

Research of Large-Scale Reduced SQP Algorithm for Chemical Process System Optimization

Jiang Aipeng*, Jiang Zhushu*, Huang Jingtao**, Wang Jian**

*School of Automation, HangzhouDianzi University, Hangzhou, China
(e-mail: jiangaipeng@163.com).

**Electronic and Information Engineering College, Henan University of Science and Technology, Luoyang, China
(e-mail: hjt.haust@gmail.com)

Abstract: In chemical process system optimization, it is common that problem has large number of equality constraints and bound constraints, and relatively few degrees of freedom. Reduced space algorithms are well suited for this category of problem. In this paper, a new version of RSQP algorithm coded and implemented in Matlab language was developed. In the RSQP algorithm, the rule of basis selection was revised, and basis selection was realized by Matlab subroutine. Also, an integrated line search of filter method was performed to obtained steplength. The method combines the advantages of filter line search and that of traditional line search, and was validated by some benchmark examples. The RSQP was applied to some chemical process optimization problems; computational results demonstrate its effectiveness and efficiency.

1. INTRODUCTION

In last twenty years, Sequential Quadratic Programming(SQP) algorithms are viewed as the best approach[1] to solve nonlinear problems. SQP method generates a sequence of point $\{x_k\}$ converging to the solution by solving a quadratic program (QP) sub-problem at each point x_k . It must deal with a $n \times n$ Hessian matrix. When number of variables increases, the matrix of Hessian needs much more memory and is very expensive for computation. So Biegler and etc. [2-5] presented a reduced SQP algorithm to problems with large number of equality constraints and relatively low degrees of freedom. In reduced SQP algorithm, QP sub-problem is solved in relatively small space by the way of space decomposition. In chemical process system, this kind of problems exists extensively. So the method is quite attractive, and it has been applied in process optimization [6-7].

In reduced SQP algorithm, how to select basic variables is of great importance. It can greatly affect the efficiency and stability of the algorithm. In this paper, Matlab coded routines are used for basis selection, and new rule of basis selection is regulated. The method for basis selection is similar and somehow different from the work of Bigler[15]. With the rule of basis selection, sparse technology was utilized to check the rank of Jacobian matrix, and multipliers were adjusted according to steplength.

Since Fletcher and Leyffer[8] presented the idea of filter method, line search of filter method has become a hot topic in the research of NLP[9-10]. Main concept of the line search of filter method is that[8]: in the filter region, a steplength is accepted if new iterate x_{k+1} minimizes infeasibility or objective function. Since the conditions to accept steplength are slackened, it is possible to obtain a bigger steplength.

Filter method requires sufficient decrease in either objective function or infeasibility at each iteration. But sometimes either the objective function or infeasibility in exchange for a small decrease in the other may occur. And when this method suffers from numerical difficulties, "Restore Phase" is generally used. In this paper, traditional and filter method of line search are integrated and performed to obtain steplength. It combines the advantages of filter method and that of traditional line search, and is easier in implementation.

2. REDUCED SQP ALGORITHM

In chemical process system, many optimization problems can be formulated as nonlinear programming (NLP) problems. Generally, cost or profit is valued as objective function; model equality and the bound requirements are valued as constraints. With introducing slack variables, inequality constraints can be converted into equality constraints. Thus the model can be described as:

$$\left. \begin{array}{l} \min_{x \in R^n} f(x) \\ c(x) = 0 \\ x^L \leq x \leq x^U \end{array} \right\} \quad (1)$$

Here $f: R^n \rightarrow R$, $c: R^n \rightarrow R^m$ are twice continuously differentiable functions. SQP algorithms approach the optimal solution by solving a sequence of quadratic programming sub-problems

$$\left. \begin{array}{l} \min_{d \in R^n} g(x_k)d_k + 1/2d_k^T W(x_k)d_k \\ c(x_k) + A(x_k)d_k = 0 \\ x^L \leq x_k + d_k \leq x^U \end{array} \right\} \quad (2)$$

Here, d_k denotes the search direction, $g(x_k)$ denotes the gradient of $f(x_k)$, $W(x_k)$ denotes the Hessian of Lagrangian function

$$L(x, \lambda, \nu, \pi) = f(x) + \lambda^T c(x) + \nu^T (x - x^U) - \pi^T (x - x^L) \quad (3)$$

$A(x_k)$ stands for the $n \times m$ matrix of gradients of $c(x_k)$.

