Computers and Chemical Engineering 24 (2000) 2193–2209

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

New strategies for flexibility analysis and design under uncertainty

C.G. Raspanti [a], J.A. Bandoni [a,*], L.T. Biegler [b]

[a] *Planta Piloto de Ingeniería Química (UNS-CONICET), Camino La Carrindanga, Km. 7, C.C. 717, 8000 Bahía Blanca, Argentina*
[b] *Chemical Engineering Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA*

Abstract

Process flexibility and design under uncertainty have been researched extensively in the literature. Problem formulations for flexibility include nested optimization problems and these can often be refined by substituting the optimality conditions for these nested problems. However, these reformulations are highly constrained and can be expensive to solve. In this paper we extend algorithms to solve these reformulated NLP problem under uncertainty by introducing two contributions to this approach. These are the use of a Constraint Aggregation function (KS function) and Smoothing Functions. We begin with basic properties of KS function. Next, we review a class of parametric smooth functions, used to replace the complementarity conditions of the KKT conditions with a well-behaved, smoothed nonlinear equality constraint. In this paper we apply the previous strategies to two specific problems: i) the'worst case algorithm', that assesses design under uncertainty and, ii) the flexibility index and feasibility test formulations. In the first case, two new algorithms are derived, one of them being single level optimization problem. Next using similar ideas, both flexibility index and feasibility test are reformulated leading to a single non linear programming problem instead of a mixed integer non linear programming one. The new formulations are demonstrated on five different example problems where a CPU time reduction of more than 70 and 80% is demonstrated.

Keywords: Design; Uncertainty; Flexibility; Non linear programming; Algorithms

1. Introduction

Optimization under uncertainty has been discussed widely in the literature (see Grossmann & Sargent, 1978; Grossmann, Halemane & Swaney, 1983; Swaney & Grossmann, 1985a,b; Ostrovsky, Volin, Barit & Senyavin, 1994; Bandoni Romagnoli & Barton, 1994; Walsh & Perkins, 1996). Here, variations in the operating conditions (e.g. feed flow and concentration, catalytic activity, equipment fouling) and uncertainties in process models for equipment design, indicate that nominal conditions are no longer sufficient to design a process. As a result, process synthesis should also satisfy operating restrictions in spite of disturbances and design uncertainties. To do this, economics and operability must be integrated at the synthesis stage. One way to tackle this problem is to ensure that the worst case occurrences of uncertainty still lead to feasible plants. Several approaches have been developed in the literature to handle this problem. Swaney and Grossmann (1985a) introduced a flexibility index that defines the maximum variations in the uncertain parameters, for which the process constraints can be satisfied. Here, in addition to design variables, control variables can also be adjusted to meet these constraints.

Bandoni et al. (1994) proposed a worst case algorithm where the maximum variation of the uncertain parameters is determined for process feasibility. This is performed using a two-level optimization strategy. First, the control vector is determined at an operational point from an optimization problem for a fixed set of uncertain parameters. This set is then extended by finding the worst constraint violations over the entire domain of uncertain parameters. This approach requires the solution of NLPs at two optimization levels and can lead to a large computational cost for process design problems. This study extends this approach through an aggregation of the constraint functions. Two types of aggregation are considered: the KS function (Kreisselmeier & Steinhauser, 1983) and smoothing functions (Chen & Mangasarian, 1996).

* Corresponding author. Fax: + 54-291-4861700.
E-mail addresses: abandoni@plapiqui.edu.ar (J.A. Bandoni), lb01@andrew.cmu.edu (L.T. Biegler).

0098-1354/00/$ - see front matter
PII: S0098-1354(00)00591-3

While the original worst case (OWC) formulation requires a two level optimization strategy, introducing the KS function to aggregate the optimization problems of the inner level leads to a single, inner level optimization problem. This strategy is called KSWC. One additional approach is then presented which further reformulate the inner and outer level optimization problems to obtain a single level problem formulation. Here the KKT conditions are formulated with the help of a Smoothing Function to replace the complementary conditions. Next, using these ideas, the OWC approach is modified to develop a strategy based on a single optimization problem for this algorithm (formulation SLWC).

The paper is structured as follows. In the next section we introduce the KS function and develop some of its basic properties. In Section 3 a class of smoothing functions is reviewed which are obtained by the double integration of a probability distribution function, and their use to replace complementary condition of a general NLP is also shown. In Section 4 we present the formulation of the original worst case algorithm and the proposed reformulations. Section 5 then presents our new formulations for the feasibility test and flexibility index of Grossmann and coworkers. Several examples are presented in Sections 4 and 5. Section 6 summarizes and concludes the paper.

2. The KS function

This section develops an aggregation technique for inequality constraints. Due to Kreisselmeier and Steinhauser (KS), this approach can be applied to a set of inequality constraints, $g(x) \leq 0$, that appear in a nonlinear programming problem (NLP). A number of definitions and properties for this function can be stated as follows:

2.1. Definition

The KS function, generates an evolving curve for the relations, $y = g_j(\boldsymbol{x})$, $j = 1,\ldots, J$, $\Re^n \rightarrow \Re$. It is assumed that each function is continuous on $\boldsymbol{x}$, but not necessarily continuously differentiable. The KS function can be expressed by the following two equivalent forms:

$$\mathrm{KS}(g_j) = \frac{1}{\rho}\ln\left[\sum_j^J \exp(\rho g_j)\right] \tag{1}$$

$$\mathrm{KS}(g_j) = M + \frac{1}{\rho}\ln\left[\sum_j^J \exp(\rho (g_j - M))\right] \tag{2}$$

Expression (2) is recommended if (1) generates very large values for the exponential term. The parameter ρ is defined by the user and M is a non-negative scalar. Some errors can be produced in the numerical computations if the value of M in (2) is much larger than the maximum value taken by the functions g_j. For example, if the difference between the maximum g_j and M take a value of approximately 50, the maximum individual exponential term $\exp(g_{\max} - M)$ in the summation of (2) is about 2×10^{-22}. Depending on the machine precision, the exponential summation in (2) could be taken as zero, producing an undefined log operation. To avoid these numerical difficulties we define M as:

$$M \approx \max(g_j) \text{ for } j = 1,\ldots, J$$

In this way, a larger value for ρ can be adopted without too much risk of overflow or underflow in the summation. The new expression for (2) would be:

$$\mathrm{KS}(g_j) = g_{\max} + \frac{1}{\rho}\ln\left[\sum_j^J \exp(\rho (g_j - g_{\max}))\right] \tag{3}$$

Taking the limit of (3) for ρ going to infinity, the KS function becomes equal to $g_{\max}$. In other words, the parameter ρ determines the way the KS function covers the functions g_j. Fig. 1 shows an example of KS for a set of functions of a single variable. The function KS can also be seen as a differentiable function similar to the nondifferentiable selective functions of the form $\max(g_j)$ that are available in many high level programming languages.

The derivatives of KS with respect to $\boldsymbol{x}$, can also be obtained for both alternative expressions (1) and (2) of KS:

$$\nabla_x KS = \frac{\sum_j^J \exp(\rho g_j)\nabla_x g_j}{\sum_j^J \exp(\rho g_j)}$$

$$\nabla_x KS = \nabla_x g_{\max} + \frac{\sum_j^J \exp(\rho (g_j - g_{\max}))(\nabla_x g_j - \nabla_x g_{\max})}{\sum_j^J \exp(\rho (g_j - g_{\max}))}$$

$$= \frac{\sum_j^J \exp(\rho g_j)\nabla_x g_j}{\sum_j^J \exp(\rho g_j)}$$

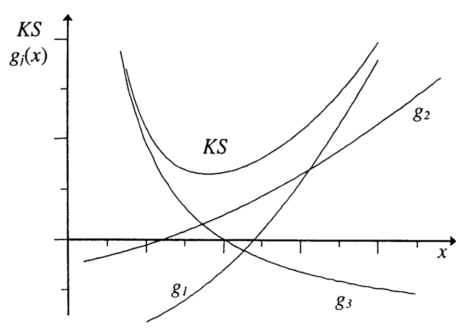


Fig. 1. Overestimation of a set of inequality constraints g_j.

2.2. Properties of the KS function

In this section we develop some basic properties of the KS function that are useful to understand its ability to work as a differentiable overestimator of a set of functions and to formulate the optimality conditions of optimization problems involving the KS function.

The KS function overestimates a set of inequalities constraints of the form $y = g_j(\mathbf{x})$, with $j = 1,\ldots, J$. Starting from the definition of KS in (1), the following properties can be derived.

Property 1

$$\mathrm{KS}(\mathbf{x},\rho) \geq \max_j(g_j(\mathbf{x})), \rho > 0$$

Property 2

$$\lim_{\rho\to\infty} \mathrm{KS}(\mathbf{x},\rho) = \max_j(g_j(\mathbf{x}))$$

Property 3

$$\mathrm{KS}(\mathbf{x},\rho_2) \geq \mathrm{KS}(\mathbf{x},\rho_1) \forall \mathbf{x} \text{ such that } \rho_1 > \rho_2 > 0$$

Property 4

The definitions of the KS functions given by (1) and (2) are equivalent for any value of M.

Property 5

The KS function (1) is insensitive to ρ as ρ becomes large.

Property 6

The gradient of the KS function with respect to $\mathbf{x}$ is independent of $g_{\max}$.

Property 7

Given a convex region defined by a set of convex inequality constraints, $g_j(\mathbf{x}) \leq 0$, the non-empty region defined by $S(\mathbf{x},\rho) \leq C_1$ is also a convex region for any $\rho > 0$ and C_1.

Proofs of these properties are given in Appendix A.

