An LVI-based Numerical Algorithm for Solving Quadratic Programming Problems*

ZHANG Yunong[1,2†] LI Xuezhong[3] ZHANG Zhijun[1] LI Jun[1]

Abstract This paper presents and investigates a numerical algorithm (termed as 94LVI algorithm) for solving quadratic programming (QP) problems with linear equality and bound constraints. To do this, the constrained QP problems are firstly converted into linear variational inequalities (LVI), which are then converted into equivalent piecewise-linear projection equations (PLPE). After that, the resultant PLPE is solved by the presented 94LVI algorithm. The optimal numerical solutions to the QP problems are thus obtained. Furthermore, the theoretical proof of the global convergence of the 94LVI algorithm is presented. The numerical comparison results between the 94LVI algorithm and the active set algorithm are provided as well, which further demonstrates the efficacy and superiority of the presented algorithm for solving such QP problems.

Keywords numerical algorithm, quadratic programming, 94LVI algorithm, global convergence

Chinese Library Classification O221.2

2010 Mathematics Subject Classification 49M37

一种基于 LVI 求解二次规划问题的数值算法

张雨浓 [1,2†] 李学忠 [3] 张智军 [1] 李 钧 [1]

摘要 给出并研究了一种数值算法 (简称 94LVI 算法), 用于求解带等式和双端约束的二次规划问题. 这类带约束的二次规划问题首先被转换为线性变分不等式问题, 该问题等价于分段线性投影等式. 接着使用 94LVI 算法求解上述分段线性投影等式, 从而得到 QP 问题的最优解. 进一步给出了 94LVI 算法的全局收敛性证明. 94LVI 算法与经典有效集算法的对比实验结果证实了给出的 94LVI 算法在求解二次规划问题上的高效性与优越性.

关键词 数值算法, 二次规划, 94LVI 算法, 全局收敛性

中图分类号 O221.2

数学分类号 49M37

收稿日期: 2011 年 5 月 10 日.

* This work is supported by The National Natural Science Foundation of China (No. 61075121, 60935001), and also by the Fundamental Research Funds for the Central Universities of China (No. 3162460).

1. School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006; 中山大学信息科学与技术学院, 广州 510006

2. Research Institute of Sun Yat-sen University in Shenzhen, Shenzhen 518057; 中山大学深圳研究院, 深圳 518057

3. School of Software, Sun Yat-sen University, Guangzhou 510006; 中山大学软件学院, 广州 510006

† 通讯作者 Corresponding author

0 Introduction

The online solution of quadratic programming (QP) problems are widely encountered in various areas, e.g., optimal controller design[1], power-scheduling[2], robot-arm motion planning[3], and digital signal processing[4]. Motivated by engineering applications of QP in robotics[5−8], the following problem formulation is considered:

$$\begin{aligned} &\text{minimize} \quad x^{\mathrm{T}}Wx/2 + q^{\mathrm{T}}x, \\ &\text{subject to} \begin{cases} Jx = b, \\ x^{-} \leqslant x \leqslant x^{+}, \end{cases} \end{aligned} \tag{1}$$

where $x \in R^n$ is the decision vector to be obtained, superscript $^{\mathrm{T}}$ denotes the transpose of a vector/matrix, and $W \in R^{n\times n}$ is a positive-definite symmetric matrix. The other coefficient matrices and vectors are defined respectively as $q \in R^n$, $J \in R^{m\times n}$ and $b \in R^m$. The n-dimensional vectors x^- and x^+ denote respectively the lower and upper bounds of x.

In the past decades, large numbers of numerical algorithms have been proposed and developed for solving constrained optimization problems in the form of QP (1), such as Lagrange multiplier method[9−10], active set method[11], weighted-path-following interior-point algorithm[12], and projection and contraction algorithm[13]. Note that most of the numerical algorithms are based on the active set strategy, and MATLAB function "quadprog" integrates such an active set method. In this paper, such QP problems are firstly converted to linear variational inequalities (LVI), which are equivalent to piecewise-linear projection equations (PLPE). Secondly, a numerical algorithm (termed, 94LVI algorithm) is employed for solving the resultant PLPE, and the optimal numerical solutions to the QP problems subject to linear equality and bound constraints are thus obtained. Note that this work extends from the paper [14] which was published in 1994 to solve the LVI problems, and, for simplicity, the presented numerical algorithm is named the 94LVI algorithm. Numerical-experiment results substantiate the efficacy and accuracy of the 94LVI algorithm. Compared with the active set method, the 94LVI algorithm shortens the running time of the algorithm program for solving QP problems observably.