$$A(x_k) = [\nabla c_1 \quad \nabla c_2 \quad \dots, \quad \nabla c_m] \quad (4)$$

Assuming the first m variables as basic variables (dependent variables) and the other $(n-m)$ variables as nonbasic variables (independent variables), A_k^T can be grouped as:

$$A(x_k)^T = [N(x_k) \quad C(x_k)] \quad (5)$$

In reduced SQP, the search direction d_k is partitioned into two parts: [4,5]

$$d_k = Y_k p_y + Z_k p_z \quad (6)$$

Here Z_k is $n \times (n-m)$ matrix spanning the null space of A_k^T (which is assumed to have full column rank for all iterations), Y_k is $n \times m$ matrix spanning the range space of A_k^T , p_z and p_y are vectors in R^{n-m} and R^m respectively. There are many choices for constructing the matrix of Y and Z [11]. By QR factorization of A_k^T to get Y_k and Z_k , Y_k and Z_k have orthonormal columns. This method gives a well-conditioned representation of null space and rang space of A_k^T . But it could be very expensive in calculation if the number of variable is large. Another way to choose Y and Z is called orthogonal bases method. This method is less expensive, but the sparse structure of A_k^T can't be kept well. In this paper the way called coordinate bases method[12] was adopted to construct Y and Z . Y and Z are defined as:

$$Z(x_k) = \begin{bmatrix} I \\ -C(x_k)^{-1} N(x_k) \end{bmatrix} \quad Y(x_k) = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad (7)$$

Since Z_k is in the null space of A_k^T , then

$$A_k^T Z_k = 0 \quad (8)$$

Substituting (6) and (8) into (2), then p_y can be solved by

$$p_y = -(A_k^T Y_k)^{-1} c_k \quad (9)$$

So d_k can be formulated as

$$d_k = -Y_k (A_k^T Y_k)^{-1} c_k + Z_k p_z \quad (10)$$

To determine the component p_z , we substitute (10) into (2) and eliminate terms not involving p_z . Then (2) can be expressed as the QP sub-problem in terms of the $(n-m)$ dimension variables p_z :

$$\left. \begin{aligned} \min_{p_z \in R^{n-m}} (Z_k^T g_k + w_k)^T p_z + 1/2 p_z^T B_k p_z \\ x^L - x_k - Y_k p_y \leq Z_k p_z \leq x^U - x_k - Y_k p_y \end{aligned} \right\} \quad (11)$$

Here w_k is a vector approximating $Z_k^T W_k Y_k p_y$, B_k denotes matrix $Z_k^T W_k Z_k$. For process system with large number of variables and relatively small degrees of freedom, i.e. $n \gg (n-m)$, B_k is quite small. It can be approximated by BFGS method with much less computational cost. The vector w_k is relatively large; it can be ignored or approximated via Broyden method.

If the cross term is not ignored, it can be calculated by Broyden's method. Matrix $Z_k^T W_k$ is approximated by U_k and can be updated as

$$U_{k+1} = U_k + \frac{(\bar{y}_k - U_k \bar{s}_k)}{\bar{s}_k^T \bar{s}_k} \quad (12)$$

where

$$\bar{y}_k = Z_k^T [\nabla L(x_{k+1}, \lambda_{k+1}, \nu_{k+1}, \pi_{k+1}) - \nabla L(x_k, \lambda_k, \nu_k, \pi_k)] \quad (13)$$

$$\bar{s}_k = x_{k+1} - x_k \quad (14)$$

then w_k is calculated by

$$w_k = S_k Y_k p_y \quad (15)$$

B_k is updated by BFGS formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \quad (16)$$

$$B_{k+1} s_k = y_k \quad (17)$$

$$s_k = \alpha_k p_z \quad (18)$$

$$\text{and } y_k = Z_k^T [\nabla L(x_{k+1}, \lambda_{k+1}, \nu_{k+1}, \pi_{k+1}) - \nabla L(x_k, \lambda_k, \nu_k, \pi_k)] - \bar{w}_k \quad (19)$$

$$\bar{w}_k = \alpha_k S_{k+1} Y_k p_y \quad (20)$$

Powell recommends keeping the Hessian positive definite even though it might be positive indefinite at the solution point. A positive definite Hessian is maintained providing $y_k^T s_k$ is positive at each update.

$$L(x, \lambda, \nu, \pi) = f(x) + \lambda^T c(x) + \nu^T (x - x^U) - \pi^T (x - x^L)$$

When $y_k^T s_k$ is not positive, y_k is modified on an element-by-element basis so that $y_k^T s_k > 0$. If $y_k^T s_k$ is still not positive with the procedure, the update is skipped.