2.3. NLP optimality conditions in terms of the KS function

In this section we now analyze the use of the KS function in nonlinear programs. Consider the following two equivalent inequality constrained NLP problems, where in (P2) the KS function has been used to aggregate the $j = 1,\ldots, m$ inequality constraints:

$$\begin{aligned} &\min_x f(x) \\ &\text{s.t.} \\ &g_j(x) \leq 0 \end{aligned} \tag{P1}$$

$$\begin{aligned} &\min_x f(\mathbf{x}) \\ &\text{s.t.} \\ &\mathrm{KS}(\mathbf{x},\rho) \leq 0 \end{aligned} \tag{P2}$$

The Kuhn–Tucker first order necessary conditions for optimality (Edgar & Himmelblau, 1988) of problem (P1) are the following:

$$\nabla L = \nabla f(\mathbf{x}^*) + \sum_j \mu_j \nabla g_j(\mathbf{x}^*) = 0 \tag{4a}$$

$$\mu_j g_j(\mathbf{x}^*) = 0 \tag{4b}$$

$$\mu_j \geq 0 \tag{4c}$$

$$g_j(\mathbf{x}^*) \leq 0 \tag{4d}$$

where L is the Lagrangian function defined as:

$$L = f(\mathbf{x}) + \sum_j \mu_j g_j(\mathbf{x})$$

On the other hand, for problem (P2), the optimality conditions are:

$$\nabla L = \nabla f(\bar{\mathbf{x}}) + \left[\frac{\mu}{\sum_j \exp(\rho g_j(\bar{\mathbf{x}}))}\right] \sum_j \exp(\rho g_j(\bar{\mathbf{x}})) \nabla g_j(\bar{\mathbf{x}}) =$$

$$= \nabla f(\bar{\mathbf{x}}) + \sum_j \bar{\mu}_j \nabla g_j(\bar{\mathbf{x}}) = 0$$

$$\mu \mathrm{KS}(\bar{\mathbf{x}},\rho) = 0 \mu \geq 0$$

$$KS(\bar{x},\rho) \leq 0$$

where:

$$\bar{\mu}_j = \mu \frac{\exp(\rho g_j(\bar{\mathbf{x}}))}{\sum_k \exp(\rho g_k(\bar{\mathbf{x}}))} > 0$$

and the Lagrangian is:

$$L = f(\mathbf{x}) + \mu \frac{1}{\rho} \ln\left[\sum_j \exp\,(\rho g_j(\mathbf{x}))\right]$$

Let us consider now the following two cases:

Case 1. If $\mathrm{KS}(\bar{\mathbf{x}},\rho) < 0$, then there is no constraint violation and by *Property* 1 of the KS function the following inequalities hold:

$$g_j(\bar{x}) \leq \max_j(g_j(\bar{x})) \leq KS(\bar{x},\rho) < 0$$

and this means there are no active or violated constraints.

Case 2. If $\mathrm{KS}(\bar{x},\rho) < 0$, and for a ρ enough large ($\rho \to \infty$), there is at least one active constraint. Separating the g_j into active and inactive constraints we have:

$A = \{ j \mid -\varepsilon \le g_j(\bar{x}) \le 0\}$ and

$N = \{ j \mid g_j(\bar{x}) < -\varepsilon < 0\}$

where the tolerance ε should be chosen such that the following relation holds for the set of inactive constraints

$$\sum_{j\in N} \exp(\rho g_j(\bar{x})) \le K$$

with $K < 1$. From the assumption of $\mathrm{KS} = 0$ we have $\Sigma_j \exp(\rho g_j(\bar{x})) = 1$, and then separating this expression by active and non active sets we have:

$$\sum_{j\in A} \exp(\rho g_j(\bar{x})) + \sum_{j\in N} \exp(\rho g_j(\bar{x})) = 1$$

and

$$\sum_{j\in A} \exp(\rho g_j(\bar{x})) = 1 - \sum_{j\in N} \exp(\rho g_j(\bar{x})) \ge 1 - K > 0$$

At this point, two lines can be followed to demonstrate that when $\mathrm{KS} = 0$, the set of inequalities in A are active.

Consider an average value for $\Sigma_A(\bullet)$ such that $m = |A|$;

Then $\quad m \exp(\rho\tilde{g}) \ge 1 - K$

and $\quad 0 \ge \tilde{g} \ge \frac{1}{\rho} \ln\left(\frac{1-K}{m}\right)$

Now, for $\rho \to \infty$, from the previous expression we have that $\tilde{g} \to 0$.

3. Smoothing functions

In this section we develop an alternative to the KS function for simplification of the optimality conditions. This arises from the use of smoothing functions and their substitution into the complementary conditions. This approach was developed by Chen and Mangasarian (1996) and has been used as a continuation method to solve general NLPs.

3.1. Definition

Here we consider the Chen and Mangasarian (1996) approximation of the fundamental plus function $(x)_+ = \max\{0, x\}$. These authors proposed a class of parametric smoothing functions obtained by twice integrating a probability density function. The operator $(x)_+ = \max\{0, x\}$ is written as $(x)_+ = \int x - \infty\sigma(y)\mathrm{d}y$, where $\sigma(x)$ is the unitary step function:

$$\sigma(x) = \begin{cases} 1 & x > 0 \\ 0 & x \le 0 \end{cases}$$

The step function can in turn be written as, $\sigma(x) = \int x - \infty(y)\mathrm{d}y$ where $\delta(x)$ is the Dirac delta function, which has the following properties:

$$\delta(x) \ge 0 \text{ and } \int_{-\infty}^{\infty} \delta(y)\mathrm{d}y = 1$$

Both properties indicate that by using a probabilistic density function $p(x)$, it is possible to smooth the Dirac delta and its integral. This function satisfies:

$$p(x) \ge 0 \text{ and } \int_{-\infty}^{\infty} p(y)\mathrm{d}y = 1$$

To make parametric the density function, a new function is defined as:

$$\hat{t}(x,\phi) = \frac{1}{\beta}\,\mathrm{d}\left(\frac{x}{\beta}\right)$$

where β is a positive parameter. When β goes to zero, the limit of $\hat{t}(x,\beta)$ is the Dirac delta function $\delta(x)$. Again, by means of twice integrating $\hat{t}(x,\beta)$, in the first integration an approximation of the highly non convex sigmoidal function is obtained

$$\hat{s}(x,\beta) = \int_{-\infty}^{x} \hat{t}(x,\beta)\mathrm{d}t \approx \sigma(x)$$

and with the second integration, a differentiable approximation of the max operator is obtained.

$$\hat{p}(x,\beta) = \int_{-\infty}^{x} \hat{s}(x,\beta)\mathrm{d}t \approx (x)_+$$

Following a similar development, Chen and Mangasarian develop a family of smoothing functions starting with a different density function $p(x)$.

3.2. Smoothing functions and its relation to the KS function

Chen and Mangasarian (1996) presented three examples of smoothing functions for different density functions. Below we show that one of those examples leads to the KS function. The density function used is:

$$p(x) = \frac{e^{-x}}{(1-e^{-x})^{-2}}$$

By a first integration of the function $\hat{t}(x,\beta)$ defined in the previous section, we have

$$\hat{s}(x,\beta) = \int_{-\infty}^{x} \frac{1}{\beta} \frac{e^{-t/\beta}}{(1+e^{-t/\beta})^2}\,\mathrm{d}t = \frac{1}{(1+e^{-x/\beta})}$$

and by a new integration of this expression $\hat{s}(x,\beta)$ we get:

$$\hat{p}(x,\beta) = \int_{-\infty}^{x} \frac{1}{\beta} \frac{1}{(1+e^{-t/\beta})^2}\,\mathrm{d}t = x + \beta\ln(1+e^{-x/\beta})$$

Note that if in the previous equation we define $\beta = 1/\rho$, we obtain directly the KS function as introduced in Section 2, if it is applied to the functions $f_1(x) = 0$ and $f_2(x) = x$ as shown below:

$$x + \frac{1}{\rho}\ln(1 + e^{-\rho x}) = \frac{1}{\rho}\ln(1 + e^{\rho x}) = \frac{1}{\rho}\ln(e^0 + e^{\rho x})$$

3.3. Application to nonlinear programming

Let us consider a subset of the Kuhn–Tucker conditions of a NLP given by the Eqs. (4b), (4c) and (4d).

$$\begin{aligned} \lambda g(x) &= 0 \\ \lambda &\geq 0 \\ g(x) &\leq 0 \end{aligned}$$

To overcome the difficulty associated with the solution of the complementary condition $\lambda.g(x) = 0$, Clark (1983) proposed the incorporation of binary variables as follow:

$$\begin{aligned} \lambda &\leq Uy \\ -g(x) &\leq U(1-y) \\ \lambda &\geq 0 \\ g(x) &\leq 0 \end{aligned} \tag{5}$$

where U is a positive scalar. On the other hand, it is easy to show (Chen & Mangasarian, 1996) that (5) is equivalent to the following expression in terms of a $\max\{0, x\}$defined above (see Appendix B for an enumeration of cases):

$$\lambda - \max(0, \lambda + g(x)) = 0$$

Now, the discontinuity in the derivative of the $\max(0, x)$ operator, can be avoided by making use of the smoothing functions results presented in Section 3.1. By considering $\lambda + g(x)$ as the argument of smoothing function $\hat{p}(x,\beta)$, we can get

$$\lambda - (\lambda + g(x))_+ = 0 \tag{6}$$

to replace the conditions (4b), (4c) and (4d) in 4. In this way, we could avoid the use of the binary variables, solving the difficulty of the complementary condition in 4 in an entirely continuous way.

4. Worst case algorithm

The problem of process flexibility and design under uncertainty has been studied extensively in the literature of Grossmann et al. (1983), Swaney and Grossmann (1985a), Grossmann and Floudas (1987), Polak (1982), Polak, Stimler and Majorisation (1988), Mayne, Michalska and Polak (1990) and Tits (1985). One of the simpler approaches to solve this problem is the so called 'worst case' algorithm. Walsh and Perkins (1996) reviewed these problems and discussed the usefulness of this approach in practical situations. Bahri, Bandoni and Romagnoli (1997), Bahri, Bandoni and Romagnoli (1996a), Bahri, Bandoni and Romagnoli (1996b), Figueroa, Bahri, Bandoni and Romagnoli (1996), Sunarto, Bandoni, Barton and Romagnoli (1994) and Bandoni et al. (1994) also attacked this problem from an optimizing control point of view in what they call the Back-Off problem.

In this section we present the worst case (WC) algorithm to solve the general NLP problem under uncertainty. First, we present the formulation, and its solution procedure through the two level optimization strategy used by the WC. We refer to this as the original worst case algorithm (OWC). Later on, we reformulate the solution strategy by introducing the KS function presented in Section 2 to aggregate the 'J' optimization problems of the inner level into a single optimization problem. This strategy is called KSWC. Two more approaches are presented, based on formulating the Karush–Kuhn–Tucker (KKT) conditions of the inner problem and adding them to the outer level optimization problem. This leads to a solution procedure with a single level optimization strategy. The KKT conditions are formulated with the help of the smoothing functions presented in Section 3 to replace the binary variables required to formulate the complementary conditions. Using these ideas, the previous approach, OWC, is reformulated to produce the single level worst case SLWC strategy. Finally, we present three applications examples and carry out a comparative study of CPU time.