The remainder of this paper is organized into the following sections. The 94LVI algorithm for solving QP problems subject to linear equality and bound constraints is presented and discussed in Section 1. The global convergence of the 94LVI algorithm is proved in Section 2. The efficacy and accuracy of the 94LVI algorithm are further investigated via numerical experiments in Section 3. Finally, Section 4 concludes this paper with final remarks.

1 The 94LVI algorithm for QP solving

In the section, following the authors' previous design methods[15−16], an LVI-based numerical algorithm for solving QP (1) is developed.

Firstly, we can convert QP (1) into the linear variational inequality (LVI) problem：

to find a vector $y^* \in \Omega := \{y | y^- \leqslant y \leqslant y^+\} \subset R^{n+m}$ such that

$$(y - y^*)^{\mathrm{T}}(Hy^* + p) \geqslant 0, \quad \forall y \in \Omega, \tag{2}$$

where the primal-dual decision vector y and its upper and lower bounds $y^{\pm}$ are defined respectively as

$$y = \begin{bmatrix} x \\ u \end{bmatrix}, \ y^+ = \begin{bmatrix} x^+ \\ +\varpi 1_u \end{bmatrix}, \ y^- = \begin{bmatrix} x^- \\ -\varpi 1_u \end{bmatrix}, \tag{3}$$

with $1_u := [1, \ldots, 1]^{\mathrm{T}}$ denoting a vector composed of ones, of which the dimension is the same as that of vector u, and $\varpi \gg 0$ being a sufficiently large constant to replace $+\infty$ for numerical and implementation purposes. In addition, vector x is the primal/original decision vector of QP (1), and vector u is the dual decision vector defined corresponding to the equality constraint in QP (1). The coefficients in LVI (2) are defined as

$$H = \begin{bmatrix} W & -J^{\mathrm{T}} \\ J & 0 \end{bmatrix}, \ p = \begin{bmatrix} q \\ -b \end{bmatrix}.$$

According to [14-15], LVI (2) is equivalent to the following piecewise-linear projection equation (PLPE):

$$\mathrm{P}_{\Omega}(y - (Hy + p)) - y = 0, \tag{4}$$

where $\mathrm{P}_{\Omega}(\cdot) : R^{n+m} \to \Omega$ is a piecewise-linear projection operator with the ith projection element of $\mathrm{P}_{\Omega}(z)$ defined as

$$\begin{cases} y_i^-, & \text{if } z_i < y_i^-, \\ z_i, & \text{if } y_i^- \leqslant z_i \leqslant y_i^+, \ \forall i \in \{1, 2, \cdots, n+m\}. \\ y_i^+, & \text{if } z_i > y_i^+, \end{cases}$$

Secondly, we can define a set Ω^* to denote the solution set of LVI (2) and PLPE (4). To solve the problem, we can define the following vector-valued error function:

$$e(y) = y - \mathrm{P}_{\Omega}(y - (Hy + p)), \tag{5}$$

so that solving PLPE (4) is equivalent to finding a zero point of (5). As proposed in [14], let us define the state-vector in the kth iteration as $y^k = [(x^k)^{\mathrm{T}}, (u^k)^{\mathrm{T}}]^{\mathrm{T}}$. Given initial state $y^0 \in R^{n+m}$, for iteration index $k = 0, 1, 2, \cdots$, if $y^k \notin \Omega^*$, then we can employ the following 94LVI iteration formula for solving (4):

$$y^{k+1} = y^k - \rho(y^k) d(y^k), \tag{6}$$

where $d(y^k) = (H^{\mathrm{T}} + I)e(y^k)$ and $\rho(y^k) = \|e(y^k)\|_2^2 / \|d(y^k)\|_2^2$ with $\|\cdot\|_2$ denoting the two-norm of a vector.