In RSQP algorithm, since QP problems are reduced from dimension of n to the relatively small dimension of $(n-m)$, they can be solved with much less memory and computational cost.

The whole algorithm of RSQP is listed in figure 1.

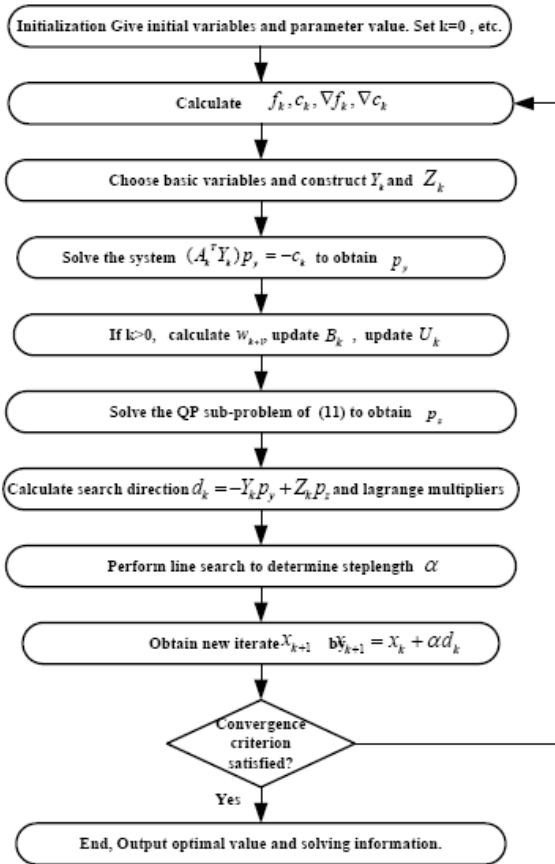


Fig. 1. Schematic of RSQP algorithm

3. SELECTION OF BASIS

When coordinate bases are used to select Y and Z , formula (9) must be solvable, that is to say $A^T Y = C$ should be nonsingular. But in the process of solving QP sub-problems, $A^T Y$ may be singular or nearly singular, which will cause the problem unsolvable. To avoid this, basic and nonbasic variables should be checked and well selected at all iterations.

To guarantee $A^T Y$ nonsingular, permuting columns of A^T

$$\bar{A}^T = A^T P \quad (21)$$

To ensure the leading square matrix of \bar{A}^T is nonsingular, the variables can be rearranged as

$$\bar{x} = P^T x \quad (22)$$

Accordingly, Z, B and etc. are converted into the form [4]

$$\bar{Z} = \begin{bmatrix} I \\ -\bar{C}^{-1} \bar{N} \end{bmatrix} \quad (23)$$

$$\nabla \bar{f} = P^T \nabla f \quad (24)$$

$$\bar{B} = \bar{Z}^T \bar{W} \bar{Z} = \bar{Z}^T P^T W P \bar{Z} \quad (25)$$

Here \bar{x} denotes the variables after basis selection, $\nabla \bar{f}$ denotes the gradient of $f(x)$, ∇f denotes the gradient of $f(\bar{x})$, P denotes permutation matrix.

In Biegler's work[15], Fortran coded software MA28 or MA48 was used to select basic variables and to get the basis

permutation matrix. To realize the same function in Matlab language, subroutine named findp was used to select the basis permutation matrix. In the subroutine P is obtained by choosing column minimum degree permutation vector[13] from A and LU factorization of A .

Since Matlab is more excellent in matrix operation, its functions were applied to the rule of basis selection. In this paper, we make some revisions for the basis selection rule described in Biegler etc [15]. The rule for selection of basis is described as:

(1) Let $\gamma = \max_{i,j} |(C^{-1}N)_{i,j}|$, if $\gamma_{k+1} \geq 20\gamma_k$ or C is singular, then new basic and nonbasic variables are selected again, Because C is check by sparse rank function, the calculation cost is very small.

(2) If steplength $\alpha < 0.01$ and $\gamma_{k+1} > 2\gamma_k$ or steplength $\alpha < 0.001$, then the multipliers of merit function $\mu_{k+1} = 10\mu_k$, and new basic variables and nonbasic variables are selected again.

