International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 3 (2011) pp.051-060

Interior Point Method based Sequential Quadratic Programming Algorithm with Quadaratic Search for Nonlinear Optimization

Hassan A. Bashir[1] and Ximing Liang[2]

[1]Department of Computer Application Technology, School of Information Science and Engineering, Central South University
Changsha, 410083, China
habash1600@gmail.com
(Correspondence Author)

[2]Department of Automation, School of Information Science and Engineering, Central South University
Changsha, 410083, China
ananxml@mail.csu.edu.cn

***Abstract*: The field of constrained nonlinear programming (NLP) has been principally challenging to various gradient based optimization techniques. The Sequential quadratic programming algorithm (SQP) that uses active set strategy in solving quadratic programming (QP) subproblems proves to be efficient in locating the points of local optima. However, its efficient determination of the optimal active set heavily relies on the initial guess of the starting point. This remains a serious drawback to both primal and dual active set (AS) approaches especially for NLPs with several inequality constraints. Thus, we propose an SQP/IIPM algorithm that uses infeasible interior point method (IIPM) for solving QP subproblems. In this approach inequality constraints can be solved directly, alleviating the burden for choosing a feasible starting point necessary for efficient convergence to optimal active set. At every iteration *k*, we evaluate step length adaptively via a simple line search and/or a quadratic search algorithm depending on the reason for the termination of our IIPM QP solver. Our SQP/IIPM algorithm falls to line search whenever the termination of the IIPM QP solver satisfies the complementary slackness condition (i.e. duality measure). This means that neither the Primal nor the Dual feasibility conditions are satisfied during the termination of the IIPM QP solver. In addition, the algorithm can also switch to quadratic search whenever the line search algorithm exceeds its maximum iteration limit. Results of comparing SQP/IIPM with SQP reveal that, the proposed algorithm is efficient and promising.**

Keywords: Sequential quadratic programming, Active set method, Infeasible interior point method, Quadratic search, Quadratic programming subproblem

I. INTRODUCTION

Formally, for any given system, optimization techniques are used to find a set of design parameters $x_i = \{x_1, \ldots, x_n\}$ that can lead the system to its optimal conditions. In a more advanced formulation, the objective function $f(x)$ to be minimized or maximized, might be subject to a set linear and/or nonlinear equality constraints $\{E\}$, inequality constraints $\{I\}$ and/or parameter bounds. A general optimization problem is depicted in system (1). The solution requires evaluation of the optimal objective function for the given set of design variables while satisfying all the constraints [1]. In this paper, we assume both the objective function and the constraints to be differentiable functions.

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t} \quad & h_i(x) = 0, \quad i \in E \\ & g_i(x) \le 0, \quad i \in I \\ & x \in \Re^{n\times 1} \end{aligned} \tag{1}$$

where

f	=	Objective function
h	=	Vector of equality constraints
g	=	Vector of inequality constraints
$x \in \Re^{n\times 1}$	=	Vector of design variables

SQP methods were first proposed in 1963 by Wilson [2] and were developed by Garcia-Palomares and Mangasarian [3], Ju-Liang and Xiang-Sun [4] and Powell [5, 6]. The successive linear programming approach is described by Fletcher [7]. The relation between SQP and augmented Lagrangian methods is explored by Melara [8]. The essential idea of SQP is to model system (1) for the current iteration point x^k by a quadratic programming subproblem and to use the minimizer of this subproblem to determine the next iteration point x^{k+1}. The challenge is to design a quadratic subproblem that will yield a reasonable step for the underlying constrained optimization problem so that the overall SQP algorithm has good convergence properties and good practical performance.

Perhaps the simplest derivation of SQP methods views them as an application of Newton's method to the Karush-Kuhn-Tucker (KKT) optimality conditions for (1).

The standard SQP algorithm is based on the equality constrained QP (EQP) technique and therefore uses active set strategy to successively solve the general NLP (1) by considering only those constraints that are active at the current iteration. The main drawback of these EQP methods is obviously in their inability to handle both equality and inequality constraints concurrently. Further, one common difficulty associated with all the line search implementations of SQP is that, the linearization of nonlinear constraints may give rise to an infeasible subproblem. A number of work-around methods have been proposed to overcome this problem, one key technique propose replacing the QP subproblem with an auxiliary problem whenever infeasibility exist in the obtained quadratic subproblem [9].

An alternative approach for solving the QP subproblem is to use an interior point method. For general problems, this approach is competitive with active set methods especially in early iterations when the active sets change substantially from iteration to iteration and not much is to be gained from hot-start information, which according to Wright [10], allows one to initialize the working set for each QP subproblem to be the final active set from the previous SQP iteration. Moreover, interior point methods move through the interior of the feasible region making smooth approximations. So they ignore much of the problem's combinatorial structure and look at the analytic structure. Therefore, on some problems with special structure (for example, certain applications in control engineering) interior point methods are better able to exploit the structure than active set methods, and therefore become very competitive [11]. Although interior point methods hardly get an exact solution without a special rounding procedure, they get a very good approximation of the optimum point at an incredibly short time. Interior point methods solve a very small number of systems of equations which are relatively complex. For a more comprehensive survey on all SQP algorithms, see the work of Boggs and Tolle [12].

Our motivation follows the fact that, solving QP subproblems via active set strategy remains crucial as the method suffers reduced efficiency in determining the optimal active set from an arbitrarily chosen starting point especially for NLPs possessing several inequality constraints. Thus, we propose solving QP subproblems via IIPM algorithm which proves to be resilient to the presence of several inequality constraints in NLPs [13].

After the introduction to constrained NLPs in section one, we will review the literature behind SQP, active set method and merit functions in section two. Section three presents the design of SQP/IIPM describing in detail an IIPM algorithm for solving QP subproblems, line search and quadratic search procedures for minimizing the merit function, damped BFGS Hessian update procedure for approximation of the Hessian matrix and a complete routine summarizing the SQP/IIPM algorithm. In section four, twenty five benchmark numerical problems are used for performance assessment. The numerical performance of the proposed SQP/IIPM algorithm is compared against that of the standard SQP algorithm. Finally, discussions of test results and conclusions are drawn in section five.

