pdf.

Inamdar, S.R., Karimi, I.A., Parulekar, S.J. and Kulkarni, B.D., (2011), A sharp cut algorithm for optimization, Comput. & Chem. Engg., 35(12), 2716–2728.

Inamdar, S.R., Karimi, I.A., Parulekar, S.J. and Kulkarni, B.D., (2012a), Solving a nonconvex program using a sharp cut algorithm, (To be communicated to Comput. & Chem. Engg).

Inamdar, S.R., Karimi, I.A., Parulekar, S.J. and Kulkarni, B.D., (2012b), Optimal control and optimization of a reactor-separator system, (To be communicated to Chem. Engng. Sci.).

Uppal, A., Ray W.H., Poore A.B., (1974), On the dynamics of continuously stirred tank reactors, Chemical Engineering Science, 29, 967–985.

Note.

The paper (from which this chapter is written) was accepted as a full oral presentation for Eleventh International Symposia on Process Systems Engineering, 2012 being hosted by National University of Singapore. The author could not attend for some reason.

Figures

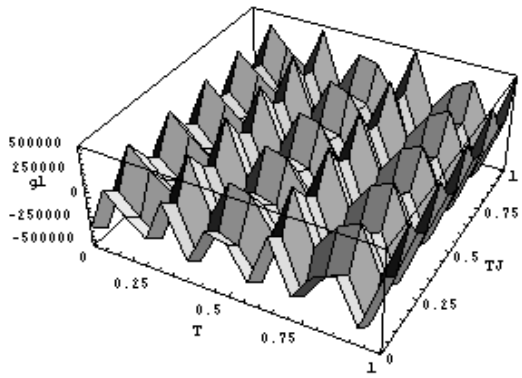

Figure 4.1: Forming perturbed scaled nonconvex function for CSTR optimization

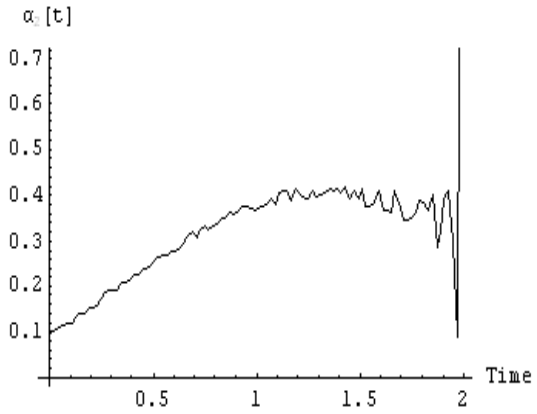

Figure 4.2: The randomly varying cost component for stochastic optimal control.

Chapter 5

Stochastic optimal control of a nonisothermal CSTR

Abstract

In previous chapter, we have seen that the multiple optimum solutions can be obtained using the sharp cut algorithm and an optimal controller can be designed effectively. We assumed in this derivation of results that the feed concentration is not experiencing any variation and remains constant.

As an extension of previous result, when analysis of such variation leads us to a stochastic differential equation and finally we can compute qualitatively expressions in standard literature to illustrate the application of the sharp cut algorithm for a single iteration. In this chapter, we have applied the theory of stochastic optimal control to make use of optimum time–continuous optimum ratio of two process water utilities i.e. cold water and brine water to mix in proportion in order to solve the problem of reactor control and operation. Here, the reactor is carrying out a first order exothermic reaction and the feed concentration is varying sinusoidally with a bias added. The steady states computed in a search range of reactor dynamics are unstable and an optimal control is selected to control the reactor. Further, when the feed concentration varies sinusoidally with time during the downhill phase the temperature of reaction falls below the optimum which is also the set point. Hence, the conversion in some range of parameters may fall below

specified minimum limit. During this interval we cause the reactor output to be by–passed. This by–passed process stream defines a loss function for stochastic optimal control as it is further processed in a batch reactor in lots and added to the main stream later. This step minimizes losses of raw material, but extra batch processing cost takes away part of profits. As the feed concentration varies sinusoidally and the bias experiences a random variation w.r.t. time, we apply the method of stochastic optimal control and derive necessary analytical results.