Here γ is the parameter to describe the biggest element of matrix $|C^{-1}N|$. For constructing well conditioned Z , it should not increase too much at each iteration. In step (2), if steplength is small and γ increases quickly, new basis is selected to obtain good search direction. With new basis selection taken, μ is increased 10 times to guarantee sufficient decrease of merit function.

4. INTEGRATED LINE SEARCH OF FILTER METHOD

To make sure the global convergence of RSQP algorithm, new iterate x_{k+1} is calculated by the following formula. Here α is the steplength at the interval of $(0, 1]$, which must be selected carefully.

$$x_{k+1} = x_k + \alpha d_k \quad (26)$$

Now line search of filter method is a very open topic. Though it may obtain a bigger steplength, it still has its disadvantages. It is difficult to obtain sufficient decrease of objective function or infeasibility each time. Also it may sometimes suffer from numerical difficulties. But traditional line search method does not have these disadvantages with merit function employed. To incorporate advantages of traditional and filter line search methods, an integrated line search method of filter was performed to obtain steplength. In this method, the maximum infeasibility of constraints is expressed as

$$mg = \max(|c(x)|_\infty, \max(0, |x - x^U|_\infty, |x^L - x|_\infty)) \quad (27)$$

The objective function is replaced by L1 merit function, which is defined as below:

$$\phi(x) = f(x) + \mu \|c(x)\|_1 + \kappa \left\| \begin{bmatrix} \max(0, x - x^U) \\ \max(0, x^L - x) \end{bmatrix} \right\|_1 \quad (28)$$

If all variables satisfy the bound constraints, it can be proved[14] that the direction derivative of $\phi(x)$ is

$$D\phi_\mu(x) = \nabla f(x)'d + \mu \|c\| \quad (29)$$

To describe the integrated line search of filter method, several combinational conditions are defined as:

Condition 1

At iterate x_{k+1} , merit function has some degrees of reduction, that is $\phi_\mu(x_{k+1}) \leq \phi_\mu(x_k) + \varepsilon\alpha D\phi_\mu(x_k)$

ε denotes a small positive number, generally it can be taken to be $1.0e-6$

Condition 2

At iterate x_{k+1} , maximum infeasibility of constraints has some degrees of reduction, that is

$$mg(x_{k+1}) \leq mg(x_k)^2 + 2\varepsilon\alpha c(x_k)^T \nabla c(x_k) d_k$$

Condition 3

Maximum infeasibility of constraints is fairly small, $mg \leq 1.0e-6$

Condition 4

$$\left| \frac{\phi_\mu(x_{k+1})}{\phi_\mu(x_k)} - 1 \right| / \left(\frac{mg_{k+1}}{mg_k} - 1 \right) \leq \beta$$

$$\left| \frac{\phi_\mu(x_{k+1})}{\phi_\mu(x_k)} - 1 \right| / \left(\frac{mg_{k+1}}{mg_k} - 1 \right) \geq 1/\beta$$

here β is a positive number

Condition 5

$$\alpha < 1.0e-4$$

The procedure of integrated line search of filter method is illustrated in figure 2. Traditional line search method is implemented as described in literature [25].

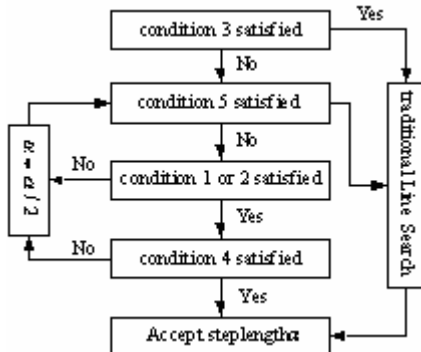


Fig. 2. Schematic of integrated line search method of filter

The integrated line search method utilizes the concept of filter and the idea of merit function. It can get sufficient reduction of merit function or maximum infeasibility at each iteration. At the same time a large increase in either the merit function or the infeasibility in exchange for a small decrease in the other can be avoided. Furthermore, when filter method of the integrated line search suffers from numerical difficulties, traditional line search is employed to obtain the steplength. To avoid Marotos effect, SOC (Second Order Correction) is employed in the new line search method.

5. NUMERICAL RESULTS

Because Matlab is the package with least effort in modeling NLP problems[24] and is excellent in matrix operation and visibility, the RSQP algorithm was coded in Matlab language. The basis selection and line search method proposed above

were incorporated into the implementation of the new version of RSQP algorithm. Efficiency was compared and analyzed in solving some benchmark problems.