II. THE SQP APPROACH

SQP algorithm is a famous direct method for solving constrained NLPs and has a long history dating back to the papers of Fiacco and McCormic 1968 [14] and Fletcher [15,16]. The method iteratively derives search directions at every iteration k by solving the following QP subproblems (2) obtained through linearization of (1) based on Taylor's approximation [14,15].

$$\begin{aligned} &\min \nabla f(x^k)^T d + \tfrac{1}{2} d^T H d \\ &\text{s. t}\;\; h_i(x^k) + \nabla h_i(x^k)^T d = 0; \quad i \in E \\ &\qquad g_i(x^k) + \nabla g_i(x^k)^T d \le 0; \quad i \in I \end{aligned} \tag{2}$$

where $d \in \Re^{n\times 1} = x - x^k$ is the new system variable, $H \in \Re^{n\times n}$ is the Hessian matrix.

Each QP subproblem minimizes a quadratic model of a certain Lagrangian function subject to linearized constraints. A merit function is reduced along each search direction to ensure convergence from any starting point, thus, yielding a method having major (SQP) and minor (QP) iterations. The major iteration yields the points (x^k, u^k, v^k) that converge to (x^*, u^*, v^*) at the optimal solution. At each minor iteration, a QP subproblem is minimized to generate a search direction towards the next points $(x^{k+1}, u^{k+1}, v^{k+1})$. Solving such a subproblem is itself an iterative procedure. An extended bibliography on large-scale NLP solvers can be found in [17].

A. Active set method for SQP

The standard SQP algorithm repeatedly solves QP subproblems (2) via active set procedure. The QP is a constrained optimization problem with a quadratic objective function and linear constraints. Primal active set method solves a convex QP in which Hessian matrix H is positive semidefinite, whereas dual active set method requires the QP to be strictly positive [17]. The method begins by making an initial guess of optimal active set. Usually, if this guess turns out to be incorrect, it iteratively uses gradient and Lagrangian multiplier information to swap indices in and out of the current active set. The more inequality constraints the QP possesses, the more iterations are needed to achieve the optimal active set. In this way, active set method solves an inequality constrained QP as a sequence of corresponding equality constrained QP problems [18]. Hence, the strategy remains computationally expensive, exhibiting poor efficiency and reduced performance for NLPs having several inequality constraints.

Although active set method took advantage of prior information of the previous solution to avoid complete factorization, a phenomenon known as "hot-start", it however lacks the ability to exploit the problem's special structures which is very essential in dealing with control engineering problems [19].

B. The choice of merit function

Following our proposal to employ a technique of adaptive selection between line search and quadratic search algorithms to evaluate the step length parameter α, we will employ two different merit functions that will suite the two different minimization algorithms. Sequel to the non-differentiability of merit function, its minimum cannot be determined through the conventional gradient methods. Thus, we propose evaluating it via either line search and/or quadratic search algorithms elaborated in section III. In essence, SQP/IIPM algorithm defaults to quadratic search and only switches to line search when the QP obtained is not unbounded and satisfy the complementary condition (section III) at termination. Hence, sensitivity to the choice of the initial solution point is significantly reduced and the overall stability is improved.

***1)** Merit function for Line Search Algorithm:* For the line search algorithm, we will use the following merit function:

$$\varphi(\alpha) = f(x^k + \alpha d) + \mu V(x^k + \alpha d) \tag{3}$$

where f is the objective function, μ is a scalar penalty parameter and V is a measure of constraint violation. For the initial iteration, at $k = 0$, let v_i^0 and u_i^0 respectively be the multipliers for the equality/inequality constraints for the QP subproblem, then

$$\mu^0 \geq \text{Max}\left[1, \quad \sum_{i \in E} Abs[v_i^0] + \sum_{i \in I} u_i^0\right]. \tag{4}$$

For iterations $k > 0$, μ will always be updated in terms of its previous values as in the following manner:

$$\mu^{k+1} = \text{Max}\left[\mu^k, \quad 2\left(\sum_{i \in E} Abs[v_i^k] + \sum_{i \in I} u_i^k\right)\right] \tag{5}$$

$$V(x^k) = \text{Max}\left[0, \quad \{Abs[h_i(x^k)], i \in E\}, \quad \{g_i(x^k), i \in I\}\right].$$

2) Merit function for Quadratic Search Algorithm: This is based on the work of Biegler et al [20] and Bhatti [21], in the situation where SQP/IIPM switches to quadratic search, we will employ the following merit function:

$$\varphi(x) = f(x) + \sum_{i \in I} \bar{u}_i \max[0, \ g_i] + \sum_{i \in E} \bar{v}_i \text{abs}[\text{h}_i] \tag{6}$$

where $\bar{u}_i$ and $\bar{v}_i$ are Lagrangian multipliers. For the initial iteration, i.e. at $k = 0$, $\bar{u}_i^0$ and $\bar{v}_i^0$ are the exact Lagrangian multipliers u_i and v_i obtained from the solution of the original QP subproblem. For subsequent iterations, $\bar{u}_i^k$ and $\bar{v}_i^k$ are updated in terms of their previous values $\bar{u}_i^{k-1}$ and $\bar{v}_i^{k-1}$ as follows

$$\begin{aligned} \bar{u}_i^k &= \max\left[u_i, \ \frac{1}{2}(\bar{u}_i^{k-1} + u_i)\right], \ i \in I \\ \bar{v}_i^k &= \max\left[\text{abs}[v_i], \ \frac{1}{2}(\bar{v}_i^{k-1} + \text{abs}[v_i])\right], \ i \in E. \end{aligned} \tag{7}$$

The search direction d^k derived from (2) together with a step length α_k obtained by minimizing merit function will be used in estimation of the next iteration $x^{k+1} = x^k + \alpha_k d^k$.

III. The Proposed SQP/IIPM Algorithm

The following is a simple flowchart of the proposed SQP/IIPM algorithm.

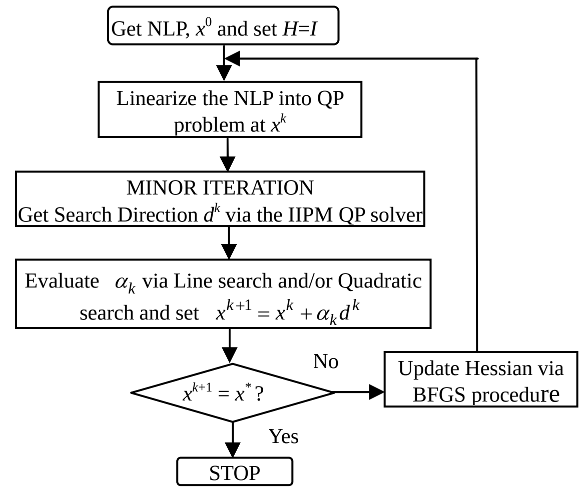

Figure 1. Flowchart of the Proposed SQP/IIPM Algorithm

This paper adopts an infeasible interior point method (IIPM) for solving the QP subproblem. IIPM originated as a result of researches made towards developing efficient techniques for solving linear programming problems. Interior point method emerged in the late 1950s, but in the last two decades, after the work of Karmarkar's [22, 23, 24], the method prove to clearly outperform the well known simplex algorithm in solving large scale linear problems. It can also be applied in dealing with nonlinear problems. The acronym 'infeasible' qualifies the algorithm to begin its search from an infeasible starting point and work towards feasibility and optimality by passing through the interior of the feasible region [21].