5.1 Introduction

The example of a first order exothermic reaction is very common in chemical reaction engineering literature. For the theoretical work, the most cited references are that of Uppal, Poore and Ray (1974, 1976). Another seminal paper is by Farr and Aris (1986) and discusses the bifurcations that occur in the dynamics of reacting system. It is then evident that in operating such a chemical reactor carrying out an exothermic reaction, it is necessary to apply control to keep the temperature at an optimum point by absorbing heat released. The numerical computations show that the steady states computed in a search range of reactor dynamics are unstable and an optimal control (Anderson and Moore, 1990; Hull, 2003) is selected to control the reactor. However, a stochastic disturbance enters along with sinusoidal variation implying a stochastic optimal controller needs to be designed. The external cooling jacket assumes no variation in feed concentration and rejects any weak disturbance entering the reactor. The variation from this design value can cause excess heat to be released. Here to make stochastic optimal control more effective we opt for two types of utility resources. For sinusoidal variation we use process water at ambient temperature as a coolant stream and for randomly varying disturbance around a bias we use brine solution

which has higher capacity to absorb heat quickly. Later an eductor can be used to mix the two streams fully and sent through the internal cooling coils. The addition of extra internal cooling coils can be an option that may be considered for design and simulation exercise and a properly designed controller brings the temperature of reaction back to the optimum value in a short time.

Our primary objective is to contain the movements of this optimal control within a small neighborhood of limited search range of inlet parameters (Inamdar et. al., 2011, 2012). However, the sinusoidal variation in feed concentration and random variation of the bias implies that a stochastic optimal control should be applied. After reaching the new initial state after a duration of stochastic optimal control, the new average bias value necessitates computation of multiple optimum solutions possible. This computation sets a relation between C_{As} as a function of T_s and T_{Js}. The numerical computations show that the conversion within this range indicates it tends to unity. Clearly, any effort to reduce the temperature of reaction and with a relation to temperature of cooling jacket means effective utility savings. This motivates us to take up this problem. First we give the model of nonisothermal CSTR and then see how this leads us to the problem of stochastic optimal

control.

Now, we derive the equations describing the process dynamics of CSTR. The reactor carries out a first order exothermic reaction A $\rightarrow$ R and the reaction is of first order and the CSTR is maintained at a favorable temperature by employing a cooling jacket. The reaction rate form is of Arrhenius type and the rate constant for step A $\rightarrow$ R is $k(T) = k_0 exp\left[\frac{-E}{RT}\right]$. The governing equations are given here.

$$V\frac{dC_A}{dt} = F_0C_{A,0} - FC_A - Vk_0 exp\left[\frac{-E}{RT}\right]C_A \tag{5.1}$$

$$\begin{aligned}\rho c_p V\frac{dT}{dt} &= \rho c_p(F_0T_0 - FT) \\ &+ (-\Delta H)Vk_0 exp\left[\frac{-E}{RT}\right]C_A - UA_H(T - T_J)\end{aligned} \tag{5.2}$$

$$\rho_J c_J V_J\frac{dT_J}{dt} = F_J\rho_J c_J(T_{J,0} - T_J) + UA_H(T - T_J) \tag{5.3}$$

Now we will see how multiple optimum solutions for nonisothermal CSTR can be computed, when small shifts in input flow rates to reactor and cooling jacket (i.e. F_0 and F_J) occur, and result in a control and optimization solution. Note that the dynamics of cooling jacket is given above and the temperature rise can be controlled by varying the coolant flow rate.

As is clear the effect of variation in feed concentration can only be re-

alized when we measure the output feedback (controlled) variable (reaction temperature in this case). However in order to decompose the output value into constituents, we need to know the bias and small amplitude variation in feed concentration. There is a small random disturbance which gets added to this input term. This stochasticity can be measured by measuring concentration as sampled data that can be recorded and this data can be sent to computer to analyze. In order to achieve this a small cell parallel to input pipe averages the spiky variation in feed concentration and we can collect samples. These samples are analyzed to measure the bias and properties of oscillatory input (amplitude and period of oscillation). This arrangement can be helpful in constructing models that when subtracted from output value gives an idea about impulses being added to flow input. Such an arrangement is necessary to implement the optimal control and later is useful in applying the stochastic optimal control. To conclude this discussion, we find that the application of external cooling jacket assumes that feed concentration C_{A0} is constant and rejects any weak disturbance entering the dynamics. The extra cooling capacity added by internal cooling coils subtracts the feedback output response calculated by first model i.e. no sinusoidal variation and bias caused in feed concentration value. The remaining variation is then assigned