Thirdly, for algorithm implementation purposes, we have the two-norm based scalar-valued function

$$\|e(y^k)\|_2 = \|y^k - \mathrm{P}_{\Omega}(y^k - (Hy^k + p))\|_2; \tag{7}$$

and, the control condition of the 94LVI algorithm implementation, which decides the accuracy of the solution, can be described as: "in every iteration-step of 94LVI algorithm, if the value of $\|e(y^k)\|_2$ is less than the preset accuracy value (PAV), the iteration will stop. The value of y^k that satisfies the PAV in the (current) kth iteration-step is regarded as a solution y^* of PLPE (4)". Note that the first n elements of y^* constitute the solution x^* to the original QP (1). It is worth mentioning that we also record the running time of the program and solution-errors in the program of the 94LVI algorithm, which are favorable for further demonstrating the efficacy and accuracy of the 94LVI algorithm.

2 Global convergence

By following [14][17][18], the important lemmas about the convergence of the 94LVI algorithm are cited and presented below (with themes similar to those in [14] but with proof details given according to the authors' engineering-type understanding, in addition to the consideration on results completeness and readers' convenience).

Lemma 1 $\forall y^* \in \Omega^*$, *the sequence* $\{y^k\}$ *(with iteration index* $k = 0, 1, 2, \cdots$*) generated by the 94LVI algorithm satisfies* $\|y^{k+1} - y^*\|_2^2 \leqslant \|y^k - y^*\|_2^2 - \rho(y^k)\|e(y^k)\|_2^2$.

Proof Referring to [14], using (6), we can obtain

$$\begin{aligned}\|y^{k+1} - y^*\|_2^2 &= \|y^k - y^* - \rho(y^k)d(y^k)\|_2^2 \\ &= [(y^k - y^*) - \rho(y^k)d(y^k)]^{\mathrm{T}}[(y^k - y^*) - \rho(y^k)d(y^k)] \\ &= ||y^k - y^*||_2^2 - (y^k - y^*)^{\mathrm{T}}\rho(y^k)d(y^k) - d^{\mathrm{T}}(y^k)\rho(y^k)(y^k - y^*) \\ &\quad + d^{\mathrm{T}}(y^k)\rho(y^k)\rho(y^k)d(y^k) \\ &= \|y^k - y^*\|_2^2 - 2\rho(y^k)(y^k - y^*)^{\mathrm{T}}d(y^k) + \rho^2(y^k)\|d(y^k)\|_2^2 \\ &= \|y^k - y^*\|_2^2 - 2\rho(y^k)(y^k - y^*)^{\mathrm{T}}(H^{\mathrm{T}} + I)e(y^k) + \rho(y^k)\|e(y^k)\|_2^2.\end{aligned}$$

Following Theorem 1 of [14], $(y^k - y^*)^{\mathrm{T}}(H^{\mathrm{T}} + I)e(y^k) \geqslant \|e(y^k)\|^2 \geqslant 0$, then we obtain

$$\begin{aligned}\|y^{k+1} - y^*\|_2^2 &\leqslant \|y^k - y^*\|_2^2 - 2\rho(y^k)\|e(y^k)\|_2^2 + \rho(y^k)\|e(y^k)\|_2^2 \\ &= \|y^k - y^*\|_2^2 - \rho(y^k)\|e(y^k)\|_2^2.\end{aligned} \tag{8}$$

The proof is thus completed.

Lemma 2 *The sequence* $\{y^k\}$ *(with iteration index* $k = 0, 1, 2, \cdots$*) generated by the 94LVI algorithm converges to a solution* y^*. *In addition, the first* n *elements of* y^* *constitute the optimal solution* x^* *to QP (1).*

Proof From (6), we can get

$$\begin{aligned}\rho(y^k) &= \|e(y^k)\|_2^2/\|d(y^k)\|_2^2 \\ &= \|e(y^k)\|_2^2/\|(H^{\mathrm{T}} + I)e(y^k)\|_2^2 \\ &\geqslant \|e(y^k)\|_2^2/(\|H^{\mathrm{T}} + I\|_2^2\|e(y^k)\|_2^2) \\ &= 1/\|H^{\mathrm{T}} + I\|_2^2 \\ &> 0.\end{aligned} \tag{9}$$