Amongst all various kinds of interior point methods, a key feature of IIPM is its flexibility in the choice of the starting point. In this regard, the starting solution needs not to satisfy all the constraints except for the positivity requirements. The solution point becomes feasible as the algorithm progresses. Thus the method features fast convergence and ability to adequately handle inequality constraints.

A. IIPM for QP Subproblem

In this section we present an IIPM algorithm responsible for solving QP subproblems which lie at the heart (core) and constitutes the inner loop of the main SQP/IIPM algorithm. This entails establishing the idea of IIPM for quadratic programming problem that has a polynomial complexity and has its origin from IIPM for linear programming [25]. A general quadratic programming problem is as follows.

$$\begin{aligned} \min \quad & c^T x + \tfrac{1}{2} x^T Q x \\ \text{s.t.:} \quad & Ax = b, \quad x \geq 0 \end{aligned} \tag{8}$$

where $c \in \Re^{n \times 1}, A \in \Re^{m \times n}, b \in \Re^{m \times 1}$ and $Q \in \Re^{n \times n}$ is symmetric matrix. The system is convex if at least Q is positive semidefinite. To establish the corresponding dual problem, according to the Karush-Kuhn-Tucker (KKT) conditions, the Lagrangian of the primal problem is:

$$\mathrm{L} = c^T x + (1/2)x^T Qx + u^T(-x + s^2) + v^T(-Ax + b) \qquad (9)$$

where u is an $n\times1$ vector of multipliers associated with positive constraints, s is a vector of slack variables and v is an $m\times1$ vector of multipliers associated with equality constraints. Differentiating (8) with respect to its variables permits the application of the concept of Lagrangian duality to determine the corresponding dual problem as follows:

$$\frac{\partial \mathrm{L}}{\partial x} = c + Qx - u - A^T v = 0 \qquad (10a)$$

$$\frac{\partial \mathrm{L}}{\partial v} = -Ax + b = 0 \qquad (10b)$$

$$\frac{\partial \mathrm{L}}{\partial u} = -x + s^2 = 0 \qquad (10c)$$

$$\frac{\partial \mathrm{L}}{\partial s} = 2u_i s_i = 0, \quad or \quad u_i s_i = 0 \qquad (10d)$$

$$u_i \geq 0, \quad i = 1, ..., n.$$

The two nonlinear equations (10c and 10d), can be taken care of by eliminating s when $u_i s_i = 0$. That is, either $u_i = 0$ (at $-x_i < 0$) or $s_i = 0$ (at $x_i = 0$). Hence by keeping x_i positive, the optimum of the QP can be obtained.

$$\begin{aligned} &Ax - b = 0 \\ &-Qx + A^T v + u = c \\ &XUe = 0,\ (x_i, u_i) \geq 0,\ i = 1,.., n \text{ and } e^T = [1,..,1] \\ &\text{where } X = \operatorname{diag}[x_i] \text{ and } U = \operatorname{diag}[u_i] \end{aligned} \qquad (11)$$

For the sake of simplicity, X and U will be used throughout this paper to stand for X^k and U^k for iteration k. In the following, we derive the dual problem.

$$M(u, v) = \min_x \begin{bmatrix} c^T x + \frac{1}{2} x^T Qx - u^T x \\ + v^T(-Ax + b) \end{bmatrix} \qquad (12)$$

$$u_i \geq 0,\ i = 1, ..., n$$

Differentiating the above dual with respect to x gives $c + Qx - A^T v - u = 0,$ multiplying by x and substituting the result into (12) yields the following expression. $M(u, v) = -(1/2)x^T Qx + v^T b,\ u_i \geq 0,\ i = 1, .., n.$

Therefore, the complete dual problem is:

$$\begin{aligned} &\max \ -(1/2)x^T Qx + v^T b \\ &\text{s.t.} \quad Qx + c - u - A^T v = 0 \\ &\qquad u_i \geq 0, \quad i = 1,.., n \end{aligned} \qquad (13)$$

This leads us to defining the feasibility equations necessary for the optimum as:

$$\begin{aligned} &r_p = -Ax^k + b = 0 && \text{Primal} \\ &r_d = Qx^k - A^T v^k - u^k + c = 0 && \text{Dual} \\ &r_c = -XUe + \mu_k e = 0 && \text{Complementarity} \\ &(x_i^k, u_i^k, v_i^k) \geq 0; \quad i = 1,.., n \end{aligned} \qquad (14)$$

1) Direction of Descent: Descent directions can be derived by disturbing (14) at (x^k, u^k, v^k) by (d_x, d_u, d_v).

$$\begin{pmatrix} A & 0 & 0 \\ -Q & I & A^T \\ U & X & 0 \end{pmatrix} \begin{pmatrix} d_x \\ d_u \\ d_v \end{pmatrix} = -\begin{pmatrix} Ax^k - b \\ -Qx^k + A^T v^k + u^k - c \\ XUe - \mu_k e \end{pmatrix}$$

$$Ad_x = -Ax^k + b \equiv r_p \qquad (15a)$$

$$-Qd_x + d_u + A^T d_v \equiv r_d \qquad (15b)$$

$$Ud_x + Xd_u = -XUe + \mu_k e \equiv r_c \qquad (15c)$$

Multiplying (15b) by X and substituting for Xd_u as in (15c) yields $-d_x + [XQ + U]^{-1} XA^T d_v = [XQ + U]^{-1}(Xr_d - r_c)$. Multi plying by A and solving for d_v permits evaluation of the descent directions as follows.

$$\begin{aligned} &d_v = [A[XQ + U]^{-1} XA^T]^{-1}(r_p + A[XQ + U]^{-1}(Xr_d + r_c)) \\ &d_x = [XQ + U]^{-1}(XA^T d_v - Xr_d + r_c) \\ &d_u = X^{-1}(r_c - Ud_x) \end{aligned} \qquad (16)$$