to internal cooling coils. Assuming there is no significant loss in level control due to coupling with the main control loop, the stochastic optimal control then can be applied to a scalar equation. We explain this in next section.

5.2 Stochastic optimal control

Since there is small amplitude sinusoidal variation in feed concentration, the reaction temperature in the chemical reactor will experience oscillations. This means the cooling duty will vary with time for the internal cooling coils. Along the rise of temperature value extra cooling will be required, while during the downfall the temperature of reaction mixture will fall below the optimum value which is the set point. We stop the action of the internal cooling coils during this time interval. The result is that the conversion level will too fall below expected value and this "sloppy cut" in this time interval (until temperature goes up above set point corresponding to optimum conversion). This sloppy cut is by–passed to a storage tank and processed using a batch reactor. This less reacted mixture after batch processing is added later to the the downstream sequence. The batch processing operation will incur extra cost.

Let us begin the analysis and design of stochastic optimal control system. We note that, there is a bias that gets added and a small amplitude oscillation as variation in feed concentration. Hence the heat released due to exothermicity of reaction will eventually cause the temperature of reaction mixture to oscillate. We will explain this in detail here.

The variation in feed concentration occurs as

$$C_{A,0}(t) = \bar{C}_{A,0} + \Delta C_{A,0} + \tilde{a}\sin(\omega_e t) \tag{5.4}$$

Let us write the expression for feed concentration varying with time.

$$\begin{aligned} C_{A,0}(t) &= C_{A,0,1}(t) + C_{A,0,2}(t) \\ &= (\Delta C_{A,0} + \xi(t)) + (\tilde{a}\sin(\omega_e t)) \end{aligned} \tag{5.5}$$

Here $\bar{C}_{A,0}$ is the feed concentration as a design value. We assume that the external cooling jacket is designed to absorb all excess heat due to exothermic reaction with $C_{A,0}(t) = \bar{C}_{A,0}$. Any weak disturbance entering the dynamics will be rejected by a feedback control loop. Also, an optimal control can be designed as we know that all steady state in search parameter are unstable as explained before.

Let us now consider the parametric disturbance term as $C_{A,0,1}(t)+C_{A,0,2}(t)$

as in Eq. (5.5). Clearly, we need to a priori determine, for this experimental apparatus set up, the unknowns, $\Delta\bar{C}_{A,0}$ (a bias), $\xi(t)$ (a random disturbance occurring around this mean value), and variables in oscillatory disturbance term i.e. $\tilde{a}$ (amplitude) and ω_e (frequency of external periodic forcing term). For this purpose, the sampling done via a cell, parallel to the feed pipeline is devised. The sampled data collected over a short interval of time can be used to estimate the unknowns ($\Delta\bar{C}_{A,0}$, $\tilde{a}$, ω_e). These constants can be used to design a stochastic optimal controller.

Now, we will analyze the control problem. The parametric disturbance in Eq. (5.5) is releasing extra heat and will cause further variation in reaction temperature. Here we have estimated the parameters i.e. $\bar{C}_{A,0}$, $\tilde{a}$, ω_e. Hence, we can decompose the parametric disturbance into known and randomly varying components. This is clearly depicted as a block diagram in Fig. 2. Thus as a known disturbance $\bar{C}_{A,0}$ is experiencing a small amplitude variation. And as a randomly varying component $\bar{C}_{A,0}$ is experiencing a bias and stochastic variation. The linearization of process dynamics has the state variables $C_A(t)$, $T(t)$ and $T_J(t)$ i.e. concentration of species A in reaction mixture, the temperature of reaction mixture, and the temperature of coolant in internal cooling coils, respectively.

