

An efficient optimization algorithm for quadratic programming problem and its applications to mobile robot path planning

Lei Cai¹, Juanjuan Yang², Li Zhao³ and Lan Wu⁴

Abstract

The quadratic programming problem has broad applications in mobile robot path planning. This article presents an efficient optimization algorithm for globally solving the quadratic programming problem. By utilizing the convexity of univariate quadratic functions, we construct the linear relaxation programming problem of the quadratic programming problem, which can be embedded within a branch-and-bound structure without introducing new variables and constraints. In addition, a new pruning technique is inserted into the branch-and-bound framework for improving the speed of the algorithm. The global convergence of the proposed algorithm is proved. Compared with some known algorithms, numerical experiment not only demonstrates the higher computational efficiency of the proposed algorithm but also proves that the proposed algorithm is an efficient approach to solve the problems of path planning for the mobile robot.

Keywords

Quadratic programming problem, global optimization, linear relaxation programming, pruning technique, branch-and-bound optimization algorithm, path planning

Date received: 5 July 2017; accepted: 27 October 2017

Topic: Special Issue - Intelligent Control Methods in Advanced Robotics and Automation

Topic Editor: Andrey V Savkin

Associate Editor: Junzhi Yu

Introduction

The mobile robot has drawn more and more attention in scientific areas and engineering applications. One of the fundamental issues in mobile robot performing tasks in an unknown environment is the path planning resolution problem. That is, given the desired task, arriving at the destination or catching a target with the shortest distance, (or spending the least amount of energy), how can the mobile robot search from the environment to start from the initial position to achieve its own purpose by the best or suboptimal path according to some performance indicators? In the path planning, navigation involves a series of common problems, with collision avoidance in some form being almost universally needed, which underlaid by more general assumptions about the problem would be considered superior.¹ In dealing with aircraft

(autonomous robot) navigation problems, second-order central difference filtering algorithm² and the new method based on wavelet multiresolution analysis and adaptive

¹School of Information Engineering, Henan Institute of Science and Technology, Xinxiang, China

²School of Mathematical Sciences, Henan Institute of Science and Technology, Xinxiang, China

³School of Mechanical and Power Engineering, Henan Polytechnic University, Jiaozuo, China

⁴College of Electrical Engineering, Henan University of Technology, Zhengzhou, China

Corresponding author:

Lei Cai, School of Information Engineering, Henan Institute of Science and Technology, Xinxiang 453003, China.

Email: cailei2014@126.com



Kalman filter³ are proposed to perform better than the classical.

The most challenging problem for path planning is to arrive at the destination or catch a target with the shortest

distance or least consumption of fuel when real time is demanded in dynamic environments involving multiple obstacles. The quadratic programming model is formulated as follows

$$\left\{ \begin{array}{l} \min J = \frac{1}{2}x^T(\omega_1 Q - \omega_2 H)x + (\omega_1 x^T - \omega_2 f^T)x \\ \text{s.t. } \tau_i \gamma_{\text{AOC}i(k)} \leq \tau_i (\gamma_{\text{AOC}i(k)} + \Delta \gamma_{\text{AOC}i(k)}) \leq \pi \\ \max \left[-\Delta_{\max}, -\left(V_{A\max} + \frac{\nu_{Ax}(k)}{T} \right) \right] \leq a_x \leq \min \left[\Delta_{\max}, \left(V_{A\max} - \frac{\nu_{Ay}(k)}{T} \right) \right] \\ \max \left[-\Delta_{\max}, -\left(V_{A\max} + \frac{\nu_{Ay}(k)}{T} \right) \right] \leq a_y \leq \min \left[\Delta_{\max}, \left(V_{A\max} - \frac{\nu_{Az}(k)}{T} \right) \right] \\ \max \left[-\Delta_{\max}, -\left(V_{A\max} + \frac{\nu_{Az}(k)}{T} \right) \right] \leq a_z \leq \min \left[\Delta_{\max}, \left(V_{A\max} - \frac{\nu_{Ax}(k)}{T} \right) \right] \\ \sin\theta_m \cos\varphi_n a_x + \sin\theta_m \sin\varphi_n a_y + \cos\varphi_m a_z \leq \Delta_{\max} \end{array} \right.$$