2) Step length and Next point: Taking the longest possible step along the obtained descent direction can allow measuring its effectiveness. This is the maximum step before violating the non-negativity conditions $(x, u) \geq 0$ with an upper bound of 1. Assuming positive initial values and introducing the step length parameters α_p^k and α_d^k, we define the next iteration as:

$$\begin{aligned} &x^{k+1} = x^k + \alpha_p^k d_x \\ &u^{k+1} = u^k + \alpha_d^k d_u \\ &v^{k+1} = v^k + \alpha_d^k d_v \end{aligned} \qquad (17)$$

Obviously, the maximum step length that will make one of the x_i^k, u_i^k or v_i^k values go to zero is at $x_i^k + \alpha_p^k d_{xi} \geq 0,$ $u_i^k + \alpha_d^k d_{ui} \geq 0$ and $v_i^k + \alpha_d^k d_{vi} \geq 0$. Thus, α_d^k and α_p^k can explicitly be evaluated as follows.

$$\begin{aligned} &\alpha_p^k = \beta \times \min\left[1, \ -x_i^k/d_{xi}; \quad d_{xi} < 0\right] \\ &\alpha_d^k = \beta \times \min\left[1, \ -u_i^k/d_{ui}; \quad d_{ui} < 0\right] \end{aligned} \qquad (18)$$

where β is a step length factor (0.9, 1], it assures feasibility by slightly reducing α_p^k and α_d^k.

3) Termination of IIPM Algorithm: Termination is subject to the following criteria.

a) Primal/Dual Feasiblity: Satisfying the feasibility requirements is an indication of the attainment of the optimum solution. This holds true when primal and/or dual feasibility equations respectively become $Ax - b = 0$ and

$-Qx + A^T v^k + u^k - c = 0$. The errors in satisfying these equations can be formulated as follows.

$$\sigma_p = \frac{\| Ax - b \|}{\| b \| + 1} \le \varepsilon 1 \text{ and } \sigma_d = \frac{\| r_d \|}{\| Qx^k + c \| + 1} \le \varepsilon 2 \qquad (19)$$

where $\varepsilon 1$ and $\varepsilon 2$ are small positive tolerances.

b) Complementary Slackness condition: The error μ_k in satisfying the complementary condition popularly known as duality measure is empirically formulated as $\mu_k = (x^k)^T u^k / n \le \varepsilon 3,$ where n is the number of variables and $\varepsilon 3$ is a small positive tolerance [21].

4) Summary of the IIPM Routine: Given the starting parameters: x^0, u^0, v^0, QPMaxIter and the matrices Q and A, together with vectors c and b, set an iteration counter $k = 0$.

Routine A. IIPM Algorithm

STEP 1: Check for convergence at the current point x^k

$$\text{If} \left\{ \begin{array}{l} \dfrac{\| Ax^k - b \|}{\| b \| + 1} \le \varepsilon 1, \quad \dfrac{\| r_d \|}{\| Qx^k + c \| + 1} \le \varepsilon 2, \\ \mu_k = \dfrac{(x^k)^T u^k}{n} \le \varepsilon 3 \ \text{ or } k = \text{QPMaxIter} \end{array} \right\} \text{Goto } \textbf{STEP 6}$$

Else proceed to **STEP 2**. Endif

STEP 2: Evaluate r_p, r_d, r_c from (14).

STEP 3: Solve for d_x, d_u, d_v from (16).

STEP 4: Compute the step length α_p^k and α_d^k (18).

STEP 5: Compute the next point (17), set $k = k + 1$, go to **STEP1**.

STEP 6: Halt and set the optimum point $x^* = x^k$.

B. Line Search and Quadratic Search Algorithms for Minimizing Merit Function

1) Line Search Algorithm: In principle, as highlighted earlier, merit functions are not easily differentiable functions, and can only be minimized via numerical search algorithms. In the following, we will describe the line search algorithm for estimating a step length parameter α . For a given merit function $\varphi(\alpha) = f(x^k + \alpha d^k)$, a line search algorithm based on Armijo's rule will be employed [21, 26].

Definitions: Let d be the search direction, α be a step length parameter and γ the control parameter, typically $\gamma = 0.5$. The line search algorithm is depicted in the following routine.

Routine B. Line Search Algorithm

Set $\alpha = 0$ and compute the corresponding value of the merit function $\varphi(0)$. Set an iteration counter $i = 1$, and $\alpha = 1$

WHILE $\varphi(\alpha) > \varphi(0) - \alpha\gamma \| d \|^2$ AND $i \le$ LSMaxIter

Set $\alpha = \left(1/2 \right)^i$,

Evaluate $\varphi(\alpha)$

Increment the iteration counter, i.e. set $i = i + 1$

END WHILE

Return α as the step length parameter.

As can be seen from the above procedure, any value of α that reduces the merit function with a factor of $\alpha\gamma \| d \|^2$ can provide a reasonable estimate for the step length parameter α . Inclusion of α in this factor means that, as the trial step length reduces, the reduction obtained in the descent direction becomes smaller. Therefore, as long as d is a descent direction, the test must pass after a finite number of iterations.

2) Quadratic Search Algorithm: In order to achieve effective and robust minimization of the merit function, the proposed SQP/IIPM algorithm is designed to use both line search and quadratic search algorithms. Quadratic search uses an interpolation technique to fit a quadratic function through a given set of three points. The minimum of this quadratic function is computed using necessary conditions. A new set of three points is selected by comparing function values at this minimum point with the given three points. The process is repeated with the three new points until the interval in which the minimum lies becomes fairly small. As the interval becomes small, the quadratic approximation becomes closer to the actual function. Thus, the method requires only one new function evaluation at every iteration and also takes into account the nature of the function itself besides the given search location.