Therefore, the temperature rise in steady state (T_s) is given as

$$T'(t) = T'_1(t) + T'_2(t) \tag{5.6}$$

where $T'_1(t)$ and $T'_2(t)$ are deviations caused by parametric disturbance corresponding to variations given as $C_{A,0,1}(t) + C_{A,0,2}(t)$. The controller pair is used for this single–input single–output (SISO) control loop is reaction temperature ($T(t)$) as a controlled (measurable) variable and $q_c(t)$ i.e. coolant flow rate for internal cooling coils (ICC) as a manipulated variable. Hence, corresponding to the temperature deviations, $T'_1(t)$ and $T'_2(t)$, we need to calculate coolant actions, which are $\tilde{q}_c(t)$ and $\bar{q}_c(t)$, respectively.

In order to obtain analytical results, we take Laplace transform of the three linearized equations, and, derive expressions for control ratios relating $C_{A,0,1}(t)$ and $T'_1(t)$, $C_{A,0,2}(t)$ and $T'_2(t)$. Later, we can find control ratio expressions for $T'_1(t)$ and $\tilde{q}_c(t)$, $T'_2(t)$ and $\bar{q}_c(t)$. For multiple inputs, the transfer function relations can be derived and we write derived control ratios as

$$\tilde{q}_c(\jmath\omega) = G_{11}(\jmath\omega)T'_1(\jmath\omega); \qquad \bar{q}_c(\jmath\omega) = G_{21}(\jmath\omega)T'_2(\jmath\omega) \tag{5.7}$$

The problem of obtaining the transfer function relations remains only a textbook exercise, once the linearized deviational equations formed around a

steady state (C_{As}, T_s, T_{Js}). This part of analysis is excluded from discussion to keep the manuscript short.

The net change in coolant flow rate, $q_c^{\text{tot}}(s)$ is obtained as a product of transfer function blocks (see Fig. 2)

$$\begin{aligned} q_c^{\text{tot}}(\jmath\omega) &= \tilde{q}_c(\jmath\omega) + \bar{q}_c(\jmath\omega) \\ &= (G_{11}(\jmath\omega)G_{12}(\jmath\omega))C_{A,0,1}(\jmath\omega) + G_{21}(\jmath\omega)G_{22}(\jmath\omega)C_{A,0,2}(\jmath\omega) \end{aligned} \tag{5.8}$$

Using this important equation (Eq. (5.8), we can derive the required stochastic differential equation (SDE). This will be used to derive an expression for stochastic optimal control action.

5.3 Controller design

The variation in feed concentration occurs as

$$C_{A,0}(t) = \bar{C}_{A,0} + \Delta C_{A,0} + \tilde{a}\sin(\omega_e t) \tag{5.9}$$

The cyclic implementation of cooling utility is expressed as a heavy–side function. The general rules that give rise to a heavy–side function are

When $T(t) \geq T_R^{\text{set}}$, the cooling action of icc is required. $q_c^{\text{tot}} > 0$

When $T(t) < T_R^{\text{set}}$, the cooling action of icc is stopped. $q_c^{\text{tot}} = \bar{q}_{c,0}$

Now, we form the stochastic differential balance for optimal control.

$$d\tilde{X}(t) = (1 - \kappa(t))\tilde{X}rdt + \kappa(t)\tilde{X}(Rdt + \sigma dW) - \vartheta(t)dt \tag{5.10}$$

where $\vartheta(t)$ is the *loss* function.

Let the minimum occur as

$$Q = \{(x,t)|t \in [0,\ T;\ x > 0]\} \tag{5.11}$$

Hence, we write that

$$\{\exists\tau|\tau \notin [0,\ T]; \tau > T\} \tag{5.12}$$

Thus, when τ occurs, when final state is reached after computing and implementing the action suggested by optimal control sequence, we say that for next move for optimal control, we have another control set to be computed that

$$\{\hat{A} = \alpha(X^*(s'), s')|\{\kappa(t); \vartheta(t)\}\} \tag{5.13}$$

and then the pay–off functional (a Gaussian distribution) to be minimized is given as

$$P_{x,t}[\hat{A}(\cdot)] = E\left(e^{-\hat{\rho}s}F(\vartheta(t))\right) \tag{5.14}$$