Combining inequalities (8) and (9), we can obtain

$$\|y^{k+1}-y^*\|_2^2 \leqslant \|y^k-y^*\|_2^2 - \|e(y^k)\|_2^2/\|H^{\mathrm{T}}+I\|_2^2 \leqslant \|y^k-y^*\|_2^2, \tag{10}$$

from which, we can get the following inequality:

$$\|e(y^k)\|_2^2/\|H^{\mathrm{T}}+I\|_2^2 \leqslant \|y^k-y^*\|_2^2 - \|y^{k+1}-y^*\|_2^2.$$

By defining $1/\|H^{\mathrm{T}}+I\|_2^2$ as ξ, for iteration index $k=0,1,2,\cdots$, we further obtain

$$\xi\sum_{k=0}^{+\infty}\|e(y^k)\|_2^2 \leqslant \|y^0-y^*\|_2^2 - \lim_{k\to+\infty}\|y^{k+1}-y^*\|_2^2 \leqslant \|y^0-y^*\|_2^2, \tag{11}$$

and, according to mathematical knowledge, there must exist a positive number h satisfying $0\leqslant h\leqslant\|y^0-y^*\|_2^2$ and

$$\lim_{j\to\infty}\xi\sum_{k=0}^{j}\|e(y^k)\|_2^2 = h.$$

Then, according the necessary condition of series convergence (i.e., a corollary from Cauchy criterion of series convergence), from (11) we can obtain

$$\lim_{k\to\infty}\|e(y^k)\|_2^2 = 0.$$

By defining $e(y^k) := [e_1(y^k), e_2(y^k), \cdots, e_{m+n}(y^k)]^{\mathrm{T}}$, then

$$\lim_{k\to\infty}\|e(y^k)\|_2^2 = \lim_{k\to\infty}[(e_1(y^k))^2+(e_2(y^k))^2+\cdots+(e_{m+n}(y^k))^2] = 0,$$

and it follows that

$$\lim_{k\to\infty}e_1(y^k)=0,\ \lim_{k\to\infty}e_2(y^k)=0,\ \cdots,\ \lim_{k\to\infty}e_{m+n}(y^k)=0. \tag{12}$$

From (12), we further obtain

$$\lim_{k\to\infty}e(y^k)=0.$$

Let y^* be a solution of PLPE (4), then the sequence $\{y^k\}$ has exactly a cluster point and

$$\lim_{k\to\infty}y^k = y^* \text{ with } e(y^*)=0.$$

In addition, the first n elements of y^* (i.e., y_1^*, y_2^*, $\cdots$, y_n^*) constitute the optimal solution x^* to the QP (1) in view of the conversion and equivalence of QP to the LVI and PLPE. The proof is thus completed.

3 Numerical-experiment results

To demonstrate the efficacy of the 94LVI algorithm, we implement this algorithm via both MATLAB and C programming languages to solve general QP problems. The numerical experiments are carried out in the MATLAB R2008a environment performed on a personal digital computer, which is equipped with a Pentium(R) Dual-Core E5700 3.00GHz CPU, 2GB DDR3 memory, and a Windows 7 Ultimate operating system. The output errors and the computing time of related algorithms are shown in this section.