4.1. Problem formulation

Consider the following general NLP problem under uncertainty:

$$\begin{aligned} &\min_z \Phi(\mathbf{x},\mathbf{z},\theta^N) \\ &\quad \text{s.t.} \\ &\mathbf{h}(\mathbf{x},\mathbf{z},\theta) = 0 \\ &\mathbf{g}(\mathbf{x},\mathbf{z},\theta) \leq 0 \\ &z \in \{\mathbf{z}:\mathbf{z}^L \leq \mathbf{z} \leq \mathbf{z}^U\} \\ &\theta \in \Gamma \end{aligned} \tag{7}$$

where: Φ: objective function, $\mathbf{z}$: vector of decision (optimization) variables, $\mathbf{x}$: vector of state variables, $\boldsymbol{\theta}$: vector of uncertain parameters, $\boldsymbol{\theta}^N$: vector of nominal values of the uncertain parameters, $\mathbf{h}$: vector of equality constraints, $\mathbf{g}$: vector of inequality constraints. Note that in this problem it is assumed that well-defined upper, lower and nominal values are available for each uncertain parameter, that is $\boldsymbol{\theta} \in \Gamma = \{\boldsymbol{\theta}:\boldsymbol{\theta}^L \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^U\}$. Given that we are assuming any value of $\boldsymbol{\theta}$ is equally provable, each realization develops a different set if inequality constraints and then, problem (7) has formally an infinite number of constraints. The WC algorithm finds a optimal point **z***, satisfying all possible realizations of **θ,** if it exists. Here, we do not discuss convexity and global optimality issues, but they are required as in any process flexibility and design under uncertainty problem, if the convergence must be guaranteed.

4.2. Review of the WC algorithm

The worst case algorithm (Bandoni et al., 1994), denoted in this paper as WC algorithm, or OWC), consists of a two level optimization strategy. At the outer level, problem (ol-OWC) is solved for fixed values of the uncertain parameters. At the inner level, the feasibility of the constraints is tested around the optimum point generated at the outer level. The algorithm iterates around these two levels until no constraint violation is possible for the current optimal point.

At the inner level, the solution of J optimization problems is required (where J is the number of inequality constraints in problem (il-OWC)), in the space of the uncertain parameters. The mathematical formulation of these two optimization levels is as follows: Original strategy for the WC algorithm-**(OWC)**

Outer level

$$\begin{aligned}&\min_{z} \Phi(\mathbf{z},\mathbf{x},\theta^N)\\ &\quad \text{s.t.}\\ &\mathbf{h}(\mathbf{x},\mathbf{z},\theta^N)=0\\ &\mathbf{g}(\mathbf{x},\mathbf{z},\theta^N)\le 0\\ &\left.\begin{aligned}\mathbf{h}(\mathbf{x},\mathbf{z},\theta_v^k)=0\\ g_v^k(x,z,\theta_v^k)\le 0\end{aligned}\right\}\quad \begin{aligned}v\in V^k\\ k=1,\ldots,K\end{aligned}\\ &\mathbf{z}\in Z=\{\mathbf{z}|\mathbf{z}^L\le\mathbf{z}\le\mathbf{z}^U\}\end{aligned}\tag{ol-OWC}$$

where k is the iteration index between both optimization levels, K is the total iteration number, V^k is the set containing the index of the violated constraints g_v^k. Therefore θ_v^k is the value of θ that produces the largest violation of constraint v in the iteration k.

Inner level

$$\left.\begin{aligned}&\max_{\theta} g_j(x,z^*,\theta)\\ &\quad \text{s.t.}\\ &h(x,z^*,\theta)=0\\ &\theta\in\Gamma=\{\theta|\theta^L\theta\le\theta^U\}\end{aligned}\right\} j=1,\ldots,J\tag{il-OWC}$$

where z^* is the optimum found one at the outer level.

4.3. WC algorithm using the KS function (KSWC algorithm)

The feasibility test of the WC algorithm at the inner level consists of verifying that for the current z^*, the inequality constraints

$g_j(x,z^*,\theta)\le 0$ for $j=1,\ldots,J$ and for $\theta\in\Gamma$

are not violated. This is done by solving maximization problems at the inner level. Using property A of KS function, the feasibility test can be reformulated as follows (il-KSWC):

$$\max_j[g_j(x,z^*,\theta)]\le \mathrm{KS}(\rho,g_j(x,z^*,\theta)) \ \text{ for } j=1,\ldots,J$$

then

$$\max_\theta \max_j[g_j(x,z^*,\theta)]\le \max_\theta \mathrm{KS}(\rho,g_j(x,z^*,\theta)) \ \text{ for } j=1,\ldots,J$$

and then the entire WC algorithm can be stated as:
WC strategy using the KS function (KSWC)
Outer level

$$\begin{aligned}&\min_{z} \Phi(\mathbf{z},\mathbf{x},\theta^N)\\ &\quad \text{s.t.}\\ &\mathbf{h}(\mathbf{x},\mathbf{z},\theta^N)=0\\ &\mathbf{g}(\mathbf{x},\mathbf{z},\theta^N)\le 0\\ &\left.\begin{aligned}\mathbf{h}(\mathbf{x},\mathbf{z},\theta^k)=0\\ g_v^k(\mathbf{x},\mathbf{z},\theta^k)\le 0\end{aligned}\right\}\quad \begin{aligned}v\in V^k\\ k=1,\ldots,K\end{aligned}\\ &\mathbf{z}\in Z=\{\mathbf{z}|\mathbf{z}^L\le\mathbf{z}\le\mathbf{z}^U\}\end{aligned}\tag{ol-KSWC}$$

where k is the iteration index between both optimization levels, K is the total iteration number, V^k is the set containing the index of the violated constraints g_v^k.

Inner level

$$\begin{aligned}&\max_{\theta} \mathrm{KS}(x,z^*,\theta)\\ &\quad \text{s.t.}\\ &h(x,z^*,\theta)=0\\ &\theta\in\Gamma=\{\theta|\theta^L\le\theta\le\theta^U\}\end{aligned}\tag{il-KSWC}$$

where:

$$\mathrm{KS}(x,z^*,\theta)=\frac{1}{\rho}\ln\left[\sum_{j=1}^{J}\exp\left(\rho g_j(x,z^*,\theta)\right)\right]$$

4.4. WC original algorithm as single level optimization problem (SLWC algorithm)

We next consider a transformation of the two level strategy of the worst-case algorithm into a single level by formulating the Karush Kuhn–Tucker optimality conditions of the inner problems (il-OWC) and including them as constraints in the outer level. By using the

Table 1
Variation ranges for the uncertain parameters in the four cases analyzed

	C_1^F	C_m	T_1^F	T_m
Case I	$19.5 \leq C_1^F \leq 21.0$	20.0	300.0	300.0
Case II	$19.5 \leq C_1^F \leq 21.0$	$19.5 \leq C_m \leq 21.0$	300.0	300.0
Case III	$19.5 \leq C_1^F \leq 21.0$	$19.5 \leq C_m \leq 21.0$	$295 \leq T_1^F \leq 315$	300.0
Case IV	$19.5 \leq C_1^F \leq 21.0$	$19.5 \leq C_m \leq 21.0$	$295 \leq T_1^F \leq 315$	$295 \leq T_m \leq 315$

smoothing functions from Section 3, the Karush Kuhn–Tucker optimality conditions of a general inner loop problem can be written as follows:

$$L(x, z^*, \theta) = -g_j(x, z^*, \theta) + \mu^T h(x, z^*, \theta) + \lambda^{LT}(\theta^L - \theta) + \lambda^{UT}(\theta - \theta^U)$$

and the KKT conditions are:

$$\nabla_x L(x, z^*, \theta) = -\nabla_x g_j(x, z^*, \theta) + \mu^T \nabla_x h(x, z^*, \theta) = 0 \tag{8a}$$

$$\nabla_\theta L(x, z^*, \theta) = -\nabla_\theta g_j(x, z^*, \theta) + \mu^T \nabla_\theta h(x, z^*, \theta) - \lambda^L + \lambda^U = 0 \tag{8b}$$

$$\lambda^T \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{z}) = 0, \ \lambda^{LT}(\theta^L - \theta) = 0, \ \lambda^{UT}(\theta - \theta^U) = 0 \tag{8c}$$

$$\lambda \geq 0 \quad \lambda^L \geq 0, \lambda^U \geq 0 \tag{8d}$$

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{z}) \leq 0 \tag{8e}$$

Now, the set of Eqs. (8c), (8d) and (8e) can be replaced using the results of Section 3.1 by the following expression $\lambda - \max\{0, \lambda + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{z})\} = 0$ and similar expressions for the bounds on θ.