Now, we will derive the Hamilton–Jacobi–Bellman (HJB) equation. Let us begin with the equation for controlled stochastic process.

$$d\tilde{X}(s) = f(\tilde{X}(s), \hat{A}(s))ds + \sigma dW(s); \quad \tilde{X}(t) = x(t) \tag{5.15}$$

The expected value of pay–off functional in Eq. (5.14) becomes

$$P_{x,t}[\hat{A}(\cdot)] = E\left\{\int_t^T \kappa(\tilde{X}(s), \hat{A}(s))ds + \Gamma(\tilde{X}(T)\right\} \tag{5.16}$$

where $\kappa(\cdot)$ is running cost and $\Gamma(\cdot)$ is terminal cost. Now, the value function is given as

$$v(x,t) = \sup_{\hat{A} \in \tilde{A}} P_{x,t}[\hat{A}(\cdot)] \tag{5.17}$$

with a boundary condition, $v(x,T) = \Gamma(x)$. Applying method of dynamic programming, we obtain

$$v(x,t) > E\left\{\int_t^{t+h} \kappa(\tilde{X}(s), \hat{A}(s))ds + v(\tilde{X}(t+h), t+h)\right\} \tag{5.18}$$

Further simplification gives

$$v_t(x,t) + \frac{\sigma^2}{2}\Delta v(x,t) + Min_{a\in\hat{A}} + f(x,a)\cdot\nabla_x v(x,t) + r(x,a) \tag{5.19}$$

This stochastic HJB equation can be applied to our problem as it occurs for various time intervals and the solution as stochastic optimal control sequence can be obtained. Referring to (Evans, 1981) we can easily derive the stochastic HJB equation as

$$\tilde{X}_t + Min\left\{\frac{(\hat{a}_1\sigma x)^2}{2}\tilde{X}_{xx} + (1-\hat{a}_1)rx + \hat{a}_1 xR - \hat{a}_2)\tilde{X}_x + exp(\rho t)F(a_2)\right\} = 0 \tag{5.20}$$

the optimum occurs for the constraints, $0 \leq \hat{a}_1 \leq 1$, $\hat{a}_2 > 0$. This Eq. (5.20) is subject boundary conditions as

$$\tilde{X}(0,t) = 0; \quad \tilde{X}(x,T) = 0 \tag{5.21}$$

The local maximum occurs when

$$\begin{aligned} \kappa^* &= -2\frac{(r-R)}{\sigma^2}; \\ F'(\vartheta^*) &= e^{\rho t}\tilde{X}_x(t) = \frac{e^{\rho t}\gamma\Gamma(t)}{2\sqrt{x}} \end{aligned} \tag{5.22}$$

Setting the value function of a form

$$\tilde{X}(t) = \Gamma(t)(\gamma\sqrt{x}) \tag{5.23}$$

and putting $\hat{a}_1 = \kappa^*$, $\hat{a}_2 = \vartheta^*$, we find that

$$\hat{a}_2 = e^{\rho t}g(t) \tag{5.24}$$

This gives us an expression

$$\vartheta^* = x(exp(\rho t)\Gamma(t))^{-2} \tag{5.25}$$

Now, we define, $h(t) \doteq e^{\rho t}g(t)$, we obtain

$$\hat{B}h'(t)h(t) + \psi h(t)^2 + \tilde{\phi}h(t) = 0 \tag{5.26}$$

The integration constant with one of the boundary conditions in Eq. (5.21) gives us

$$\hat{C} = \frac{\tilde{\phi}}{\psi} exp\left(\frac{T\psi}{\hat{B}}\right) \tag{5.27}$$

Using the integration constant in Eq. (5.27), we obtain an analytical expression for $g(t)$ as

$$\Gamma(t) = \frac{\tilde{\phi}}{\psi}\left(-1 + exp\left(\frac{(T-t)\psi}{\hat{B}}\right)\right) \tag{5.28}$$

The various constant expressions used in above analysis are

$$\tilde{\phi} = 2\gamma\sigma^2 \tag{5.29}$$

$$\psi = \gamma(r^2 + R^2 - (1+2\rho)\sigma^2 + r(-2R + \sigma^2)) \tag{5.30}$$

$$\hat{B} = 2\gamma\sigma^2 \tag{5.31}$$

Thus, we see that we have derived all necessary expressions to implement the stochastic optimal control and we are ready now to apply the same to an illustrative example.