The proposed quadratic search algorithm comprises of three main parts, namely;

- Approximation of the merit function with a quadratic model.
- A procedure to evaluate the starting points, and
- Minimization of the quadratic model.

a) The Quadratic Model: For any given three points, a_l, a_m, a_u, $a_l \ne a_m \ne a_u$, a quadratic model derived via interpolation technique can be set to fit the function $\varphi(x^k + ad^k)$ as follows.

$$\begin{aligned} \phi_q(a) = {} & \phi_l \frac{(a - a_m)(a - a_u)}{(a_l - a_m)(a_l - a_u)} + \phi_m \frac{(a - a_l)(a - a_u)}{(a_m - a_l)(a_m - a_u)} \\ & + \phi_u \frac{(a - a_l)(a - a_m)}{(a_u - a_l)(a_u - a_m)} \end{aligned} \qquad (20)$$

where $\varphi_l = \varphi(x^k + a_l d^k)$, $\varphi_m = \varphi(x^k + a_m d^k)$ and $\varphi_u = \varphi(x^k + a_u d^k)$. Differentiating (20) with respect to a yields:

$$a_q = \frac{1}{2} \left(\frac{\phi_l (a_m^2 - a_u^2) + \phi_m (a_u^2 - a_l^2) + \phi_u (a_l^2 - a_m^2)}{\phi_l (a_m - a_u) + \phi_m (a_u - a_l) + \phi_u (a_l - a_m)} \right) \qquad (21)$$

b) Evaluation of the Starting Point: Arbitrary selection of starting points often lead to division by zero in (21), thus for any starting points, the following condition must be satisfied.

$$\frac{\phi_l + \phi_u}{2} > \phi_m$$

Based on this condition, the following procedure is proposed to obtain an appropriate starting point.

Routine C. Begin with an initial step $\delta > 0, a_l = 0$, evaluate $\varphi(a_l)$, set $a_1 = \delta$ and compute $\varphi(a_1)$.

A- CASE $\varphi(a_1) > \varphi(a_l)$, minimum lies at a point less than a_1. Set $a_u = a_1$ and $\varphi(a_u) = \varphi(a_1)$, set $a_m = \delta/2$ and compute $\varphi(a_m)$, go to **E**.
B- CASE $\varphi(a_1) \le \varphi(a_l)$, the minimum is not bracketed and a_1 becomes the middle point.
Set $a_m = a_1$ and $\varphi(a_m) = \varphi(a_1)$, set $a_2 = 2\delta$ and evaluate $\varphi(a_2)$, go to **C**.
C- IF $\varphi(a_2) > \varphi(a_1)$, the 3rd point is located, set $a_u = a_2$ and $\varphi(a_u) = \varphi(a_2)$, go to **E**.
D- IF $\varphi(a_2) < \varphi(a_1)$, the minimum is not bracketed.
Set $a_1 = a_2, \varphi(a_1) = \varphi(a_2)$ and $\delta = 2\delta$, loop back to A.
E- IF $\varphi_l + \varphi_u > 2\varphi_m$, stop, else set $\delta = 2\delta, a_1 = \delta$, compute $\varphi(a_1)$ and loop back to **A**.

c) Minimization of the Quadratic Model: In this section, we present a routine that will minimize the developed quadratic model based on the starting points obtained form Routine C above. After determining the minimum point a_q via minimizing system (21), the next task is to determine which one of the three given points (a_l, a_m, a_u) to replace with a_q. Having a_q as the minimum point, the relation between a_q and the initially assumed minimum point a_m is either of the two cases depicted in figure 2 where a_q either lies between $(a_l \text{ and } a_m)$ or $(a_m \text{ and } a_u)$. This figure illustrates four possible situations for the two cases.

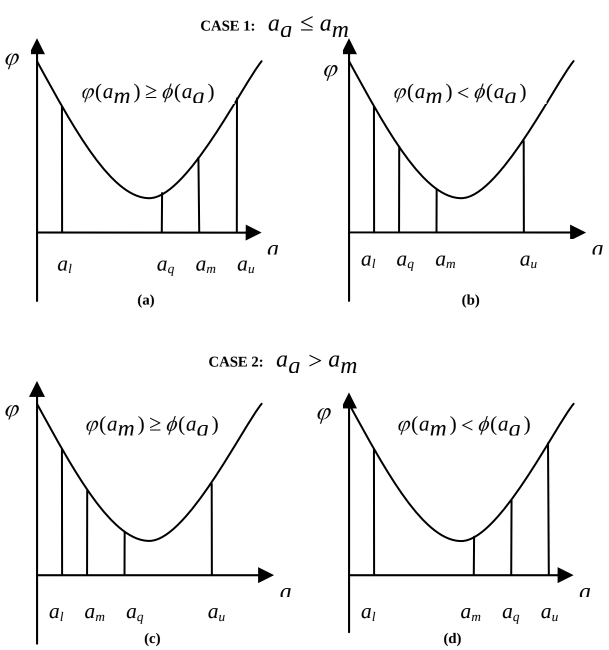

Figure 2. Four Cases in Quadratic Search Algorithm

With reference to the above four cases, the following routine presents the entire processes involved in our proposed quadratic search algorithm.

Routine D. The quadratic model is minimized using the starting points obtained from Routine C.

STEP 1
CASE $a_q \le a_m$
If $\varphi(a_m) \ge \varphi(a_q)$, the minimum lies within (a_l, a_m), the next point is (a_l, a_q, a_m).
If $\varphi(a_m) < \varphi(a_q)$, the minimum lies within (a_q, a_u), the next point is (a_q, a_m, a_u).
Continue to **STEP 2**
CASE $a_q > a_m$
If $\varphi(a_m) \ge \varphi(a_q)$, the minimum lies within (a_m, a_u), the next point is (a_m, a_q, a_u).
If $\varphi(a_m) < \varphi(a_q)$, the minimum lies within (a_l, a_q), the next point is (a_l, a_m, a_q).
Continue to **STEP 2**
STEP 2
Convergence check: Let QSTol be a tolerance parameter and QSMaxIter be a maximum iteration limit.
If $\text{abs}[(\varphi_q(a_q) - \varphi(a_q))/\varphi(a_q)] \le Tol$,
or $k = \text{QSMaxIter}$
Stop and set the step length $\alpha = a_m$.
Else loop back to **STEP 1**.
Endif.

C. Hessian Update for QP Subproblem

It follows that the evaluation of exact Hessian often results in an indefinite matrix. A vast majority of researchers advocate approximation of the Hessian and updating it via BFGS update procedure [19]. Using Quasi-Newton approximation, a positive definite *H* can effectively be approximated at iteration *k* via BFGS update procedure. At $k = 0$, H^0 is set to identity matrix *I*, for iteration *k*, H^k is calculated as follows:

Routine E. The Hessian Update Procedure

Define vectors: $s^k = x^{k+1} - x^k$ and $q^k = \nabla_x L^{k+1} - \nabla_x L^k$

Let the parameter $\gamma = \theta q^k + (1-\theta) H^k s^k$

Where

$$\theta = \begin{pmatrix} 1; & (q^k)^T s^k \geq 0.2 (s^k)^T H^k s^k \\ \dfrac{0.8 (s^k)^T H^k s^k}{(s^k)^T H^k s^k - (q^k)^T s^k}; & (q^k)^T s^k < 0.2 (s^k)^T H^k s^k \end{pmatrix}$$

If $(q^k)^T s^k > 0$, i.e. all the eigenvalues of $\nabla_{xx}^2 L$ are positive, the Hessian is updated as:

$$H^{k+1} = H^k + \frac{\gamma \gamma^T}{(q^k)^T s^k} - \frac{H^k s^k (s^k)^T H^k}{(s^k)^T H^k s^k}$$

Else, previous value of H is retained.
End
If (all eigenvalues of H) > 0
H is positive definite, keep H
Else
H is not positive definite, reset *H* to *I*, i.e. set $H = I$
End.