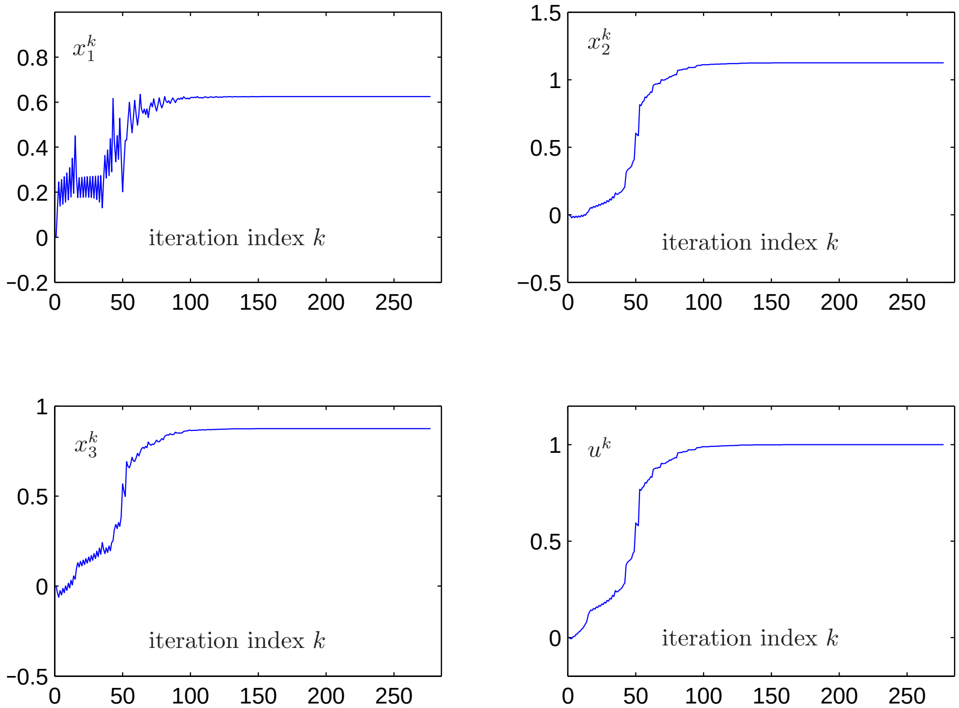

Figure 1: Solution trajectories of QP (13) starting from initial state $y^0 = [0, 0, 0, 0]^{\mathrm{T}}$

3.1 Efficacy verification

In this subsection, a numerical example is presented to demonstrate the effectiveness of the 94LVI algorithm for solving QP (1) subject to linear equality and bound constraints. That is, the following QP problem is considered:

$$\begin{aligned}&\text{minimize} \quad 10x_1^2 + x_2^2 + x_3^2 - 2x_1x_2 - 6x_1x_3 - 4x_1,\\ &\text{subject to} \begin{cases} x_1 + x_2 - 2x_3 = 0, \\ -3 \leqslant x_1, x_2, x_3 \leqslant 3. \end{cases}\end{aligned} \tag{13}$$

QP (13) can be rewritten in the compact matrix-vector form as (1), and thus we have the coefficient matrices/vectors as follows:

$$W = \begin{bmatrix} 20 & -2 & -6 \\ -2 & 2 & 0 \\ -6 & 0 & 2 \end{bmatrix}, \; q = \begin{bmatrix} -4 \\ 0 \\ 0 \end{bmatrix}, \; J = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}^{\mathrm{T}}, \; b = 0, \; x^+ = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix}, \text{ and } x^- = \begin{bmatrix} -3 \\ -3 \\ -3 \end{bmatrix}.$$

By employing the 94LVI algorithm program with preset accuracy value being 10^{-6} to solve this QP problem, numerical-experiment results can be summarized below.

- Figure 1 shows the solution trajectories of the 94LVI algorithm for solving QP (13) with initial state $y^0 = [(x^0)^{\mathrm{T}}, (u^0)^{\mathrm{T}}]^{\mathrm{T}}$ being $[0, 0, 0, 0]^{\mathrm{T}}$. As seen from Figure 1, through

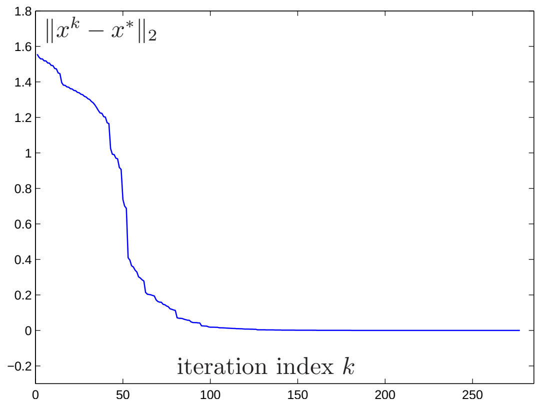

Figure 2: Solution-error trajectory of QP (13) starting with initial state $y^0 = [0,0,0,0]^{\mathrm{T}}$

277 iterations, state vector $y = [x^{\mathrm{T}}, u^{\mathrm{T}}]^{\mathrm{T}}$ can converge to the optimal solution $y^* = [5/8, 9/8, 7/8, 1]^{\mathrm{T}}$, of which the the first n elements of y^* constitute the QP's optimal solution $x^* = [5/8, 9/8, 7/8]^{\mathrm{T}}$. Figure 2 shows the trajectory of solution-error $\|x^k - x^*\|_2$ $(k = 0, 1, 2, \cdots)$, which converges to zero within 277 iterations. Note that, the solution-error at the final iteration [when employing the 94LVI algorithm to solve QP (13)] is about 6.694457×10^{-7} [i.e., less than PAV 10^{-6}], which validates the accuracy of the presented 94LVI algorithm for solving the QP problem.