Single level WC strategy (SLWC)

$$\min_z \ \Phi(x,z,\theta^N)$$
$$\text{s.t.}$$
$$h(x,z,\theta^N) = 0$$
$$g(x,z,\theta^N) \leq 0$$

$$\left.\begin{array}{r} h_j(x_j,z,\theta_j) = 0 \\ g_j(x_j,z,\theta_j) \leq 0 \\ -\nabla_x g_j(x_j,z,\theta_j) + \mu_j^T \nabla_x h_j(x_j,z,\theta_j) = 0 \\ -\nabla_\theta g_j(x_j,z,\theta_j) + \mu_j^T \nabla_\theta h_j(x_j,z,\theta_j) - \lambda_j^L + \lambda_j^U = 0 \\ \lambda_j^L - (\lambda_j^L + \theta_j^L - \theta_j)_+ = 0 \\ \lambda_j^U - (\lambda_j^U + \theta_j - \theta_j^U)_+ = 0 \end{array}\right\} j =$$

$$1,\ldots,J z \in Z = \{z | z^L \leq z \leq z^U\}$$

where: $(^\bullet)_+ = \max(0, ^\bullet)$

4.5. Numerical results

4.5.1. Example problems

For the four worst case formulations developed in the previous section, we consider three examples to compare the required computational effort. Following are the mathematical formulation of the examples. (see Table 1)

Example 1: (Pistikopoulos, 1988)

$$\min_z \Phi = (z-2)^2 + \theta^N$$
$$\text{s.t.}$$
$$g_1 = z - \theta \leq 0$$
$$g_2 = -z - \theta/3 + 4/3 \leq 0$$
$$g_3 = z + \theta - 4 \leq 0$$
$$0.5 \leq z \leq 4.0$$
$$\theta \in \Gamma = \{\theta : 1 \leq \theta \leq 3\}$$
$$\theta^N = 2$$

Example 2: (Bandoni, 1987)

$$\min_z \Phi = 1(z_1 + 3z_2)$$
s.t.
$$g_1 = x_1 z_1 + x_2 z_2 - x_5 \leq 0$$
$$g_2 = x_3 z_1 + x_4 z_2 - x_6 \leq 0$$
$$h_1 = x_1 - (-4\theta_1^2 + 3\theta_2^2 + 3.25) = 0$$
$$h_2 = x_2 - (\theta_1 + 2\theta_2 - 1.75) = 0$$
$$h_3 = x_3 - (\theta_2 - 6) = 0$$
$$h_4 = x_4 - (-3\theta_1^2 + 4\theta_2^2 + 4\theta_1\theta_2 - \theta_1 - 1) = 0$$
$$h_5 = x_5 = (3\theta_1 + \theta_2 + 5.5) = 0$$
$$h_6 = x_6 - 10 = 0$$
$$0 \leq z_1 \leq 2$$
$$0 \leq z_2 \leq 2$$
$$\theta \in \Gamma = \{\theta : 0.5 \leq \theta 1 \leq 2.5, \ 0.5 \leq \theta 2 \leq 2.5, \ \theta_1^2 - 3\theta_1 \text{-} \theta_2 + 2.75 \leq 0\}$$
$$\theta^N = (1.0 \quad 1.5)$$

Example 3: (De Hennin & Perkins, 1991)

$$\min \Phi = 10(F_1^F C_1^F + F_m C_m - 0.3F_2) - 0.01Q_1^r - 1.0Q_2^r - 0.1F_1^F - 0.1F_m$$

$$g_1 = T_1 - 350 \leq 0 \quad K$$

$$g_2 = T_2 - 350 \leq 0 \quad K$$

$$g_3 = F_m + F_1^F - 0.8 \leq 0 \quad \mathrm{m^3/s}$$

$$g_4 = Q_1^r - 30 \leq 0 \quad \mathrm{cal/s}$$

$$g_5 = Q_2^r - 20 \leq 0 \quad \mathrm{cal/s}$$

$$g_6 = C_2 - 0.3 \leq 0 \quad \mathrm{mol/m^3}$$

$$h_1 = -k_0 e^{\frac{-E}{RT_1}} C_1 V_1 + F_1^F(C_1^F - C_1) = 0$$

$$h_2 = -(\Delta H_r) k_0 e^{\frac{-E}{RT_1}} C_1 V_1 + F_1^F(T_1^F - T_1) - Q_1^r = 0$$

$$h_3 = -k_0 e^{\frac{-E}{RT_2}} C_2 V_2 + F_2^F(C_2^F - C_2) = 0$$

$$h_4 = -(\Delta H_r) k_0 e^{\frac{-E}{RT_2}} C_2 V_2 + F_2^F(T_2^F - T_2) - Q_2^r = 0$$

$$h_5 = F_1 + F_m - F_2^F = 0$$

$$h_6 = (F_1 C_1 + F_m C_m)/F_2^F - C_2^F = 0$$

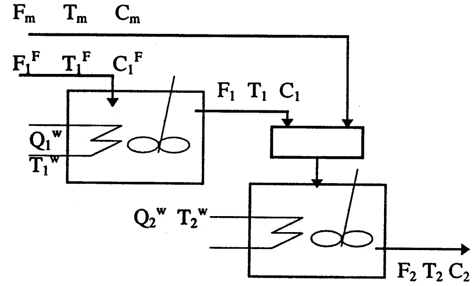
$$h_7 = (F_1 T_1 - T_1^w)/F_2^F - T_2^F = 0$$

$$h_8 = UA(T_1 - T_1^w) - Q_1^r = 0$$

$$h_9 = UA(T_2 - T_2^w) - Q_2^r = 0$$

$$0.05 \leq F_1^F$$

$$0.05 \leq F_m$$



4.6. Discussion of results

Table 2 presents the CPU times obtained for each of the three examples considered for the four approaches presented in Section 4. From the table, we observe that the SLWC approach reduces the solution time to about 84% of the time required by the original OWC approach. Also note that this percentage of reduction is quite similar for the three examples. The results also show that instead of this constant percentage of reduction, the KSWC approach leads to different percentages of reduction, depending on how many outer level iterations are required.

Table 3 shows the influence of the number of uncertain parameters on the solution time. These results correspond to Example 3. It can be observed that in the case of the OWC algorithm, the CPU time required is quite similar using up to four uncertain parameters. In the case of the KSWC approach, the increment in CPU time from one to four uncertain parameter is given by a jump (in this case produced when passing from two to three uncertain parameters). This behavior is typical of the KSWC approach, where the jumps correspond to an additional outer level iteration required by the algorithm.

Table 2
CPU times (in 1/100 s) for the three examples with the four approaches. The results reported for Example 3 correspond to two uncertain parameters for this example (Case II)

Algorithm		Example 1		Example 2		Example 3	
		CPU time	Reduction (%)	CPU time	Reduction (%)	CPU time	Reduction (%)
OWC	98	–	80	–	211	–	
KSWC	50	49.0	73	8.7	56	73.4	
SLWC	15	84.7	13	83.7	33	84.4	

Table 3
Comparison of solution times for the Example 3 using the four approaches and considering different number of uncertain parameters

Algorithm	Case I		Case II		Case III		Case IV	
	CPU time	Reduction (%)	CPU time	Reduction (%)	CPU time	Reduction (%)	CPU time	Reduction (%)
OWC	199.5	–	203.0	–	200.8	–	206.6	–
KSWC	57.8	71.0	58.6	71.1	88.8	55.8	89.2	56.8
SLWC	25.8	87.1	29.0	85.7	29.4	85.4	30.2	85.4

Table 4
CPU time for increasing number of inequality constraints for Example 1[a]

Algorithm	1 constraint		2 constraints		3 constraints	
	CPU time	Reduction (%)	CPU time	Reduction (%)	CPU time	Reduction (%)
OWC	50.0	–	74.6	–	98.2	–
KSWC	50.0	0.0	50.4	32.4	51.0	48.1
SLWC	12.8	74.4	13.8	81.5	13.4	86.4

[a] The CPU times are expressed in 1/100 sec. All results were obtained with GAMS using MINOS running on a micro VAX.

Table 4 presents the influence of the inclusion of additional constraints in Example 1, solved by the four approaches. As can be expected, OWC requires more CPU time for larger numbers of constraints, due to the larger number of inner level optimization problems. In this example, the proportional increase in the solution time with respect to the number of constraints is due to the similarity of the inner problem complexity. This behavior cannot be expected in a general case.

The optimization problems in the KSWC algorithm are smaller and then easier to solve, particularly for problems with large number of inequality constraints. In our experience it was quite easy to set the correct value of the adjusting parameter ρ for the KSWC algorithm. Normally a single value of ρ between 5 and 10 is good enough to get solution without numerical difficulties. These range for ρ is also mentioned as good enough to solve practical problems by Sobieszczanski (1992) and Sobieszczanski, James and Riley (1987), who have made extensive use of the KS function to solve different optimization problems for root location and structural sizing.

On the other hand, given that the SLWC algorithm solves a single NLP, it is much faster getting the solution (about 85% less CPU time is required in the examples solved). Despite this advantage, this single NLP is harder to solve than the individual NLPs of the inner loop of the KSWC algorithm. Additionally to the fact that the NLP of the SLWC is larger in size, the main reason for this behavior is that the nonlinearity added to the original problems because the solution strategies, in the case of the KSWC goes to the objective function, while in the case of the SLWC goes into the constraints. This makes harder obtaining initial points and feasible solutions in the case of SLWC.

For highly non linear problems or with large number of constraints the KSWC would be preferred to the SLWC, because the reasons given above. Diaz, Raspanti, Bandoni and Brignole (1999) have used this strategy to study worst case situations in a large scale plant with thirty inequality constraints. The KSWC was interfaced to an ad-hoc process simulator of a natural gas plant, and the KSWC got the same solution as the OWC algorithm in 95% lees of the CPU time. On the other hand, if the problem is not highly non linear or when it is known that only a few active constraints could occurs, the SLWC could be preferred.

5. KS and smoothing function in flexibility analysis

5.1. Problem formulation and existing solution procedures

In this section we present new solutions to the flexibility test and the feasibility index developed by Grossmann and coworkers. They are based on the incorporation of the KS function to reformulate the solution strategy for solution of these two problems. We also present a strategy, that making use of smoothing functions avoid the binary variables in the previous formulation of these problems. For the sake of clarity, we repeat here the basic formulations for both problems. The Feasibility Test (Grossmann et al., 1983) is defined as:

$$\begin{aligned} &\chi(\boldsymbol{d}) = \max_{\theta\in\Gamma} \min_{z} \max_{j\in J} g_j(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta) \\ &\text{s.t. } h(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta) = 0 \end{aligned} \tag{9}$$

where $\chi(\boldsymbol{d})$ is considered the feasibility degree of a given design. If $\chi(\boldsymbol{d}) \leq 0$ the operation is feasible for all $\theta\in\Gamma$, where $\Gamma = \{\theta | \theta^U \geq \theta \geq \theta^L\}$. On the other hand, if $\chi(\boldsymbol{d}) > 0$, the design will not be feasible at least for some values of θ due to the violation of at least one constraint.

The flexibility index (Swaney & Grossmann, 1985a) is expressed in the following way:

$$F = \max \delta \tag{10}$$

$$\text{s.t. } \max_{\theta\in\Gamma(\delta)} \min_{z} \max_{j\in J} g_j(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta) \leq 0$$

$$h(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta) = 0$$

$$\Gamma(\delta) = \{\theta^N - \delta\Delta\theta^- \leq \theta \leq \theta^N + \delta\Delta\theta^+\}, \quad \delta \geq 0$$

where $\Gamma(\delta)$ is the set of uncertain parameters defined through the scalar variable δ. This index F defines the maximum set parameter $\Gamma(F)$ so that a determined design can be permanently feasible.