Examples in Table 1 come from literatures of Schmid and Biegler[4,15]. All of these examples have large number of equality constraints and variable dimensions. Exam1 and Exam3 have relatively low degrees of freedom. But Exam2 has relatively large degrees of freedom. They increase quickly with the increase of dimension of variables. Some benchmark examples from Murtagh[16] and Hock[17] are also selected to validate the performance of the improved version of RSQP algorithm.

Table 1. large benchmark examples

Examples	Expression
Exam1	$\min 0.5 \sum_{i=1}^n x_i^2$ $x_1(x_{j+1}-1) - 10x_{j+1} = 0 \quad j = 1, \dots, n-1$ $x_j^0 = 0.1, x_j^* = 0$
Exam2	$\min 0.5 \sum_{i=1}^n x_i^2$ $x_j(x_{j+n/2}-1) - 10x_{j+n/2} = 0 \quad j = 1, \dots, n/2$ $x_j^0 = 0.1, x_j^* = 0$
Exam3	$\min \sum_{i=1}^{n-1} (x_i + x_{i+1})^2$ $x_j + 2x_{j+1} + 3x_{j+2} - 1 = 0, \quad j = 1 \dots n-2$ $x_1^0 = -4; \quad x_i^0 = 1, i > 1$

5.1 Performance of RSQP with new rule of basis selection

In RSQP algorithm, coordinate bases method was used to construct Y and Z , traditional line search was used to obtain steplength. Performance of the RSQP algorithm with and without rule of basis selection was compared. The results are listed in Table 2 and Table 3.

Table 2. Results of large benchmark examples

Examples	N/M	Basis Selection	No Basis Selection
		Iter/FunEvals	Iter/FunEvals
Exam1	1000/999	9/24	10/27
	2000/1999	4/15	9/24
	3000/2999	6/19	10/28
Exam2	1000/500	4/9	21/69
	2000/1000	4/9	29/96
	3000/1500	4/9	24/79
Exam3	300/298	4/12	#
	600/598	4/12	#
	900/898	4/12	#

* N/M denotes variables/ equality constraints; Iter/FunEvals denotes iterations/ function evaluations ; # denotes failed or iterations >500;

From Table 2 and Table 3, we can see that with the rule of basis selection employed, number of iterations and function evaluations reduced obviously. The stability of RSQP algorithm was also improved obviously. Most examples

require less iterations and function evaluations. The RSQP algorithm succeeded in solving Exam3, Entropy and Hs6 with basis selection, but failed without basis selection. This is because the leading square matrix C is singular or nearly singular in solving QP sub-problems.

Table 3. Results of small benchmark examples

Examples	N/M	Basis Selection	No Basis Selection
		Iter/FunEvals	Iter/FunEvals
Entropy	10/1	10/136	#
Box	3/1	11/26	11/26
Wright4	5/3	9/22	9/21
Hs6	2/1	14/32	#
Hs12	2/1	8/17	31/104
Hs36	3/1	3/7	2/5
Hs39	4/2	31/113	113/1222
Hs40	4/3	6/13	6/13
Hs42	4/1	16/49	#
Hs46	5/2	13/31	15/37
Hs80	5/3	10/25	9/24
Hs99	7/2	37/135	468/1836

5.2 Performance of RSQP with integrated line search of filter method

To validate performance of the proposed line search method, we incorporated it into RSQP algorithm. Then the algorithm was used to solve these benchmark examples. All results were compared with those solved by the RSQP algorithm with traditional line search to obtain steplength.

The comparison results of Exam1-3 are listed in table 4. For the first two examples, the iterations and function evaluations of integrated line search of filter method were no less than those of traditional line search. But for Exam3, the integrated line search method decreased iterations and function evaluations obviously.

Table 4. Results of Exam1-3 with integrated line search of filter method

Examples	N/M	Traditional Line Search	Integrated Line Search
		Iter/FunEvals	Iter/FunEvals
Exam1	1000/999	9/24	6/17
	2000/1999	4/15	8/22
	3000/2999	6/19	7/19
Exam2	1000/500	4/9	4/9
	2000/1000	4/9	4/9
	3000/1500	4/9	4/9
Exam3	300/298	4/12	2/6
	600/598	4/12	2/6
	900/898	4/12	2/6

From table 5, it can be found that with integrated line search of filter method employed, these benchmark examples could be solved more effectively. Most examples need less iterations and function evaluations. With traditional line search adopted, the total number of iterations/function evaluations is 168/606. But with integrated line search of filter method adopted, the total number of iterations/function

evaluations is 134/496. The integrated line search method outperforms traditional line search.