5.4 Results and Discussion

The analytical expressions are obtained for applying the stochastic optimal control to a continuously stirred tank reactor carrying out a first order

exothermic reaction. We see that defining a value function $v(x,t)$ as in Eq. (5.23), we obtain the local optimum values of $\kappa^*(t)$ and $\vartheta^*(t)$ as given by expressions in Eqs. (5.22) and (5.25) respectively. These can be used to find out an optimum usage of utility resources. The cost coefficients of utility resources are r and R for process cold water for sinusoidal motion and brine water for bias with random noise (where excess variation can be expected). The two selected quantities of two utility resources can be mixed well using an eductor and sent into the internal cooling coils. This arrangement then achieves the economic benefits. The terminal cost $g(t)$ is given by Eq. (5.28).

Now, we will see how to implement the analytical results of stochastic optimal control obtained in above section. In order to simulate the controlled process, we find that the sampling arrangement as discussed before gives us estimated values of $\Delta C_{A,0}$, $\tilde{a}$ and ω_e as noted before. Then by arrangement as a step in computational procedure that can be implemented using a computer process control system, we can measure the output feedback control deviation $T'(t)$ and subtract from it the sum $(T_1'(t) + T_2'(t))$. These deviations in temperature i.e. $T_1'(t)$, $T_2'(t)$ are calculated using model components as derived and given by Eqs.(5.7) and (5.8). Therefore subtracting deviations due to sinusoidal forcing function and that due to bias added from measure

temperature deviation we obtain deviation in temperature caused by random variation in feed concentration. Since, this is clearly based on instrumentation used in this apparatus setup, the control system is properly defined. Hence as an implementation of the formulas derived as transfer function relations is clear to us, we can set up a computational procedure to simulate the action of stochastic optimal control sequence.

First we can select a parameter set $\bar{C}_{A,0}$, $\Delta\mathbf{C}_{A,0}$, $\tilde{a}$, ω_e and initial design values of utility flow rates which are $\tilde{q}_{c,0}$ and $\bar{q}_{c,0}$. These two coolant flow rates refer to process cold water and brine water respectively. The economic addition of two sources based on cost of utility resource r and R is used to find out how to compute the value of manipulated variable (cooling flow rate) in order to implement the stochastic optimal control. The computations reduce to a standard MATLAB and SIMULINK simulation and are not done here to keep the manuscript short.

5.5 Conclusions

In this chapter, we derived the analytical results which can be used to form a stochastic optimal controller. The simulations carried out clearly explain

the action and how two utility resources for cooling action can be economically mixed together to produce an effective control action in presence of a randomly varying inlet parameter.

References

Evans, L.C. (1983) *Lecture Notes: An Introduction to Mathematical Optimal Control Theory Version 0.2.*

Inamdar, S.R., Karimi, I.A., Parulekar, S.J. and Kulkarni, B.D., (2011), *A sharp cut algorithm for optimization*, Computers and Chemical Engineering **35** pp. 2716- 2728.

Inamdar, S.R., Karimi, I.A., Parulekar, S.J. and Kulkarni, B.D., (2012), *Application of sharp cut to nonconvex constraints*, Computers and Chemical Engineering, (communicated).

Farr, W., & Aris, R. (1986). *Yet who would have thought the old man to have so much blood in him?Reflections on the multiplicity of steady states of the stirred tank reactor*, Chemical Engineering Science, **41** (6), pp. 13851402.

Anderson, B.D.O., Moore, J.B. (1990), *Optimal Contrl. Linear Quadratic Methods*, Prentice Hall International, Englewood Cliffs, New Jersey.

Hull, D.G. (2003), *Optimal Control Theory for Applications*, Springer–verlag, New York.