D. Termination of SQP/IIPM Algorithm

Termination of the major loop of SQP/IIPM algorithm is basically subject to the following three conditions:

i. SQP/IIPM algorithm can terminates if the norm of the search direction obtained by solving the QP subproblem via the IIPM Routine A is less than a small positive tolerance. In other words, when $\| d^k \| < \mathrm{Tol1}$.
ii. Secondly, SQP/IIPM can also be terminated when the difference between the norms of successive gradients of the Lagrangian function is less than a user specified tolerance parameter, i.e. $\| \nabla_x L^{k-1} \| - \| \nabla_x L^k \| < \mathrm{Tol2}$.
iii. Finally, the algorithm will be terminated if a user specified maximum iteration limit is reached, i.e. when $k = \mathrm{SQPMaxIter}$

E. Summary of SQP/IIPM Algorithm

In the following we present a summary of the proposed SQP/IIPM algorithm. For concise definitions of the tolerance parameters and maximum iteration limits used in this routine, please refer to the parameter chart in table II.

Routine F. The complete SQP/IIPM Algorithm

Given any starting points x^0, the tolerance parameters and maximum iteration limits are as depicted in table II. Set an iteration counter $k = 0$ and $H^0 = I$.

i. Linearize (1) into a QP subproblem (2) at the current point x^k.
ii. Set QSoption = 0
iii. Call the IIPM routine A to evaluate d^k by minimizing the QP (2).
iv. *if* QSoption = 0 OR $\mu_k = \dfrac{(x^k)^T u^k}{n} \leq \varepsilon 3$, i.e. QP termination satisfies the complementary slackness condition. Set a counter $i = 1$.
if $k = 0$
Pass the merit function (3), d^0 and the penalty parameter μ^0 obtained from (4) into the line search routine B, compute α_0.
if $i >$ LSMaxIter
Set QSoption = 1
endif
else
Pass the merit function (3), d^k and the penalty parameter μ^{k+1} obtained from (5) into the line search routine B, compute α_k.
if $i >$ LSMaxIter
Set QSoption = 1
endif
endif
elseif QSoption = 1
if $k = 0$
Pass the merit function (6), d^0 and $\bar{u}_i^0$ and $\bar{v}_i^0$ obtained from routine A into the quadratic search routine D and compute α_0
else
Pass merit function (6), d^k and the updated $\bar{u}_i^k$ and $\bar{v}_i^k$ (7) into the quadratic search routine D and compute α_k
endif
endif
v. Compute the next point $x^{k+1} = x^k + \alpha_k d^k$
vi. *if* $\| d^k \| < \mathrm{Tol1}$, $\| \nabla_x L^{k-1} \| - \| \nabla_x L^k \| < \mathrm{Tol2}$ or $k = \mathrm{SQPMaxIter}$, stop and return x^k as the optimum point.
else
Call Routine E to update the Hessian matrix *H*
endif
vii. Set $k = k + 1$ and loop back to step **i**.

IV. PERFORMANCE EVALUATION

The proposed SQP/IIPM algorithm is implemented in Matlab and series of tests were carried out using twenty five constrained nonlinear benchmark problems for performance evaluation [27, 28, 29]. Details of the test problems are presented in Table I. Both SQP/IIPM and the standard SQP algorithms are tested with similar but strategically chosen starting points that are significantly away from the optimal solution.

TABLE I. TEST PROBLEMS PARAMETERS

S/No.	Problem Category	Variables	Constraints		Bounds		Active Constraints
			<	=	U	L	
1	PQR-T1-2 [27]	2	2	0	2	1	4
2	QQR-T1-7 [27]	2	5	0	2	2	5
3	PQR-T1-8 [27]	3	2	0	1	3	4
4	LGR-T1-1 [27]	2	3	0	2	2	5
5	PPR-P1-1 [27]	3	2	0	3	3	2
6	PGR-P0-1 [27]	4	5	0	0	0	4
7	QGR-P0-2 [27]	7	4	0	0	0	3
8	GQR-P1-1 [27]	7	5	0	0	7	12
9	PQR-P1-1 [27]	2	3	0	2	2	7
10	GQR-P1-1 [27]	2	3	0	2	2	5
11	QQR-T1-11 [27]	4	3	0	0	0	2
12	PPR-P1-3 [27]	4	1	1	4	4	2
13	LGI-P1-1 [27]	4	2	1	0	4	3
14	QQR-P1-4 [27]	5	6	0	5	5	5
15	PPR-P1-7 [27]	7	4	0	0	0	2
16	PPR-P1-11 [27]	8	6	0	8	8	4
17	QQR-T1-8 [27]	3	1	0	3	3	2
18	QPR-T1-2 [27]	3	1	1	0	3	2
19	section 2.16 [28]	2	0	0	2	2	0
20	section 2.6 [28]	20	0	0	2	2	0
21	section 2.10 [28]	5	0	0	5	5	0
22	section 2.1 [28]	20	0	0	2	2	0
23	section 2.12 [28]	5	0	0	5	5	0
24	size 20 [28]	20	5	0	2	2	45
25	size 100 [28]	100	25	0	1	1	225

In the following, table II shows default parameter settings used for the test. These settings prove compliant and suitable for various problem types of different nature. However, performance can be improved by adjusting the settings for each individual problem.

TABLE II. PARAMETER SETTINGS

Parameters	Default Values
SQP/IIPM algorithm tolerance SQPTol-1 (Tol1)	1e-3
SQP/IIPM algorithm tolerance SQPTol-2 (Tol2)	1e-3
IIPM QP solver tolerance QPTol-1 (e1)	1e-16
IIPM QP solver tolerance QPTol-2 (e2)	1e-16
IIPM QP solver tolerance QPTol-3 (e3)	1e-8
Quadratic Search tolerance (QSTol)	1e-1
SQP/IIPM algorithm MaxIter (SQPMaxIter)	250
IIPM QP solver MaxIter (QPMaxIter)	100
Quadratic Search MaxIter (QSMaxIter)	10
Line search MaxIter (LSMaxIter)	3

With regard to the test results tabulated in III, attention is given to three key performance measures including total number of iterations, number of function evaluations and execution time required to converge to the optimum solution point. As can be seen from the test results shown in table III, SQP/IIPM algorithm is considerably faster than the standard SQP algorithm. This can be attributed to the infeasible nature of our IIPM QP solver that overcomes the burden for finding a feasible starting point.