The solution of these two problems is generally complicated because the max-min-max operator represents a nonlinear, nondifferentiable multilevel optimization problem. Grossmann and coworkers proposed a solution strategy for both problems based on a decomposition into two optimization levels. In this way, the feasibility test problem (9) was formulated as:

$$\chi(\boldsymbol{d}) = \max_{\theta \in \Gamma} \psi(\boldsymbol{d}, \theta) \tag{11}$$

$$\text{s.t.} \quad \psi(\boldsymbol{d}, \theta) = \min_{z} \max_{j \in J} g_j(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta)$$
$$h(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta) = 0$$

where $\psi(\boldsymbol{d},\theta)$ corresponds to the following nonlinear problem:

$$\psi(\boldsymbol{d}, \theta) = \min_{z, u} u \tag{12}$$

$$\text{s.t.} \quad g_j(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta) \le u \quad j \in J$$
$$h(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta) = 0$$

where u is a scalar variable. Grossmann and Floudas (1987) developed a solution procedure based on the formulation of the Kuhn–Tucker conditions of problem (12) and included as constraints into the outer problems. Binary variables were used to resolve the complementary conditions. In these way, they derived two Mixed Integer Non Linear programming formulations. For the feasibility text the formulation is:

$$\chi(\boldsymbol{d}) = \max_{z} u \tag{13}$$

$$\text{s.t.} \quad g_j(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta) + s_j - u = 0 \quad j \in J$$
$$h_i(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta) = 0 \quad i \in I$$

$$\sum_{j \in J} \lambda_j = 1$$

$$\sum_{i \in I} \mu_i \frac{\partial h_i}{\partial z} + \sum_{j \in J} \lambda_j \frac{\partial g_j}{\partial z} = 0$$

$$\sum_{i \in I} \mu_i \frac{\partial h_i}{\partial x} + \sum_{j \in J} \lambda_j \frac{\partial g_j}{\partial x} = 0$$

$$\left.\begin{array}{r} \lambda_j - y_j \le 0 \\ s_j - U(1 - y_j) \le 0 \\ \lambda_j \ge 0, \quad s_j \ge 0 \end{array}\right\} j \in J$$

$$\sum_{j \in J} y_j \le n_z + 1$$

$$\theta^U \ge \theta \ge \theta^L$$

In the same way, the flexibility index problem is formulated as follows:

$$F = \min_{z} \delta$$

$$\text{s.t.} \quad g_j(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta) + s_j = 0 \quad j \in J$$
$$h_i(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta) = 0 \quad i \in I$$

$$\sum_{j \in J} \lambda_j = 1$$

$$\sum_{i \in 1} \mu_i \frac{\partial h_i}{\partial z} + \sum_{j \in J} \frac{\partial g_j}{\partial z} = 0$$

$$\sum_{i \in I} \mu_i \frac{\partial h_i}{\partial x} + \sum_{j \in J} \lambda_j \frac{\partial g_j}{\partial x} = 0$$

$$\left.\begin{array}{r} \lambda_j - y_j \le 0 \\ s_j - U(1 - y_j) \le 0 \\ \lambda_j \ge 0, \quad s_j \ge 0 \end{array}\right\} j \in J$$

$$\sum_{j \in J} y_j \le n_z + 1$$

$$\theta^N - \delta \Delta\theta^- \le \theta \le \theta^N + \delta \Delta\theta^+, \quad \delta \ge 0 \tag{14}$$

5.2. Reformulation by incorporating the KS function

As discussed in the first section of this paper, the KS function overestimates a set of inequalities of the form $g(x) \le 0$. That means that for a large ρ, it verifies that:

$$\text{KS}(\boldsymbol{x}, \rho) \ge \max_{j}(g_j(\boldsymbol{x})), \quad \rho > 0$$

where x denotes the vector of variables, and the scalar ρ is the adjustable parameter of the KS function.

Now, if we look at the Eq. (11) of the flexibility analysis, it can be recognized that by incorporating the KS function to replace the $\max_j(g_j(x))$ the flexibility test can be written in an equivalent form as follows:

$$\chi(\boldsymbol{d}) = \max_{\theta \in \Gamma} \min_{z} \text{KS}(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta)$$
$$\text{s.t.} \ h(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta) = 0$$

As can be observed, the use of the KS function allows the elimination of one of the optimization levels, simplifying the formulation and consequently its resolution procedure as it will become clear later. The feasibility function $\psi(\boldsymbol{d}, \theta)$ can now be defined as:

$$\psi(\boldsymbol{d}, \theta) = \min_{z} u \tag{15}$$

$$\text{s.t.} \quad \text{KS}(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta) \le u$$
$$h_i(\boldsymbol{d}, \boldsymbol{z}, \boldsymbol{x}, \theta) = 0 \quad i \in I$$

The Lagrangian function for this problem is

$$L = u + \lambda(\text{KS} - u) + \sum_{i \in I} \mu_i h_i = 0$$

and the corresponding Kuhn–Tucker conditions are

$$\frac{\partial L}{\partial u} = 1 - \lambda = 0 \Rightarrow \lambda = 1 \Rightarrow \text{KS} = u \tag{16a}$$

$$\frac{\partial L}{\partial z} = \frac{\partial \text{KS}}{\partial z} + \sum_{i \in I} \mu_i \frac{\partial h_i}{\partial z} = 0 \tag{16b}$$

$$\frac{\partial L}{\partial x} = \frac{\partial \text{KS}}{\partial x} + \sum_{j \in I} \mu_i \frac{\partial h_i}{\partial x} = 0 \tag{16c}$$

As can be observed from (16a) Lagrange multiplier corresponding to the single inequality constraint in problem (15) is always $\lambda = 1$ at the optimum. This means that the inequality constraint is always active and $\mathrm{KS} = u$. Then, the feasibility test problem (13) can be formulated as below. However, note that the KKT first order conditions used to derived equations (16) are only necessary but not sufficient in the general case, and therefore the formulation below might not be equivalent to (15) in the general case.

Feasibility test problem with the KS function (KSFT)

$$\max_{\theta} \mathrm{KS}$$
$$\text{s.t. } h_i(\boldsymbol{d}, z, \boldsymbol{x}, \theta) = 0, i \in I$$
$$\frac{\partial \mathrm{KS}}{\partial \mathbf{z}} + \sum_{i \in I} \mu_i \frac{\partial h_i}{\partial \mathbf{z}} = 0$$
$$\frac{\partial \mathrm{KS}}{\partial \mathbf{x}} + \sum_{i \in I} \mu_i \frac{\partial h_i}{\partial \mathbf{x}} = 0$$
$$\theta^L \le \theta \le \theta^U$$

Note that this problem is just an NLP problem, as compared with the MINLP problem of formulation from Grossmann and Floudas (1987). The reason why integer variables are not required is that the complementary condition of single inequality constraint of problem (15) is always satisfied for $\lambda = 1$, and then no development in terms of binary variables is required.

Following a similar derivation, the problem for the flexibility index (14) can be reformulated also as a NLP problem as follows:

Flexibility index problem with the KS function (KSFI)

$$F = \min_{\theta} \delta$$
$$\text{s.t. } \mathrm{KS} = 0$$
$$h_i(\boldsymbol{d}, z, \boldsymbol{x}, \theta) = 0, i \in I$$
$$\frac{\partial \mathrm{KS}}{\partial z} + \sum_{i \in I} \mu_i \frac{\partial h_i}{\partial z} = 0$$
$$\frac{\partial \mathrm{KS}}{\partial x} + \sum_{i \in I} \mu_i \frac{\partial h_i}{\partial x} = 0$$
$$\theta^N - \delta \Delta\theta^- \le \theta \le \theta^N + \delta \Delta\theta^+, \quad \delta \ge 0$$

5.3. Incorporation of smoothing functions

Another derivation for the flexibility test and flexibility index problems can be obtained by directly using a smoothing function to reformulate the complementary conditions of problem (13) as follow:

$$\left.\begin{aligned} s_j = u - g_j(\boldsymbol{d}, z, \boldsymbol{x}, \theta) \\ \lambda_j s_j = 0 \\ \lambda_j \ge 0, \quad s_j \ge 0 \end{aligned}\right\} \Leftrightarrow \lambda_j - \max(0, \lambda_j + g_j(\boldsymbol{d}, z, \boldsymbol{x}, \theta) - u)$$

Now, once again the non differentiability of the $\max\{0, x\}$ operator can be avoided using the smoothing function as it was shown in Section 3.3. By applying these transformations, both problems (13) and (14) can be reformulated as the following two NLP problems:

Feasibility test problem with smoothing functions (SFFT)

$$\chi(\boldsymbol{d}) = \max u$$
$$\text{s.t. } g_j(\boldsymbol{d}, z, \boldsymbol{x}, \theta) - u \le 0 \quad j \in J$$
$$h_i(\boldsymbol{d}, z, \boldsymbol{x}, \theta) = 0 \quad i \in I$$
$$\sum_{j \in I} \lambda_j = 1$$
$$\sum_{i \in I} \mu_i \frac{\partial h_i}{\partial z} + \sum_{j \in I} \lambda_j \frac{\partial g_j}{\partial \boldsymbol{z}} = 0$$
$$\sum_{i \in I} \mu_i \frac{\partial h_i}{\partial x} + \sum_{j \in j} \lambda_j \frac{\partial g_j}{\partial \boldsymbol{x}} = 0$$
$$\lambda_j - (\lambda_j + g_j(\boldsymbol{d}, z, \boldsymbol{x}, \theta) - u)_+ = 0$$
$$\theta^U \ge \theta \ge \theta^L$$

where: $(^\bullet)_+ = \max(0, {}^\bullet)$

Flexibility index problem with smoothing function (SFFI)

$$F = \min_{z} \delta$$
$$\text{s.t. } g_j(\boldsymbol{d}, z, \boldsymbol{x}, \theta) \le 0 \quad j \in J$$
$$h_i(\boldsymbol{d}, z, \boldsymbol{x}, \theta) = 0 \quad i \in I$$
$$\sum_{j \in J} \lambda_j = 1$$
$$\sum_{i \in I} \mu_i \frac{\partial h_i}{\partial \boldsymbol{z}} + \sum_{j \in J} \lambda_j \frac{\partial g_j}{\partial \boldsymbol{z}} = 0$$
$$\sum_{i \in I} \mu_i \frac{\partial h_i}{\partial \boldsymbol{x}} + \sum_{j \in J} \lambda_j \frac{\partial g_j}{\partial \boldsymbol{x}} = 0$$
$$\lambda_j - (\lambda_j + g_j(\boldsymbol{d}, z, \boldsymbol{x}, \theta))_+ = 0$$
$$\theta^N - \delta \Delta\theta^- \le \theta \le \theta^N + \delta \Delta\theta^+, \quad \delta \ge 0$$

5.4. Numerical results

5.4.1. Example problems

Example 4:

Consider the following simple example of two inequality constraints (see Fig. 2) with a single control variable and a single uncertain parameter:

$$g_1 = z^2 - 4z + \theta \le 0$$
$$g_2 = z - \theta \le 0$$
$$0 \le \theta \le 5$$

The numerical results reported in the literature for this problem are the following

Feasibility test:
$\chi(\boldsymbol{d}) = \max\psi(\boldsymbol{d},\theta) = 1.0$, corresponding to $\theta = 5$. This can be easily observed in Fig. 3.