Uppal, A., Ray, W.H., Poore, A.B. (April 1974), *On the dynamic behavior of continuous stirred tank reactors*, Chemical Engineering Science, **29**, Issue (4) pp. 967–985.

Uppal, A., Ray, W.H., Poore, A.B. (1976), *The classification of the dynamic behavior of continuous stirred tank reactorsinfluence of reactor residence time*, Chemical Engineering Science, **31**, (3), pp. 205–214.

Notation

$\tilde{a}$	The amplitude of sinusoidal variation in feed concentration.
$\hat{a}_1$	The fractional value of coolant resource used for sinusoidal variation.
$\hat{a}_2$	The extent of loss due to stochastic variation in inlet feed concentration.
$\hat{A}$	The control set that appears in Eq. (5.13).
$\tilde{A}$	The set of all possible control action values in Eq. (5.17).
A_H	The heat transfer area of internal cooling coils.
$\hat{B}$	The constant appearing in the partial differential equation in Eq. (5.26) and given by Eq. (5.31).
C_A	The concentration of reactant A in the reaction mixture.
C_{As}	The steady state value of concentration of A in reactor dynamics.
$C_{A,0}$	The feed concentration value of species A entering the reactor.
$\bar{C}_{A,0}$	The steady state concentration of A when feed concentration is not experiencing any variation.
$\Delta C_{A,0}$	The bias term added to initial parameter value of $\bar{C}_{A,0}$.

$C_{A,0,1}(t)$ The variation in feed concentration.

$C_{A,0,2}(t)$ The variation in feed concentration.

c_J The heat capacity of coolant (brine and cold water) in internal cooling coils.

c_p The heat capacity of reaction mixture in reactor vessel.

E The activation energy in Arrhenius rate form in Eq. (5.1).

F The volumetric flow rate of feed entering the reactor vessel.

F_J The volumetric flow rate of coolant mix in internal cooling coils.

$G_{11}(\jmath\omega)$ The transfer function (control ratio) relating the $\tilde{q}_c(\jmath\omega)$and the temperature deviation $T_1'(\jmath\omega)$.

$G_{12}(\jmath\omega)$ The transfer function (control ratio) relating the $T_1'(\jmath\omega)$ and $C_{A,0,1}(\jmath\omega)$.

$G_{21}(\jmath\omega)$ The transfer function (control ratio) relating the $\bar{q}_c(\jmath\omega)$and the temperature deviation $T_2'(\jmath\omega)$.

$G_{22}(\jmath\omega)$ The transfer function (control ratio) relating the $T_2'(\jmath\omega)$ and $C_{A,0,2}(\jmath\omega)$.

$(-\Delta H)$	The reaction exothermicity as heat released during reaction.
k_0	The pre–exponential factor appearing in Arrhenius rate form.
$\tilde{q}_c(\jmath\omega)$	The value of manipulated variable for coolant i.e. brine.
$\bar{q}_s(\jmath\omega)$	The value of manipulated variable for coolant i.e. cold process water.
$\bar{q}_{c,0}$	The value of deviation in coolant mix flow rate when $T' \to 0$.
$q_c^{\text{tot}}(\jmath\omega)$	The deviation in total coolant mix flow rate as a control action.
Q	A variable defined in Eq. (5.11).
R	Ideal gas law constant.
r	Cost of process cold water available per unit volume and time.
$\hat{R}$	Cost of process cold water available per unit time.
s	A dummy variable in time domain.
$T(t)$	Temperature of reaction mixture in CSTR.
T_s	Steady state value of reaction temperature.
$T_J(t)$	Temperature of coolant water in internal cooling coils.

T_{Js} Steady state value of coolant (water) in internal cooling coils.

T_{J0} Inlet temperature of coolant (brine and cold process water mixed through an eductor).

t Time variable.

$T'(t)$ The deviation in temperature (measured and controlled variable) from set point value.

T_R^{set} Steady state value specified as a set point for control.

$T_1'(t)$ The temperature deviation caused by sinusoidal variation.

$T_2'(t)$ The temperature deviation caused by bias and random variation.

$\hat{T}$ Time interval as free time evolution to move from initial to final state when stochastic optimal control is applied.