Table 5. Results of benchmark examples with integrated line search of filter method

Examples	N/M	Traditional Line Search	Integrated Line Search
		Iter/FunEvals	Iter/FunEvals
Entropy	10/1	10/136	9/121
Box	3/1	11/26	11/24
Wright4	5/3	9/22	11/32
Hs6	2/1	14/32	6/34
Hs12	2/1	8/17	8/17
Hs36	3/1	3/7	2/5
Hs39	4/2	31/113	24/134
Hs40	4/3	6/13	6/13
Hs42	4/1	16/49	11/29
Hs46	5/2	13/31	9/29
Hs80	5/3	10/25	10/30
Hs99	7/2	37/135	27/128
Total		168/606	134/496

5.3 Chemical case study

To make comparisons, chemical problems were also solved by the standard SQP algorithm (Fmincon)[19] in Matlab6.5 and SNOPT. The operation system is Windows 2000. The computing platform has Intel 1.7GHz, 1 GB memory.

5.3.1 Distillation column optimization

Two distillation columns in ethylene process were optimized for case study. The flowsheet of the system is described as figure 3:

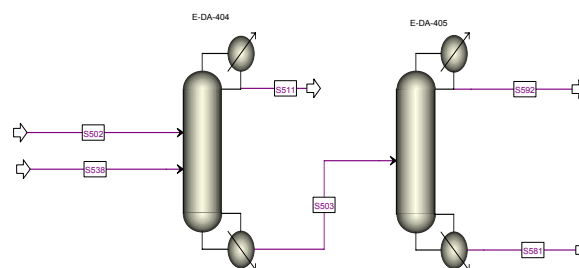


Fig. 3. Flowsheet of the columns in ethylene process

E-DA-404 denotes depropanizer and E-DA-405 denotes debutanizer. Depropanizer has 47 theoretical trays and two feeds input. Debutanizer has 35 theoretical trays and its feed comes for the bottom output of depropanizer. Details of the system can be seen in literature [26].

With open formed model based on rigorous principle founded, the objective function is as below:

$$\text{Max} : F = (278.88 \times S1_bw + 227.752 \times S1_dw) / 10000$$

$S1_bw$ denotes top product flow of depropanizer and $S1_dw$ denotes top product flow of debutanizer.

F denotes the total economic efficiency.

The constraints are:

Mesh equality of distillation process
Relationship equality for the two columns

Bound constraints:

In product $S1_bw$, $C3H6 \geq 0.93$;

In product $S1_dw$, $C5 \leq 0.0095$;

The whole system has 4190 free variables, 103 fixed variables and 4188 equality constraints. So the whole system's degrees of freedom are 2. We solve the problem by RSQP, SNOPT and Fmincon. Fmincon failed to solve the problem because of its difficulty to deal with large matrix. But RSQP and SNOPT can use the sparse information of the problem and can solve large scale problems quickly. Computing information and optimization results are shown in table6, table7 and figure4, figure5.

Table 6. Computing Comparison of RSQP and SNOPT

Algorithm	Iter/ FunEval	Grad call	Objective	Time(s)
SNOPT	92/364	117	-20.2943	5685.31
RSQP	28/93	45	-20.2901	1240.164

Table 7. Computing information of RSQP in detail

Iter/ FunEval	Statistics of computing time(s)					Total
	BFGS& Broyden	gradient	Line search	QP	Space decomp	
28/93	311.160	572.000	7.906	1.722	345.473	1240.164

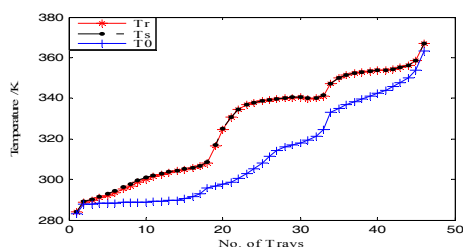


Fig. 4. Temperature profile comparison of depropanizer

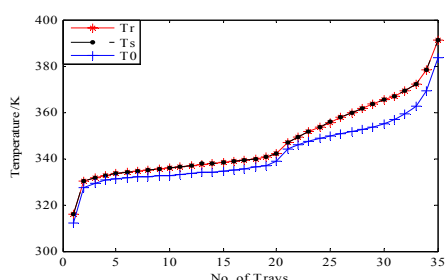


Fig. 5. Temperature profile comparison of debutanizer

The results of table6 show that compared with the famous SNOPT, the RSQP algorithm decreased solving time greatly. The iterations and function evaluations of the RSQP algorithm are much less than SNOPT. Table7 gives the Statistics of RSQP's computing time, from which we can get bottleneck of the algorithm. We can see that the gradient computing time is 572s, improvement of gradient computing can decrease the total solution time obviously. Figure4 and