- For comparison and illustration, the numerical-experiment results with a different initial state $y^0 = [2, 1/2, -2, 1]^{\mathrm{T}}$ is presented in Figures 3 and 4. From Figure 3, we can see that, through 266 iterations, state vector $y = [x^{\mathrm{T}}, u^{\mathrm{T}}]^{\mathrm{T}}$ converges to the optimal solution $y^* = [5/8, 9/8, 7/8, 1]^{\mathrm{T}}$, and thus $x^* = [5/8, 9/8, 7/8]^{\mathrm{T}}$ is the optimal solution of QP (13). Figure 4 shows the trajectory of corresponding solution-error $\|x^k - x^*\|_2$, which converges to zero within 266 iterations. Note that the solution-error at the final iteration is about 6.665722×10^{-7}. The above numerical-experiment results further demonstrate the efficacy and accuracy of the 94LVI algorithm for solving the QP problem.

In summary, the above illustrative example substantiates well the efficacy and accuracy of the 94LVI algorithm for solving such a QP problem.

3.2 Comparative experiments

In this subsection, with coefficients randomly generated, a series of QP problems in the form of QP (1) are solved via the active set algorithm and the 94LVI algorithm. Moreover, both MATLAB and C version programs of the 94LVI algorithm are developed and employed to solve such QPs. The computational time τ_{A} of the active set algorithm, as well as the computational time τ_{94LVIM}, τ_{94LVIC} and output errors $\epsilon_{\mathrm{94LVIM}}$, $\epsilon_{\mathrm{94LVIC}}$ of the 94LVI algorithm per problem, are recorded. Here, ϵ is defined as $\|e(y^k)\|_2$ [i.e. (7)] of the final

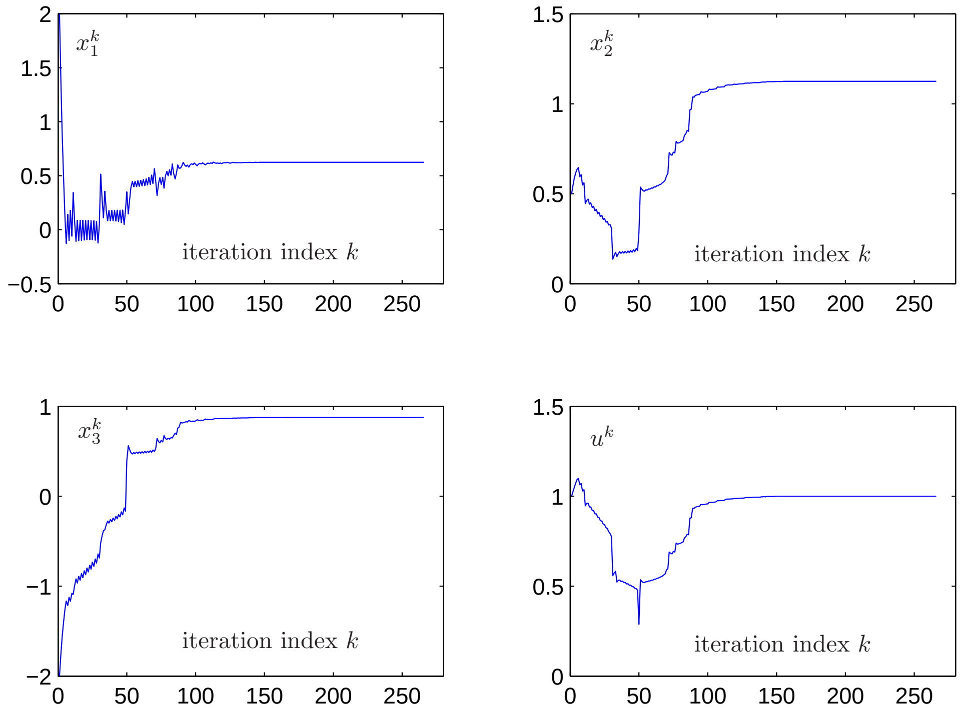

Figure 3: Solution trajectories of QP (13) starting from initial state $y^0 = [2, 1/2, -2, 1]^{\mathrm{T}}$