U The overall heat transfer coefficient.

V The volume of the stirred tank reactor.

V_J The volume swept by coolant within the internal cooling coils.

$v(t)$ Value function for stochastic optimal control (Eq. (5.18)).

$v(x,t)$	The state variable appearing in the stochastic Hamilton–Jacobi–Bellman (HJB) equation (Eq.(5.19)).
$\tilde{X}(t)$	The temperature variable that appears in stochastic differential equation.
x	The deviation from reference state as the system moves from initial to final state as dictated by stochastic optimal control.

Greek alphabets.

$\alpha(X(t'), t')$	Stochastic optimal control value in Eq. (5.13).
γ	A constant that appears in Eq. (5.22) for stochastic optimal control.
$\vartheta(t)$	The loss function term in the stochastic HJB equation in Eq. (5.14).
$\kappa(t)$	The time–continuous ratio to mix process cold water and brine.
$\xi(t)$	The random variation around mean value of bias in feed concentration.
ρ	The density of reaction mixture in the reactor vessel.
ρ_J	The density of mixture (brine and cold water) in internal cooling coils.

$\hat{\rho}$ A constant that appears in the Gaussian probability density function in Eq. (5.16).

σ The standard deviation that appears in the probability density function in Eq. (5.13).

τ The time variable that appears in Eq. (5.12).

$\tilde{\phi}$ A constant expression appearing in Eq. (5.28) and given by Eq. (5.29).

ψ A constant expression appearing in Eq. (5.28) and given by Eq. (5.30).

ω_e The period of oscillation of sinusoidal disturbance term appearing in Eq. (5.9).

$\Gamma(t)$ Terminal pay–off value appearing in the stochastic optimal control law in Eq. (5.28).

Chapter 6

Concluding remarks

6.1 Conclusions

In this doctoral thesis, we were successful in finding an alternative to existing cutting plane methods, which we termed as a sharp cut method. This was found to be at least twice faster than the existing cutting plane variants. Since it has a higher rate of convergence, an application in control and optimization of a nonisothermal CSTR was solved using the sharp cut. Here, the analysis was extended to a case of considering sinusoidal variation (added with a bias due to parameter shift) and random noise entering the CSTR dynamics. using sufficient instrumentation, we found how we could use cold process water for elimination of sinusoidal variation and fractional addition of brine water to take care of risky spikes appearing in the output response of plant dynamics; this design of stochastic optimal control is presented in detail.

6.2 Recommendations

The principle disadvantage of the optimization method is that the objective function considered is linear. We may use methods known to form piecewise linear approximations of nonlinear objective function within search range of

decision variables. However, it is still a better idea to devise another variant of cutting plane method having higher rate of convergence which can consider the nonlinear objective functions and also will be able to find all (multiple) local optimum solutions. Our initial analysis has identified the weaknesses in the sharp cut algorithm, and, a new approach to overcome this difficulty are underway.

Lastly, we would like to recommend that the flexibility provided by any new algorithm should be embed the same within GAMS software and in its module for global optimization BARON so that the constraint of extending the algorithm to large scale computing to solve real life problems can be overcome!

6.3 Publications from the work done in this thesis

1. Inamdar, S.R., Karimi, I.A., Parulekar, S.J. and Kulkarni, B.D., (2011), Optimal control and optimization of a nonisothermal CSTR. Application of sharp cut algorithm, Eleventh International Symposium on Process Systems Engineering, 15–19 July 2012, Singapore.

2. Inamdar, S.R., Karimi, I.A., Parulekar, S.J. and Kulkarni, B.D., (2011), A sharp cut algorithm for optimization, Computers and Chemical Engineering **35** pp. 2716- 2728.

3. Inamdar, S.R., Karimi, I.A., Parulekar, S.J. and Kulkarni, B.D., (2012), Nonconvex functions and applications of the sharp cut algorithm, Computers & Chemical Engineering, (Revised and under review).

4. Inamdar, S.R., Karimi, I.A., Parulekar, S.J. and Kulkarni, B.D., (2012), Stochastic optimal control of a nonisothermal CSTR, automatica IFAC, (communicated).