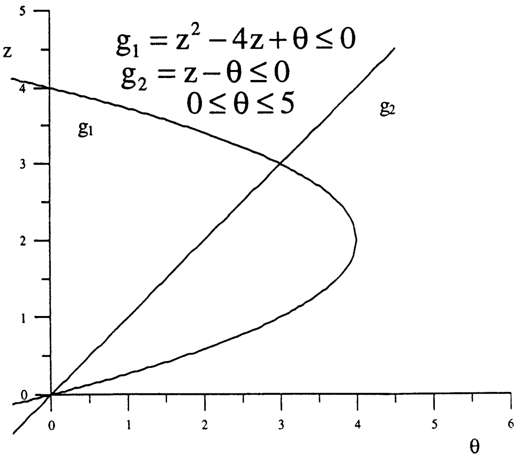


Fig. 2. Plot of the constraints of Example 4 in the space z-θ.

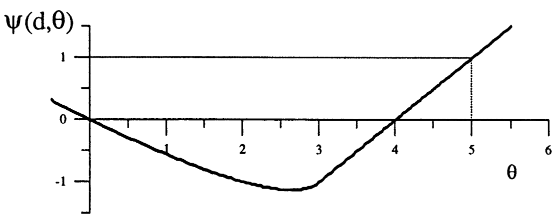


Fig. 3. Plot of the feasibility function for Example 4.

Table 5
CPU time for the flexibility studies of Example 4

	MINLP problem	NLP problem	NLP problem
	Eq. (13)	KSFT	SFFI
Feasibility test	2.56 s	0.57 s	0.61 s
% of reduction	–	77.7	76.2
	Eq. (14)	KSFI	SFFI
Flexibility index	4.17 s	0.58 s	0.57 s
% of reduction	–	86.1	86.3

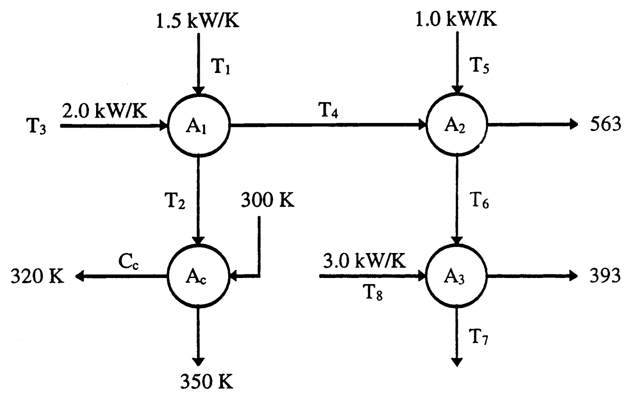
Table 6
Nominal values and range of variations for uncertain parameters of Example 5

Uncertain parameter	Low bound	Nominal value	Upper bound
T_1 (K)	610	650	670
T_3 (K)	370	376	390
T_5 (K)	570	585	593
T_8 (K)	308	310	314
U (kW/m^2 K)	0.38	0.40	0.42

Flexibility index:
For a nominal value of $\theta^N = 2.5$, with $\Delta\theta^- = \Delta\theta^+ = 2.5$. The report flexibility index is $F = 0.6$.

Table 5 presents the reduction in the solution time of this example for both, the flexibility test and the flexibility index using the original formulations from Grossmann and Floudas (1987) and the ones proposed in this work

Example 5:
Consider the following heat exchanger network taken from Grossmann and Floudas (1987). The uncertain parameters are the input temperatures to the network, T_1, T_3, T_5 and T_8 and the heat transfer coefficient U. In Table 6below are the nominal values and maximum variations allowed for these parameters.



The mathematical model for this network is the following:

Energy balances:

$Q_1 = 1.5(T_1 - T_2)$,

$Q_1 = 2.0(T_4 - T_3)$,

$Q_2 = 1.0(T_5 - T_6)$,

$Q_2 = 2.0(563 - T_4)$,

$Q_3 = 1.0(T_6 - T_7)$,

$Q_3 = 3.0\ (393 - T_8)$,

$Q_C = 1.5(T_2 - 350)$,

$Q_C = C_C(320 - 300)$;

Feasibility constraints

$T_2 - T_3 \geq \Delta T_{\min}$,

$T_6 - T_4 \geq \Delta T_{\min}$,

$T_7 - T_8 \geq \Delta T_{\min}$,

$T_6 - 363 \geq \Delta T_{\min}$,

Table 7
CPU for the flexibility studies of example 5

	MINLP problem	NLP problem	NLP problem
	Eq. (13)	KSFT	SFFT
Feasibility test	26.12 s	6.70 s	4.89 s
% of reduction	–	74.3	81.3
	Eq. (14)	KSFI	SFFI
Flexibility index	26.36 s	3.06 s	2.10 s
% of reduction	–	88.4	92.0

$T_7 \geq 323$,

$Q_1 \leq UA_1 \Delta T_{\mathrm{ML1}}$,

$Q_2 \leq UA_2 \Delta T_{\mathrm{ML2}}$,

$Q_3 \leq UA_3 \Delta T_{\mathrm{ML3}}$,

$Q_C \leq UA_C \Delta T_{\mathrm{MLC}}$;

where: $\Delta T_{\mathbf{min}} = 3\mathrm{K}$.

ΔT_{ML} is approximated using the expression provided by Chen (1987).

$$\Delta T_{\mathrm{ML}} = \frac{(T_1 - t_2) - (T_2 - t_1)}{\ln \left\{ \dfrac{T_1 - t_2}{T_2 - t_1} \right\}}$$

$$\cong \left\{ \frac{(T_1 - t_2)\,(T_2 - t_1)\,(T_1 - t_1)\,(T_1 - t_2 + T_2 - t_1)}{2} \right\}^{1/3}$$

The control variable of this problem is the flow rate C_c. The results reported in the literature are:

Feasibility test:
Results:

$\chi(\boldsymbol{d}) = 69.531$

$T_1 = 621.4$ K

$T_3 = 384.7$ K

$T_5 = 570.0$ K

$T_8 = 311.0$ K

$U = 0.38$ (kW/m^2 K)

A positive value for $\chi(\boldsymbol{d})$, means that the operation can not be feasible for the whole range of variation of the uncertain parameters. In order to get feasible operation $\chi(\boldsymbol{d}) \leq 0$ for the given uncertainty, the transfer areas of the heat exchanger must be modified.

Feasibility index:
Results:

$F = 0.117$

$T_1 = 645.3$ K

$T_3 = 375.3$ K

$T_5 = 583.2$ K

$T_8 = 309.8$ K

$U = 0.402$ (kW/m^2 K)

A flexibility index of 0.117 means the given design, accepts only 11.7% of the maximum variation ($\Delta\theta^- \Delta\theta^+$) considered for the uncertain parameters. In this example, the design constraint on heat exchanger C and the operating constraints on the temperature T_7 are the limiting constraints.

Table 7 below, presents the reduction in the CPU times for this example, using as before the original formulation for feasibility test and flexibility index and the two new formulations proposed in this work, based on the KS and smoothing functions.

In order to get good approximations with the KS and smoothing functions in the feasibility test (formulations KSFT and KSFI) and the flexibility index (formulations SFFT and SFFI), large values of ρ and low values of β respectively are required. Considering that a wrong selection of the parameters could give numerical difficulties, we used a strategy consisting in the solution of a series of problems, starting with a given value of the adjusting parameter and doubling o halving it until the final solution was obtained. The optimum of one problem was used as the initial point for the next one. For the formulations KSFT and KSFI, an initial value of $\rho = 2$ was used and 4 iteration were required. For formulation SFFT and SFFI a starting value of β of about 18 was used and 7 iterations were required. The CPU times reported in Table 5 and Table 7correspond to the total CPU time required for all problems.

The formulations based on the KS function (KSFT and KSFI) are of smaller size than the formulations based on the smoothing function (SFFT and SFFI). In our experience, the problems with the KS are easier to solve, in the first place because the size reason, but also because it seems that the smoothing function adds more nonlinearity to the problems than the KS function. The problems with KS are much less sensitive to the initial points. For the examples we solved, the reduction in the CPU time obtained by the two approaches (related to the time of the MINLP formulation), are similar.

Fig. 4 below shows how the algorithms based on the KS function and in the smoothing functions performs for different values of the adjusting parameter ρ, on the optimal value of the feasibility function $\chi(\boldsymbol{d}) = \max \psi(\boldsymbol{d},\theta)$. The correct solution is $\chi(\boldsymbol{d}) = 69.531$. As it can be observed, for values of ρ larger than 1 the correct solution is obtained. The formulation based on the *KS* function gives solutions closer to the correct one of 69.531 for small values of the adjustable parameter.

6. Conclusions

This paper presents new solution strategies for optimization problems under uncertainty, based on the use of a kind of aggregation function, called the KS function, and a type of smoothing function. Specifically, we developed a new solution procedure for the worst case algorithm and the flexibility index and feasibility test problems. We introduced the use of the KS and smoothing functions into the original formulations of these problems leading to an important reduction in the size of the problems, and consequently in the solution time. The proposed algorithms were tested on several examples, showing reduction in the solution time of about 70–90%. In the case of flexibility analysis, the important advantage of the new formulations is that they require the solution of single NLP problems, instead of a multilevel optimization strategy or the solution of a MINLP problem, as in the standard solution procedures.