TABLE III. PERFORMANCE COMPARISON: SQP/IIPM AND SQP ALGORITHM

S/No.	Iteration		Function Count		Execution Time(sec)	
	SQP/IIPM	SQP	SQP/IIPM	SQP	SQP/IIPM	SQP
1	4	10	7	36	0.2602	2.0802
2	6	7	11	24	0.2893	2.0267
3	5	5	9	24	0.2711	1.5529
4	18	8	19	32	0.4227	1.6039
5	33	10	66	41	0.5567	1.1059
6	34	34	68	199	0.6071	1.6969
7	21	21	45	168	0.4594	1.6286
8	12	14	23	115	0.3584	1.6475
9	6	7	11	21	0.3174	1.6782
10	30	7	60	21	0.5228	1.9805
11	36	13	72	70	0.5539	1.2232
12	7	11	13	58	0.2441	2.0270
13	4	5	5	25	0.2083	1.6421
14	28	4	55	24	0.7608	1.6070
15	117	21	234	216	1.5913	2.1306
16	56	30	112	274	1.1265	1.6892
17	14	15	28	60	0.3745	1.1779
18	4	3	7	12	0.2040	1.6590
19	6	9	11	30	0.2867	1.6227
20	2	3	3	75	0.2906	1.3162
21	11	13	21	92	0.4437	1.2294
22	2	3	3	6	0.2717	1.2189
23	15	15	29	105	0.5287	1.4502
24	31	54	61	1157	1.3998	2.2953
25	84	253	167	2579	121.5179	123.248

As a result of the technique of adaptive switching between line search and quadratic search algorithms for evaluating the step length parameter, compared to the standard SQP algorithm, our SQP/IIPM algorithm always requires less iteration for all problems of size 20 or higher. Critical observation of the results shown in table III reveals that, out of the 25 test problems, compared to the standard SQP algorithm, SQP/IIPM algorithm converges to the optimum solution with less number of iteration for 13 test problems, equal number of iterations for 4 test problems and higher number of iterations for 8 test problems. In essence, we can infer that, SQP/IIPM algorithm is iteration-wise more efficient than the standard SQP algorithm for all medium to large sized constrained nonlinear optimization problems.

Moreover, the number of function evaluations or gradient counts required by SQP/IIPM algorithm is only at most twice the required number of iterations for all problem types. This is by far smaller compared to that of the standard SQP algorithm which takes at least 3-times its number of iterations. Furthermore, for small to medium sized problems, a remarkably reduced execution time (CPU time) is required by the SQP/IIPM algorithm for convergence to the optimal solution as compared to standard SQP algorithm.

In order to make clear the distinction between the two algorithms, in the following we will demonstrate graphically a comparison of the required number of iterations, function evaluations and execution time via the following charts in figures 3, 4 and 5 respectively.

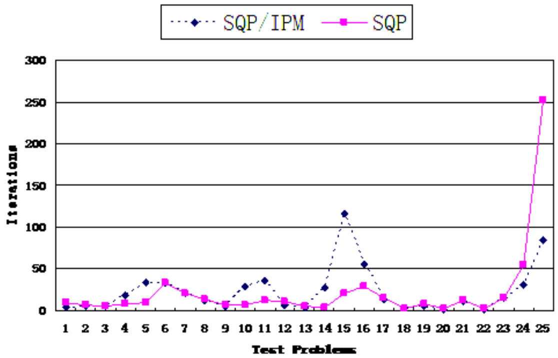

Figure 3. Number of Iterations: SQP/IIPM versus SQP Algorithm

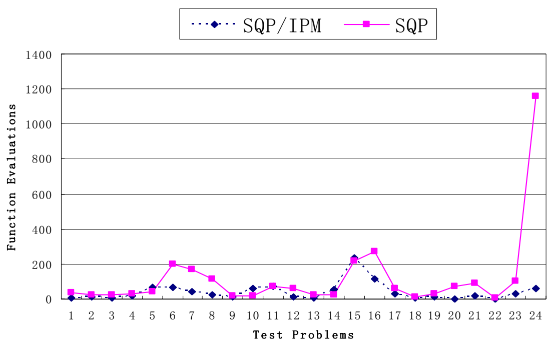

Figure 4. Functions Evaluations: SQP/IIPM versus SQP Algorithm

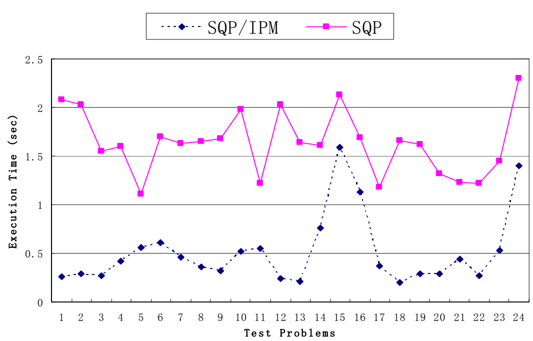

Figure 5. Execution Time: SQP/IIPM versus SQP Algorithm

V. Conclusions

SQP/IIPM algorithm uses IIPM to minimize a series of QP subproblems derived by linearization of the originally nonlinear constrained problem. The test results reveal that beside improved efficiency, the IIPM QP solver increases flexibility in the choice of starting points thereby lifting the burden for the determination of a starting feasible active set. The performance improvement demonstrated by the proposed algorithm could be attributed to the notion that, unlike the Active set algorithms that search by tracking around the boundary of the feasible region, IIPM algorithm cut through the interior of the search space but it admittedly obtained rather approximate solutions. Thus, we can conclude that SQP/IIPM exhibits higher efficiency, robustness and remains promising.

For future improvements, we recommend combining SQP/IIPM algorithm with a genetic algorithm (GA). This is hoped to establish a hybrid system that combines local and global optimization algorithms to enhance the chances of converging to global optimum solutions for all NLP types. Further considerations are needed to secure exact Hessian values especially for the indefinite problems as second derivatives are increasingly available.