figure5 are the comparison of optimization results. T0 denotes initial temperature profile; T1 and T2 denote optimal temperature profile gotten by RSQP and SNOPT. From the figures we can see that the optimal results of RSQP and SNOPT are the same, it demonstrate that the optimal results of RSQP is correct.

5.3.2 Parameter estimation of catalytic cracking of gasoil[22]

The reaction mechanism of the catalytic cracking of gasoil(A) to gasoline(Q) and other byproducts(S). The reaction model is formulated as

$$\left. \begin{aligned} \frac{dz_1}{dt} &= -(\theta_1 + \theta_3)z_1^2 \\ \frac{dz_2}{dt} &= \theta_1 z_1^2 - \theta_2 z_2 \end{aligned} \right\}$$

Initial conditions: $z_0 = (1, 0)$

Variable bounds: $\left. \begin{aligned} 0 \leq z \leq 1 \\ 0 \leq \theta \leq 20 \end{aligned} \right\}$

The problem is to minimize $\sum_{j=1}^{21} \|z(\tau_j, \theta) - y_j\|^2$, here

y_j denote the concentration measurements for z at time points $\tau_1, \tau_2, \dots, \tau_{21}$.

To solve the dynamical problem, the model was converted into NLP by approximate the state profiles by orthogonal collocation on finite elements. The whole time region was partitioned into 50 finite elements, and three collocation points was used in every element. The discretized model has 503 variables, 500 equality constraints and 1006 bound constraints.

To obtain first order gradient information of the discretized model accurately and quickly, automatic differentiation technology (AD) was utilized to calculate the gradient information. Sparse structure of the Jacobian matrix of the discretized model was also utilized. The problem was successfully solved by RSQP and standard SQP algorithm, and computational results of the problem were list in table8 and figure6.

Table 8. Computational results of discretized gasoil model

Algorithm	BFGS & Broyden	QP	Iter/ FunEvals	Total (s)
Standard SQP	27.078	146.578	48/198	189.500
RSQP	3.890	0.992	63/229	37.156

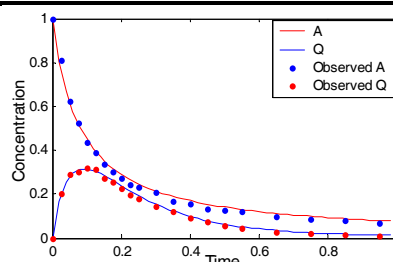


Fig. 6. Solution and measurements of gasoil

Though RSQP algorithm still needs more iteration and function evaluations, computational time is largely reduced. In RSQP algorithm, though most effort is spent on space decomposition and gradient calculation, it is much less than the effort saved in QP solving. Figure 6 shows the solution of concentration and the concentration measurements. It can be found that the solution fits the measurements well.

6. CONCLUSIONS

RSQP algorithm is designed for chemical process system optimization problem with relatively large number of equality constraints and few degrees of freedom. In this paper, a new version of RSQP coded and implemented in Matlab is developed. In the algorithm, the rule for basis selection is revised and basis selection is realized by Matlab subroutines, it has the same function as MA28 and MA48. An Integrated line search of filter method is incorporated in the RSQP algorithm to obtain step length. The RSQP algorithm was validated by some benchmark examples and then was applied to solve chemical process optimization cases. Computational results demonstrate that it is quite effective and is more efficient than standard SQP algorithm.

Since it is very important to keep the global stability of RSQP algorithm, more attention will be placed on trust region method and the calculation of cross term. Also, interior-point method will be combined with RSQP for real-time optimization of chemical process.