One limitation of both the KS function and the smoothing functions is that the resulting NLPs are generally nonconvex formulations and these are currently addressed with 'local' NLP solvers. As a result, local solutions can be obtained in the flexibility analysis even if the original feasible region is convex. Addressing this topic will be the subject of future research. Nevertheless, these formulations were very successful in tackling a number of challenging problems in flexibility analysis in a very efficient manner.

Appendix A

To prove properties 1, 2 and 3, we first consider the following lemma:

Lemma:

Consider a set of real valued parameters $\xi_j \geq 0$, the following property holds:

$$\left[\sum_j \xi_j^{\rho_1}\right]^{1/\rho_1} \leq \left[\sum_j \xi_j^{\rho_2}\right]^{1/\rho_2} \quad \text{if } \rho_1 > \rho_2 \text{ and } \rho_1, \rho_2 > 0$$

Proof:

Defining: $\bar{\xi}_j \leq 1$ as

$$\bar{\xi}_j = \frac{\xi_j}{\max_k(\xi_k)} = \frac{\xi_j}{\bar{x}} \in [0, 1]$$

we have $\Sigma_j \bar{\xi}_j^{\rho_1} \leq \Sigma_j \bar{\xi}_j^{\rho_2}$ because $\bar{\xi}_j^{\rho_1} \leq \bar{\xi}_j^{\rho_2}$

and $\left[\Sigma_j \bar{\xi}_j^{\rho_1}\right]^{1/\rho_1} \leq \left[\Sigma_j \bar{\xi}_j^{\rho_2}\right]^{1/\rho_1}$.

Therefore, given that the maximum is equal to 1, we have that

$$\sum_j \bar{\xi}_j \geq 1 \text{ and } \left[\sum_j \bar{\xi}_j^{\rho_2}\right]^{1/\rho_1} \leq \left[\sum_j \bar{\xi}_j^{\rho_2}\right]^{1/\rho_2}$$

Finally, combining the previous deductions it can be concluded that:

$$\left[\sum_j \xi_j^{\rho_1}\right]^{1/\rho_1} = \bar{x}\left[\sum_j \bar{\xi}_j^{\rho_1}\right]^{1/\rho_1} \leq \bar{x}\left[\sum_j \bar{\xi}^{\rho_2}\right]^{1/\rho_2} = \left[\sum_j \xi_j^{\rho_2}\right]^{1/\rho_2}$$

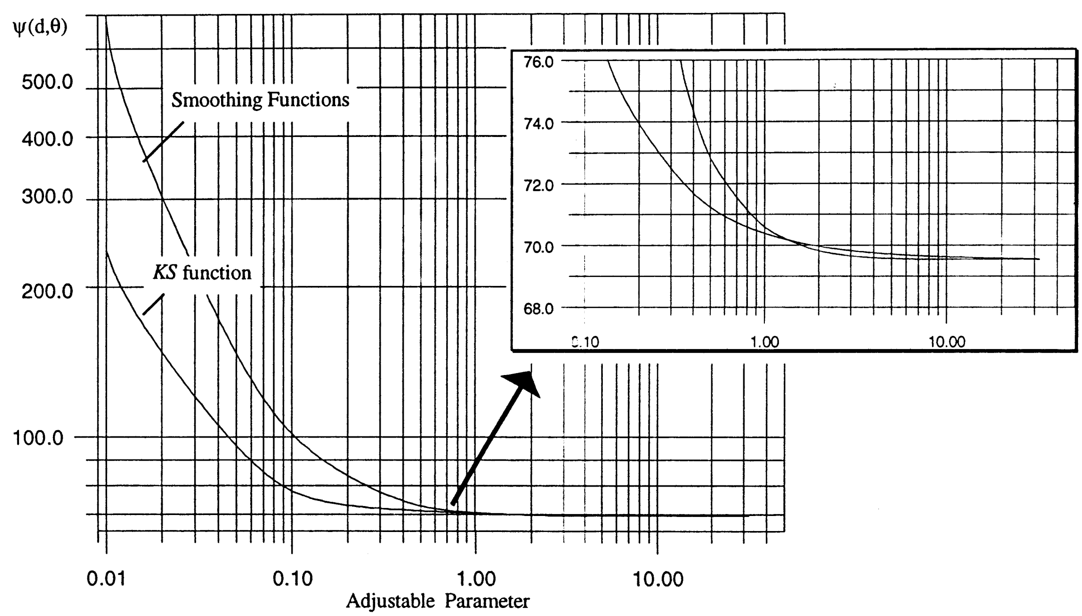


Fig. 4. $\chi^{(d)}$ as function of adjustable parameters.

Property 1:

$\mathrm{KS}(x,\rho) \geq \max_j(g_j(x)),\ \rho > 0$

Proof:

Considering a set of real valued parameters $\xi_j \geq 0, j = 1,\ldots J$, the following relation holds:

$$\sum_k \xi_k^\rho \geq \xi_j^\rho \text{ for every } j\ \xi_j \geq 0$$

or

$$\left[\sum_k \xi_k^\rho\right]^{1/\rho} \geq \xi_j$$

Considering now that we replace parameters ξ_j by real valued functions exp $(g_j(\boldsymbol{x}))$ to be used as the base functions, we have:

$$\left[\sum_k \exp(g_j(\boldsymbol{x}))^\rho\right]^{1/\rho} \geq \exp\,(g_j(x)) \text{ for all } x, j$$

Taking logs of both sides of this expression, we have that:

$$\frac{1}{\rho}\ln\left[\sum_j \exp(g_j(\boldsymbol{x}))^\rho\right] \geq g_j(\boldsymbol{x})$$

and since $\exp(\xi)^\rho = \exp(\rho\xi)$ we can conclude that:

$$\mathrm{KS}(\boldsymbol{x},\rho) = \frac{1}{\rho}\ln\left[\sum_j \exp(\rho g_j(\boldsymbol{x}))\right] \geq g_j(\boldsymbol{x}) \quad \text{for every pair } x, j$$

Property 2:

$$\lim_{\rho\to\infty} \mathrm{KS}(\boldsymbol{x},\rho) = \max_j(g_j(\boldsymbol{x}))$$

Proof:

Note that for $\rho \to \infty$, the ∞-norm of a set of ξ_j ($j = 1,\ldots, J$) can be obtained and the following relation holds:

$$\lim_{\rho\to\infty}\left(\sum_j \xi_j^\rho\right)^{1/\rho} = \max_j \xi_j$$

If now we proceed as before and the real valued parameters ξ_j are replaced by the real valued functions exp $(g_j(\boldsymbol{x}))$, we have:

$$\lim_{\rho\to\infty}\left(\sum_j \exp(\rho g_j(x))\right)^{1/\rho} = \max_j \exp(g_j(x))$$

and then:

$$\lim_{\rho\to\infty}\frac{1}{\rho}\ln\left(\sum_j \exp(\rho g_j(\boldsymbol{x}))\right) = \lim_{\rho\to\infty}\mathrm{KS}(\boldsymbol{x},\rho) = \max_j(g_j(\boldsymbol{x}))$$

Property 3:

$\mathrm{KS}(\boldsymbol{x},\rho_2) \geq \mathrm{KS}(\boldsymbol{x},\rho_1)\forall x$ such that $\rho_1 > \rho_2 > 0$

Proof:

From the Lemma we have:

$$\left[\sum_j \xi_j^{\rho_2}\right]^{1/\rho_2} \geq \left[\sum_j \xi_j^{\rho_1}\right]^{1/\rho_1};\ \rho_1 > \rho_2$$

or

$$\left[\sum_j \exp(\rho_2 g_j(\boldsymbol{x}))\right]^{1/\rho} \geq \left[\sum_j \exp(\rho_1 g_j(\boldsymbol{x}))\right]^{1/\rho_1}$$

and by taking logs of both sides we get:

$$\frac{1}{\rho_2}\ln\left[\sum_j \exp(\rho_2 g_j(\boldsymbol{x}))\right] \geq \frac{1}{\rho_1}\ln\left[\sum_j \exp(\rho_1 g_j(\boldsymbol{x}))\right]$$

which leaves: $\mathrm{KS}(\boldsymbol{x},\rho_2) \geq \mathrm{KS}(\boldsymbol{x},\rho_1)$.

Property 4

The following development demonstrates that the definitions of the KS function in Eqs. (1) and (2) are equivalent. By simple algebraic manipulation of Eq. (2) we get:

$$\rho(\mathrm{KS} - M) = \ln\left[\sum_j \exp\,(\rho(g_j - M))\right]$$

and by log property

$$\exp\,(\rho(\mathrm{KS} - Md)) = \sum_j \exp\,(\rho(g_j - M))$$

or

$$\exp\,(\rho\,\mathrm{KS})\,[\exp\,(\rho M)]^{-1} = [\exp\,(\rho M)]^{-1}\sum_j \exp\,(\rho g_j)$$

Finally, from this last expression we get

$$\mathrm{KS} = \frac{1}{\rho}\ln\left[\sum_j \exp\,(\rho g_j)\right]$$

showing the equivalency of both definition of the KS function

Property 5:

By differentiating Eq. (3) we get:

$$\frac{\partial \mathrm{KS}}{\partial\rho} = \frac{1}{\rho^2}\ln\left[\sum_{\substack{j\\ j\neq \max}} \exp\,(\rho(g_j - g_{\max})) + 1\right] + \frac{1}{\rho}\frac{\sum_{\substack{j\\ j\neq\max}} [(g_j - g_{\max})\exp\,(\rho(g_j - g_{\max}))]}{\sum_{\substack{j\\ j\neq\max}} \exp\,(\rho(g_j - g_{\max})) + 1}$$

Applying limit to this expression for $\rho \to \infty$, we have

$$\lim_{\rho\to\infty}\frac{\partial \mathrm{KS}}{\partial\rho} = 0$$

what means that the KS function becomes insensitive to ρ for a sufficiently large value. Moreover, it can be proved that for sufficiently large ρ, KS → $g_{\max}$. This can be seen by showing that the KS function and its derivatives tend to $g_{\max}$ and its derivative, respectively.