Acknowledgment

I would like to thank the National Key Basic Research and Development Program of China (2002CB312203), National Natural Science Foundation of China (60874070) and National Research Foundation for the Doctoral Program of Higher Education of China (20070533131).

References

[1] Esben L. H. and Carsten V. Numerical Algorithms for Sequential Quadratic Optimization. [Thesis]. Kongens Lyngby, Technical University of Denmark Informatics and Mathematical Modelling 2007 IMM-M.Sc-2007-70

[2] Wilson R. B. A simplicial algorithm for concave programming: [Dissertation]. Harvard, Graduate School of Business Administration, Harvard University, 1963

[3] Garcia-Palomares U. M, Gonzalez-Castano F. J. and Burguillo-Rial J. C. A combined Global & Local Search (CGLS) Approach to Global optimization. Journal of Global optimization, 2006, 34(3)

[4] Ju-Liang Z. and Xiang-Sun Z. A modified SQP method with Nonmonotone Linesearch Technique. Journal of Global optimization, 2001, 21(2): 201 – 218

[5] Powell M. J. D. Approximation Theory and Optimization. edited by M. D. Buhmann and A. Iserles. Cambridge: Cambridge University Press, 1997

[6] Powell M. J. D. Nonconvex minimization calculations and the conjugate gradient method: [Lecture Notes in Mathematics]. Philadelphia, 1066: 122 – 141, 1984

[7] Fletcher R. A New Low-Rank Quasi-Newton update scheme for Nonlinear Programming. International Federation for Information Processing, System modelling and Optimization, 2006, 199: 275–293

[8] Melara L. A., Kearsley J. and Tapia R. A. Argumented Lagrangian Homotopy method for the Regularization of Total Variation Denoising Problems. Journal of optimization Theory and Application, 2007, 134(1)

[9] P. E. Gill, W. Murray, and M. A. Saunders NPSOL and SNOPT – SQP Methods MS&E 318 (CME 338) Large Scale Numerical optimization. Stanford University, Department of Science and Engineering, 2008

[10] Wirght S. J. "Numerical Optimization" New York: Springer series in Operations Research, 1995

[11] Michael Todd Optimization: An Interior Point View. Newsletter, Institute for Mathematical Sciences, NUS, 2007, (10): 17–20

[12] Boggs P. T., Kearsley A. J. and Tolle J. W. A Practical Algorithm for General Larges scale Nonlinear Optimization problem. Siam Journal of optimization. 1999, 9(3): 755–778

[13] Gould N. I. M., Hribar M. E., and Nocedal J. On the solution of equality constrained quadratic programming problems arising in optimization. SIAM J. Sci. Comput., 2001, 23(4):1376–1395

[14] Fiacco A. V. and McCormick G. P. Nonlinear Programming: Sequential Unconstrained Minimization Techniques. John Wiley & Sons, New York, NY, reprinted by SIAM Publications, 1990

[15] Fletcher R. A New Low-Rank Quasi-Newton update scheme for Nonlinear Programming. International Federation for Information Processing, System modelling and Optimization, 2006, 199: 275–293

[16] Gill P. E., Murray W. and Saunders M. A. NPSOL and SNOPT – SQP Methods MS&E 318 (CME 338) Large Scale Numerical optimization. Stanford University, Department of Science and Engineering, 2008

[17] Francisco A., Gomes M. A sequential quadratic programming algorithm that combines merit function and filter ideas. Department of Applied Mathematics, IMECC, Universidade Estadual de Campinas 2004, 13081-970, Campinas, SP, Brazil

[18] Murray W. Sequential quadratic programming methods for large-scale problems. J. Comput. Optim. 1997, 7

[19] Wirght S. J. "Numerical Optimization" New York: Springer series in Operations Research, 1999

[20] Biegler, L. T., Coleman, T. F., Conn, A. R., Santosa, F. N. Large Scale optimization with Applications Part I: Optimization in Inverse Problems and Design. Springer-Verlag, 1997

[21] Bhatti M. A. Practical Optimzation methods with Mathematica applications. New York: Springer New York Inc. Library of Congress Cataloging in Publication Data, 2000

[22] Adler I. M., Resende G. C., Geraldo V., and Karmarkar N. An implementation of Karmarkar's algorithm for linear programming. Mathematical Programming, 1989, 44: 297–335

[23] Dikin I. I. Examples in the Theory of Convergence of the Affine Scaling Method. Cybernetics and System Analysis, 2006, 41(6): 157 – 158

[24] Megiddo N. Pathways to the optimal set in linear programming, Interior-Point and Related Methods. N. Megiddo, ed., [Thesis]. Springer-Verlag, New York, N.Y., 1989, (8): 131–158

[25] Waechter A. and Biegler L. T. On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. Math. Program., 2006, 106(1): 25–57

[26] Ju-Liang Z. and Xiang-Sun Z. A modified SQP method with Nonmonotone Linesearch Technique. Journal of Global optimization, 2001, 21(2): 201–218

[27] Hock W. and Schittkowski K. Test Examples for Nonlinear Programming Codes: [Lecture Notes in Economics and Mathematical Systems]. Springer-Verlag, Berlin Heidelberg, New York, 1981

[28] Pohlheim H., GEATbx Examples of Objective functions. 2005, 3.7. www.geatbx.com 2005

[29] Vassiliadis V. S. and Floudas C. A. The modified Barrier Function Approach for Large-scale optimization. Computers Chem. Engng., 1997, 21(8): 855–574

Author Biographies

Hassan A. Bashir received the Degrees of B. Eng. Electrical from Bayero University Kano, Nigeria in 2004 and M. Eng. Computer Application Technology from Central South University, China in 2009. His research interests include Nonlinear Numerical Optimization, Hybrid Optimization approaches and Optimization of Engineering Control systems. He has made four publications so far.

Ximing Liang received the B.S. degree from Yunnan University, Kunming, China, in June 1989 and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in June 1992 and April 1998, respectively, all in Mathematics. From September 1998 to June 2000, he was a postdoctoral researcher in the Institute of Industrial Control Technology, Zhejiang University, Hangzhou, China. Since then, he has been teaching and doing research in the School of Information Science and Engineering, Central South University, Changsha, China. From March 2005 to September 2006, he visited the Department of Chemical Engineering, Department of Petroleum Engineering, The University of Texas at Austin, Austin, Texas, USA, as a visiting scholar. He is presently a professor of Control Science and Engineering at Central South University, Changsha, China. His research interests include process control, optimization method and evolutionary computation. Over 90 papers have been published.