where $Q \in R_{n \times n}$ is semidefinite matrix; $c \in R_{n \times 1}$ is coefficient vectors and $H \in R_{n \times 1}$ is right-hand side vectors; $x \in R_{n \times 1}$ is variable vector; ω_1 and ω_2 are the weights used to balance the relative distance and the optimal velocity direction; τ_i is assigned to evaluate the threat imposed by an obstacle on the robot; the angle between V_{AO} and L_{AO} is defined as obstacle angle, denoted by $\gamma_{\text{AO}i(k)}$, V_{AO} denotes the vehicle velocity relative to the obstacle, and L_{AO} denotes the vehicle position relative to the obstacle; $\gamma_{\text{AOC}i(k)}$ is the collision region angle in time step $i(k)$, $\gamma_{\text{AO}i(k)}$ represents the obstacle angle in step $i(k)$ after robot's movement, I denotes the label of the obstacle; Δ_{\max} represents the upper bound of acceleration and $V_{A\max}$ stands for the upper bound of the robot velocity; a_x , a_y and a_z represent the acceleration elements of the mobile robot; T is the planning period; and k is the time step. Such problems can be expressed as the following general quadratic programming problem (QPP)

$$(\text{QPP}) : \left\{ \begin{array}{l} \min G_0(y) = \sum_{j=1}^n \sum_{k=1}^n \beta_{jk}^0 y_j y_k + \sum_{k=1}^n \alpha_k^0 y_k \\ \text{s.t. } G_i(y) = \sum_{j=1}^n \sum_{k=1}^n \beta_{jk}^i y_j y_k + \sum_{k=1}^n \alpha_k^i y_k \leq b_i, \quad i = 1, \dots, m \\ y \in Y^0 = \{y \in R^n : 1^0 \leq y \leq u^0\} \end{array} \right.$$

where $\beta_{jk}^i, \alpha_k^i, b_i \in R, i = 0, 1, \dots, m, j = 1, \dots, n, k = 1, \dots, n; 1^0 = (1_1^0, \dots, 1_n^0)^T, u^0 = (u_1^0, \dots, u_n^0)^T$.

The QPPs have a wide variety of applications in information science, control science and engineering, management science, finance and economy, and so on.⁴⁻⁷ Except for the above reviewed applications, the QPP presents important

theoretical and computational difficulties since the kind of problems possess many local optimum points which are not global optimal. So, the QPP has attracted attention from many scientists. In past several decades, many special optimization algorithms have been proposed for solving the special forms of the QPP, such as reformulation convexification,⁸ Lagrangian underestimate and interval Newton method,⁹ semidefinite relaxation,¹⁰ polyhedral approximation,¹¹ duality bound,¹² robust solution,¹³ parametric relaxation,¹⁴ branch and bound,¹⁵⁻¹⁷ and so on. Although some known algorithms can also be used to compute the QPP, it is rather challenging to globally compute the QPP because of its complication.¹⁸⁻²⁰

In this article, we will present a branch-and-bound algorithm for globally solving the QPP. To accomplish this goal, by utilizing the convexity of univariate quadratic functions, we construct the linear relaxation programming problem (LRPP) of the QPP, which can be embedded within a branch-and-bound framework without introducing new variables and constraints. In addition, a new pruning technique is inserted into the branch-and-bound framework for improving the convergence of the algorithm. Next, by combining the linear relaxation bounding operation with the bisection rule and pruning operation, an efficient branch-and-bound optimization algorithm is presented for globally solving the QPP. Finally, compared with the known algorithms, numerical experiments demonstrate the higher computational efficiency of the proposed method.

The remaining sections of this article are listed as follows. By utilizing the convexity of univariate quadratic functions, "new linear relaxation programming" section derives a

novel linear relaxation technique, and then the LRPP of the QPP is established. Based on the LRPP derived in “new linear relaxation programming” section, “optimization algorithm and its global convergence” section presents an efficient branch-and-bound optimization algorithm by combining the linear relaxation bounding operation with the bisection rule and pruning operation, and its global convergence is proved. In “numerical experiments” section, in order to compare with some known algorithms, numerical experiments are given to show the computational superiority of the proposed algorithm. Finally, some concluding remarks are discussed.

New linear relaxation programming

In this section, we derive a new linear relaxation technique, based on the relaxation technique, on which the LRPPs of the initial problem and its partitioned subproblems can be established, and the detailed deriving process is demonstrated as follows.

For any $y \in Y = [l, u] \subseteq Y^0$, by the convexity of univariate quadratic function y_j^2 , $j = 1, 2, \dots, n$, we have

$$(l_j + u_j)y_j - \frac{(l_j + u_j)^2}{4} \leq y_j^2 \leq (l_j + u_j)y_j - l_j u_j \quad (1)$$

Similarly, for any $y \in Y = [l, u] \subseteq Y^0$, we can easily derive the following inequalities

$$\begin{cases} (l_j + u_j + l_k + u_k)(y_j + y_k) - \frac{(l_j + u_j + l_k + u_k)^2}{4} \leq (y_j + y_k)^2 \\ (y_j + y_k)^2 \leq (l_j + u_j + l_k + u_k)(y_j + y_k) - (l_j + l_k)(u_j + u_k) \end{cases} \quad (2)$$