REFERENCES

- [1] Boggs, P.T., Tolle, J.W (1995). Sequential quadratic programming. *Acta Numerica*, 1-51.
- [2] Biegler, L.T., Nocedal, J., Schmid, C (1995). A reduced hessian method for large-scale constrained optimization. *SIAM Journal on Optimization*, **5**, 314-347.
- [3] Vansantharajan, S., Biegler, L.T (1988). Large-scale decomposition for successive quadratic programming. *Computers and Chemical Engineering*, **12**, 1087-1103.
- [4] Schmid, C(1994). Reduced Hessian successive quadratic programming for large-scale process optimization, Ph. D dissertation.
- [5] Ternet, D. J., Biegler, L. T(1998). Recent improvements to a multiplier-free reduced Hessian successive quadratic programming algorithm. *Computers and Chemical Engineering*, **22 (7)**: 963-978.
- [6] Shao, Zhijiang., Qian, Jixin(1998). Reduced-space SQP Algorithm for simultaneous optimization of distillation column. *Journal of Chemical Industry and Engineering*, **49 (4)**: 511-514. (In Chinese).
- [7] Wang, Huaishi., Xu, Yifang., Shen, Fu(1997). Advance in the research of reduced hessian SQP. *Journal of the University of Petroleum*, **21(4)**:311-314. (In Chinese).
- [8] Fletcher, R., Leyffer, S (2002). Nonlinear programming without a penalty function. *Mathematical Programming*, **91(2)**:239-269.
- [9] Wächter, A., Biegler, L. T (2005). Line Search Filter Methods for Nonlinear Programming: Motivation and Global Convergence. *SIAM Journal on Optimization* **16(1)**: 1-31
- [10] Hande, Y. B., Robert, J.V., David, F. S(2002). Interior-point methods for non-convex nonlinear programming: Filter methods and merit function, *Computational Optimization and Applications*, **23(2)**:57-272.
- [11] Murray, W., Prieto, F. J(1995). A sequential quadratic programming algorithm using an incomplete solution of the subproblem. *SIAM J. on Optimization*, **5**:589-639.
- [12] Locke, M. H., Edahl, R., Westerberg, A. W(1983). An improved successive quadratic programming optimization algorithm for engineering design problems. *AICHE*, **29 (5)**: 871-874
- [13] Amestoy, P. R., Davis, T. A., and Duff, I. S (1996). An approximate minimum degree algorithm. *SIAM J. Matrix Anal. Applic.*, **17(4)**: 886-905
- [14] Byrd, R. H., Nocedal, J (1991). An analysis of reduced Hessian methods for constrained optimization. *Math. Programming*, **49**:285-308
- [15] Biegler, L.T., Nocedal, J., Schmid, C(2000). Numerical experience with a reduced Hessian method for large-scale constrained optimization, *Computational Optimization and Applications*, **15**:45-67.
- [16] Murtagh, B. A., and Saunders, M. A(1987). MINOS5.1 User's Guid, *Technical Report*, SOL83-20R, Department of OP, Stanford University.
- [17] <http://www.sor.princeton.edu/~rvdb/ampl/nlmodels/hs>
- [18] Shao, Zhijiang., Zhang, yuyue., Qian, jixin (1997). Unified approach for distillation column simulation and optimization based on equation-oriented algorithms. *Journal of Chemical Industry and Engineering*, **48(1)**: 46-51. (In Chinese).
- [19] Coleman, T. F., Branch, M. A. Grace.A(1999). Optimization Toolbox for User with Matlab, Version 2. Natick, MA: The MathWorks Inc.
- [20] Floudas, C. A. Pardalos, P. M. and et al (1999). Handbook of test problems in local and global optimization, London, Kluwer Academic Publishers
- [21] Tjoa, I. B., and Biegler, L. T (1991). Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equations systems. *Industrial Engineer Chemical Research*, **30**, 376-385.
- [22] Dolan, E. D., and Moré, J. J(2001). Benchmarking optimization software with COPS. Technical Report ANL/MCS-246, Argonne National Laboratory.
- [23] Zhong, W. T., Shao, Z., Zhang, Y., Qian, J.X (2000). Applying analytical derivative and sparse matrix techniques to large-scale process optimization problems. *Chinese Journal of Chemical Engineering*, **8**: 212-217.
- [24] Chiang, K. (1998). Performance of several nonlinear programming software packages on microcomputers. *Computers and Operations Research*, **25 (10)**: 807-816.
- [25] Nocedal J., and Wright, S. J(1999): Numerical Optimization. *Springer Series in Operations Research*, New York.
- [26] Jiang, Aipeng., Shao, Zhijiang., Chen, Xi (2006). Operation optimization of distillation column based on reduced SQP algorithm and hybrid derivative method, *Journal of Chemical Industry and Engineering(China)*, **57 (06)**:1378-1384.