Consider:

$$\frac{\partial \mathrm{KS}}{\partial g_{\max}} = 1 - \sum_{\substack{j \\ j \neq \max}} \exp(\rho(g_j - g_{\max})) \Big/ \left[\sum_{\substack{j \\ j \neq \max}} \exp(\rho(g_j - g_{\max})) + 1 \right]$$

and applying the limit

$$\lim_{\rho \to \infty} \frac{\partial \mathrm{KS}}{\partial g_{\max}} = 1 \quad \Rightarrow \quad \mathrm{KS} = g_{\max}$$

Now by taking derivatives of the KS function with respect to the variables x, it can be seen that for sufficiently large ρ, this derivative tends to the derivative of $g_{\max}$. So, we have:

$$\frac{\partial \mathrm{KS}}{\partial x_i} = \frac{\partial g_{\max}}{\partial x_i} + \frac{\sum_{\substack{j \\ j \neq \max}} \left[\left(\frac{\partial g_j}{\partial x_i} - \frac{\partial g_{\max}}{\partial x_i} \right) \exp(\rho(g_j - g_{\max})) \right]}{\sum_{\substack{j \\ j \neq \max}} \exp(\rho(g_j - g_{\max})) + 1}$$

and applying the limit leads to:

$$\lim_{\rho \to \infty} \frac{\partial \mathrm{KS}}{\partial x_i} = \frac{\partial g_{\max}}{\partial x_i}$$

Property 6:
We now consider the optimality conditions of Problem 2 (P2) and demonstrate by differentiation of ∇L with respect to x the independence of these conditions to $g_{\max}$ and insensitivity to ρ as ρ becomes large. Starting from the Lagrange function for P2, we have:

$$L = f(x) + \mu \left\{ g_{\max} + \frac{1}{\rho} \ln \left[\sum_{\substack{j \\ j \neq \max}} \exp[\rho(g_j(x) - g_{\max})] + 1 \right] \right\}$$

and

$$\nabla L = \nabla f(x) + \mu \nabla \left\{ g_{\max} + \frac{1}{\rho} \ln \left[\sum_{\substack{j \\ j \neq \max}} \exp[\rho(g_j(x) - g_{\max})] + 1 \right] \right\}$$

then:

$$\frac{\partial}{\partial g_{\max}} \nabla L = \mu \frac{\partial}{\partial g_{\max}} \left\{ \frac{\sum_{\substack{j \\ j \neq \max}} [(\nabla g_j - \nabla g_{\max}) \exp(\rho(g_j - g_{\max}))]}{\sum_{\substack{j \\ j \neq \max}} \exp(\rho(g_j - g_{\max})) + 1} \right\} = \mu \frac{(-\rho) \sum_{\substack{j \\ j \neq \max}} [(\nabla g_j - \nabla g_{\max}) \exp(\rho(g_j - g_{\max}))]}{\left[\sum_{\substack{j \\ j \neq \max}} \exp(\rho(g_j - g_{\max})) + 1 \right]}$$

and finally:

$$\lim_{\rho \to \infty} \frac{\partial}{\partial g_{\max}} \nabla L = 0$$

Property 7:
The KS function defined in Eq. (1) is no-convex because the log expression in concave for any positive argument. However, we note that the region defined as

$$\frac{1}{\rho} \ln \left[\sum_{j}^{J} \exp(\rho g_j(x)) \right] \leq C_1$$

is equivalent to the region given by the constraint

$$\Omega(x, \rho) = \sum_{j}^{J} \exp(\rho g_j(x)) \leq \exp(\rho C_1)$$

The Hessian of this constraint is given by

$$\nabla_{xx} \Omega(x, \rho) = \nabla_{xx} \left[\sum_{j}^{J} \exp(\rho g_j(x)) \right] = \sum_{j}^{J} \{ \exp(\rho g_j(x)) \times [\rho \nabla^2 g_j(x) + \rho^2 \nabla g_j(x) \nabla g_j(x)^T] \}$$

which is positive definite if at least one $\nabla^2 g_j(x)$ is positive definite. If the $\nabla^2 g_j(x)$ are all positive semidefinite, then $\nabla_{xx}\Omega(x, \rho)$ is positive semidefinitive. As a results, $\Omega(x, \rho)$ is a convex function and define a convex region. Since this region is equivalent to the one defined by $\mathrm{KS}(x, \rho) \leq C_1$ this region is convex as well, and then providing the property.

Appendix B

Consider the following two set of conditions:

$$\lambda.g(x) = 0 \tag{S1}$$

$$\lambda \geq 0$$

$$g(x) \leq 0$$

$$\lambda - \max(0, \lambda + g(x)) = 0 \tag{S2}$$

There are nine possible combinations of signs of g and λ for set S1 as shown in the second row of the table below. Only cases 1, 2 and 7 correspond to feasible cases. The third row of the table shows the solutions for S2, showing that it has coincident feasible (F) and non feasible (NF) solutions with set S1. Therefore, both sets are equivalent.

1	2	3	4	5	6	7	8	9
$\lambda=0$ $g=0$	$\lambda>0$ $g=0$	$\lambda<0$ $g=0$	$\lambda=0$ $g>0$	$\lambda>0$ $g>0$	$\lambda<0$ $g>0$	$\lambda=0$ $g<0$	$\lambda>0$ $g<0$	$\lambda<0$ $g<0$
$0+0=0$	$\lambda-\lambda=0$	$\lambda-0\neq 0$	$0-g\neq 0$	$\lambda-(\lambda+g)\neq 0$	$\lambda-0\neq 0$ ó $\lambda-(\lambda+g)\neq 0$	$0-0=0$	$\lambda-(\lambda+g)\neq 0$ ó $\lambda+0\neq 0$	$\lambda-0\neq 0$
F	F	NF	NF	NF	NF	F	NF	NF

References

Bahri, P., Bandoni, A., & Romagnoli, J. (1996a). Operability assessment in chemical plants. *Computers & Chemical Engineering*, *20S-B*, S787–S792.

Bahri, P., Bandoni, A., & Romagnoli, J. (1996b). Effect of disturbances in optimising control: The steady-state open-loop back-off problem. *American Institute of Chemical Engineering Journal*, *42*(4), 983–994.

Bahri, P., Bandoni, A., & Romagnoli, J. (1997). Integrated flexibility and controllability analysis in design of chemical processes. *American Institute of Chemical Engineering Journal*, *43*(4), 997–1015.

Bandoni, A. (1987) Programación lineal bajo incertidumbre: Aplicación a una planta productora de jugo concentrado de manzanas. PhD Thesis, Universidad Nacional del Sur.

Bandoni, A., Romagnoli, J., & Barton, G. (1994). On optimising control and the effect of disturbances: calculation of the open-loop back-offs. *Computers & Chemical Engineering*, *18S*, S505–S509.

Chen, J. J. J. (1987). Comments on improvement on a replacement of a logarithmic mean, Letter to the Editors. *Chemical Engineering Science*, *42*, 2488.

Chen, C., & Mangasarian, O. L. (1996). A class of smoothing functions for nonlinear and mixed complementary problems. *Computer Optimimization Applications*, *5*, 97.

Clark, P. (1983). Embedded optimization problems in chemical process design. PhD. Thesis, Carnegie Mellon University, Pittsburgh, USA.

De Hennin, S., & Perkins, J. (1991). Structural decisions in on-line optimisation. Technical Report B93-37, Imperial College, London.

Diaz, S., Raspanti, C., Bandoni, A., & Brignole, E. (1999). Flexibility Study on a Dual Mode Natural Gas Plant in Operation. AIChE Annual Meeting, Dallas, USA, November

Edgar, X.X., & Himmelblau, X.X. (1988). *Optimization of Chemical Processes*, McGraw-Hill.

Figueroa, J., Bahri, P., Bandoni, A., & Romagnoli, J. (1996). Economic impact of disturbances and uncertain parameters in chemical processes: A dynamic back-off analysis. *Computers & Chemical Engineering*, *20*(4), 453–461.

Grossmann, I. E., & Floudas, C. A. (1987). Active Constraint strategy for flexibility analysis in chemical processes. *Computers & Chemical Engineering*, *6*, 675–693.

Grossmann, I. E., & Sargent, R. W. H. (1978). Optimum design of chemical plants with uncertain parameters. *American Institute of Chemical Engineering Journal*, *24*(6), 1021–1028.

Grossmann, I. E., Halemane, K. P., & Swaney, R. E. (1983). Optimization strategies for flexible chemical processes. *Computers & Chemical Engineering*, *7*, 439–462.

Kreisselmeier, G., & Steinhauser, R. (1983). Application of vector performance optimization to a robust control loop design for a fighter aircraft. *International Journal of Control*, *37*, 251–284.

Mayne, D., Michalska, Q., & Polak, E. (1990). An efficient outer approximations algorithm for solving infinite sets of inequalities. *Proceedings of the 29th conference of decision and control*, 960-965.

Ostrovsky, G. M., Volin, Y. M., Barit, E. I., & Senyavin, M. M. (1994). Flexibility analysis and optimization of chemical plants with uncertain parameters. *Computers and Chemical Engineering*, *18*(8), 755–767.

Pistikopoulos, E. (1988). Systematic procedures to improve process flexibility in retrofitting design. PhD Thesis, Carnegie Mellon University.

Polak, E. (1982). An implementable algorithm for the optimal design centering, tolerancing, and tuning problem. *Journal of Optimization Theory and Applications*, *37*, 45–67.

Polak, E., Stimler, D., & Majorisation, M. (1988). A computational complexity reduction technique in control system design. *IEEE Trans. Autom. Control* **AC-31**, 1010-1021.

Sobieszczanski, S. (1992). A technique for locating function roots and for satisfying equality constraints in optimization. *Structural Optimization*, *4*, 241–243.

Sobieszczanski, S., James, B. B., & Riley, M. F. (1987). Structural sizing by generalized multilevel optimization. *American Institute of Aeronautics and Astronautics Journal*, *25*, 139–145.

Sunarto, N., Bandoni, A., Barton, G., & Romagnoli, J. (1994). On optimising control and the effect on disturbances: Application to a gas-tail system. *IChemE Symposium Series No.*, *133*, 159–166.

Swaney, R. E., & Grossmann, I. E. (1985a). An index for operational flexibility in chemical process design. Part I. Formulation and Theory. *American Institute of Chemical Engineering Journal*, *31*, 621–630.

Swaney, R. E., & Grossmann, I. E. (1985b). An index for operational flexibility in chemical process design. Part II. Computatonal algorithms. *American Institute of Chemical Engineering Journal*, *31*, 631–641.

Tits, A. L. (1985). On the optimal design centering, tolerancing, and tuning problem. *Journal of Optimization Theory and Applications*, *45*, 487–494.

Walsh, S., & Perkins, J. (1996). Operability and control in process synthesis and design. *Advances in Chemical Engineering*, *23*, 301–402.