iteration, while subscripts $_{\mathrm{A}}$, $_{\mathrm{94LVIM}}$ and $_{\mathrm{94LVIC}}$ denote the active set algorithm, the 94LVI algorithm implemented in MATLAB and C, respectively. In addition, the average time and errors of the experiments are recorded for further comparison.

Firstly, we show 10 comparison results synthesized by the active set algorithm and the MATLAB version 94LVI algorithm in Table 1. The values of the coefficient matrices/vectors W, q, J, b, x^-, x^+ of the 10 QP-problems are all randomly generated through MATLAB function "rand( )". In this case, we set $n = 3$ and $m = 1$, with the preset accuracy value being 10^{-3}. As seen from Table 1, for such 10 randomly-generated QP problems, compared with the traditional active set algorithm, by adopting the MATLAB version 94LVI algorithm, the running time is shortened by about 20 times. In addition, all errors of final solutions are less than the preset accuracy value.

Secondly, we also show 10 numerical-experiment results using C program in Table 1. The coefficient matrices/vectors of these 10 QP-problems are set the same as above. As seen from Table 1, the runtime of the C version 94LVI algorithm is about 8 times in average shorter than the MATLAB version one. In addition, all the errors of the solutions synthesized by the C version 94LVI are less than the PAV (i.e. 10^{-3}) again.

In summary, the above numerical-experiment results sufficiently substantiate the efficacy and accuracy of the 94LVI algorithm for solving such QP problems. Compared to the active set algorithm, the 94LVI algorithm can obtain better performance.

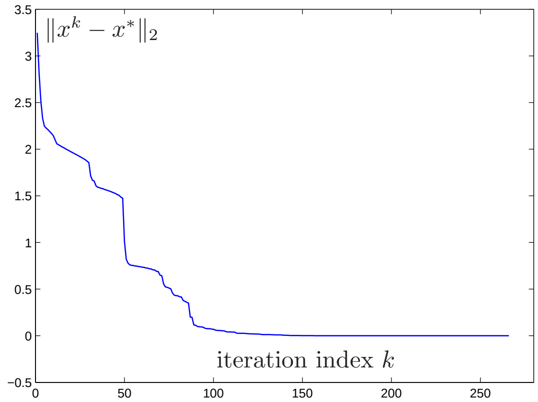

Figure 4: Solution-error trajectory of QP (13) with initial state $y^0 = [2, 1/2, -2, 1]^{\mathrm{T}}$

Table 1 The running time of the active set algorithm and 94LVI algorithm and the solution-errors of 94LVI programs (i.e., 94LVIM and 94LVIC programs)

	τ_{A} (s)	τ_{94LVIM} (s)	$\epsilon_{\mathrm{94LVIM}}$	τ_{94LVIC} (s)	$\epsilon_{\mathrm{94LVIC}}$
1	0.715455	0.028507	9.599926×10^{-4}	0.003365	9.513458×10^{-4}
2	0.614927	0.030423	9.606960×10^{-4}	0.003456	9.917611×10^{-4}
3	0.616497	0.023424	9.463763×10^{-4}	0.003546	9.068035×10^{-4}
4	0.356106	0.016992	9.869118×10^{-4}	0.002908	9.869118×10^{-4}
5	0.600025	0.019868	8.605755×10^{-4}	0.003250	9.904625×10^{-4}
6	0.607770	0.023295	7.155840×10^{-4}	0.003017	4.518490×10^{-4}
7	0.616976	0.029874	9.936125×10^{-4}	0.003408	9.936125×10^{-4}
8	0.604829	0.029088	9.850526×10^{-4}	0.004373	9.656885×10^{-4}
9	0.614515	0.019124	6.942093×10^{-4}	0.003064	9.810901×10^{-4}
10	0.600338	0.026207	9.925356×10^{-4}	0.003029	9.962816×10^{-4}
average	0.594744	0.024680	9.095546×10^{-4}	0.003342	9.215806×10^{-4}

4 Conclusions

In this paper, a numerical quadratic-programming algorithm (i.e., the 94LVI algorithm) has been presented and investigated to solve quadratic programming problems subject to linear equality and bound constraints. The global convergence of the 94LVI algorithm has been proved. Moreover, the numerical-experiment results have demonstrated the efficacy, accuracy and superiority of the presented 94LVI algorithm for QP solving. To further demonstrate the advantages, 10 more comparison results between the active set algorithm and the 94LVI algorithm have been provided, illustrating the high accuracy and faster convergence of the 94LVI algorithm.

References

[1] Johansen T A, Fossen T I, Berge S P. Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming[J]. *IEEE Transactions on Control Systems Technology*, 2004, **12**(1): 211-216.

[2] Grudinin N. Reactive power optimization using successive quadratic programming method[J]. *IEEE Transactions on Power Systems*, 1998, **13**(4): 1219-1225.

[3] Wang J, Zhang Y N. Recurrent neural network for real-time computation of inverse kinematics of redundant manipulators[M]. *Machine Intelligence: Quo-Vadis?* (eds: P. Sincak, J. Vascak, and K. Hirota), World Scientific, Singapore, 2004, 299-319.

[4] Leithead W E, Zhang Y N. O(N^2)-operation approximation of covariance matrix inverse in gaussian process regression based on Quasi-Newton BFGS method[J]. *Communications in Statistics – Simulation and Computation*, 2007, **36**(2): 367-380.

[5] Zhang Y N. Towards piecewise-linear primal neural networks for optimization and redundant robotics[C]//MengChu Zhou, *Proceedings of IEEE International Conference on Networking, Sensing and Control*, Florida: IEEE Press, 2006, 374-379.

[6] Zhang Y N, Ge S S, Lee T H. A unified quadratic programming based dynamical system approach to joint torque optimization of physically constrained redundant manipulators[J]. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B, 2004, **34**(5): 2126-2132.

[7] Zhang Y N, Wang J. A dual neural network for constrainted torque optimization of kinematically redundant manipulators[J]. *IEEE Transactions on Systems, Man and Cybernetics*, Part B, 2002, **32**(5): 654-662.

[8] Zhang Y N, Wang J. Obstacle avoidance of kinematically redundant manipulators using a dual neural network[J]. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B, 2004, **34**(1): 752-759.

[9] Powell M J D. A method for nonlinear constraints in minimization problems[M]. *Optimization* (ed: R. Fletcher), Academic Press, London, 1969.

[10] Hestens M R. Multiplier and Gradient Methods[J]. *Journal of optimization theory and applicatuins*, 1969, **4**(5): 303-320.

[11] Fletcher R. A general quadratic programming algorithm[J]. *IMA Journal of Applied Mathematics*, 1971, **7**(1): 76-91.

[12] Jin Z J, Bai Y Q, Han B S. A weighted-path-following interior-point algorithm for convex quadratic optimization[J]. *Operations Research Transactions*, 2010, **14**(1): 55-65.

[13] Zhao S F, Fei P S, Li J. A projection and contraction method for convex quadratic programming[J]. *Journal of Wuhan University*, 2001, **47**(1): 22-24.

[14] He B S. A new method for a class of linear variational inequalities[J]. *Mathematical Programming*, 1994, **66**(1-3): 137-144.

[15] Zhang Y N. On the LVI-based primal-dual neural network for solving online linear and quadratic programming problems[C]//Suhada Jayasuriya, *Proceedings of American Control Conference*, Portland: IEEE Press, 2005, 1351-1356.

[16] Zhang Y N, Cai B H, Zhang L, Li K N. Bi-criteria velocity minimization of robot manipulators using a linear variational inequalityes-based primal-dual neural network and PUMA560 example[J]. *Advanced Robotics*, 2008, **22**(13-14): 1479-1496.

[17] He B S. Solving a class of linear projection equations[J]. *Numerische Mathematik*, 1994, **68**(1): 71-80.

[18] He B S. A projection and contraction method for a class of linear complementarity problem and its application in convex quadratic programming[J]. *Applied Mathematics and Optimization*, 1992, **25**(3): 247-262.