By equations (1) and (2), for any $y \in Y = [l, u] \subseteq Y^0$, we can easily derive the following inequalities

$$\begin{aligned} y_j y_k &= \frac{1}{2} [(y_j + y_k)^2 - y_j^2 - y_k^2] \\ &\geq \frac{1}{2} [(l_j + u_j + l_k + u_k)(y_j + y_k) \\ &\quad - \frac{(l_j + u_j + l_k + u_k)^2}{4}] \\ &\quad - \frac{1}{2} [(l_j + u_j)y_j - l_j u_j + (l_k + u_k)y_k - l_k u_k] \\ &= \frac{1}{2} [(l_j + u_j)y_k + (l_k + u_k)y_j + l_j u_j + l_k u_k \\ &\quad - \frac{(l_j + u_j + l_k + u_k)^2}{4}] \end{aligned}$$

and

$$\begin{aligned} y_j y_k &= \frac{1}{2} [(y_j + y_k)^2 - y_j^2 - y_k^2] \\ &\leq \frac{1}{2} [(l_j + u_j + l_k + u_k)(y_j + y_k) - (l_j + l_k)(u_j + u_k)] \\ &\quad - \frac{1}{2} [(l_j + u_j)y_j - \frac{(l_j + u_j)^2}{4} + (l_k + u_k)y_k \\ &\quad - \frac{(l_k + u_k)^2}{4}] \\ &= \frac{1}{2} [(l_j + u_j)y_k + (l_k + u_k)y_j + \frac{(l_j + u_j)^2}{4} + \frac{(l_k + u_k)^2}{4} \\ &\quad - (l_j + l_k)(u_j + u_k)] \end{aligned}$$

Let

$$\begin{cases} f_j^l(y) = (l_j + u_j)y_j - \frac{(l_j + u_j)^2}{4} \\ f_j^u(y) = (l_j + u_j)y_j - l_j u_j \\ f_{jk}^l(y) = \frac{1}{2} [(l_j + u_j)y_k + (l_k + u_k)y_j + l_j u_j + l_k u_k \\ \quad - \frac{(l_j + u_j + l_k + u_k)^2}{4}] \\ f_{jk}^u(y) = \frac{1}{2} [(l_j + u_j)y_k + (l_k + u_k)y_j + \frac{(l_j + u_j)^2}{4} + \frac{(l_k + u_k)^2}{4} \\ \quad - (l_j + l_k)(u_j + u_k)] \end{cases}$$

Obviously, we have

$$f_j^l(y) \leq y_j^2 \leq f_j^u(y), \quad f_{jk}^l(y) \leq y_j y_k \leq f_{jk}^u(y)$$

Since

$$y_j^2 - f_j^l(y_j) = y_j^2 - \left[(l_j + u_j)y_j - \frac{(l_j + u_j)^2}{4} \right]$$

is a convex function about y_j over the interval $[l_j, u_j]$. Thus, at the point l_j or u_j , it can obtain the maximum value

$$\max[y_j^2 - f_j^l(y_j)] = \frac{(u_j - l_j)^2}{4}$$

Similarly, since

$$f_j^u(y_j) - y_j^2 = (l_j + u_j)y_j - l_j u_j - y_j^2$$

is a concave function about y_j over the interval $[l_j, u_j]$, therefore, at the point $\frac{l_j + u_j}{2}$, $[f_j^u(y_j) - y_j^2]$ can obtain maximum value

$$\max[f_j^u(y_j) - y_j^2] = \frac{(u_j - l_j)^2}{4}$$

Similarly, for any $y \in Y = [l, u] \subseteq Y^0$, then $(y_j + y_k) \in [l_j + l_k, u_j + u_k]$, we have

$$\begin{cases} \max \left\{ (y_j + y_k)^2 - \left[(l_j + u_j + l_k + u_k)(y_j + y_k) - \frac{(l_j + u_j + l_k + u_k)^2}{4} \right] \right\} = \frac{(u_j + u_k - l_j - l_k)^2}{4} \\ \max \left\{ [(l_j + u_j + l_k + u_k)(y_j + y_k) - (l_j + l_k)(u_j + u_k)] - (y_j + y_k)^2 \right\} = \frac{(u_j + u_k - l_j - l_k)^2}{4} \end{cases}$$

Obviously, when $\|u - l\| \rightarrow 0$, we have

$$\begin{cases} \max[y_j^2 - f_j^l(y_j)] \rightarrow 0 \\ \max[f_j^u(y_j) - y_j^2] \rightarrow 0 \\ \max \left\{ (y_j + y_k)^2 - \left[(l_j + u_j + l_k + u_k)(y_j + y_k) - \frac{(l_j + u_j + l_k + u_k)^2}{4} \right] \right\} \rightarrow 0 \\ \max \left\{ [(l_j + u_j + l_k + u_k)(y_j + y_k) - (l_j + l_k)(u_j + u_k)] - (y_j + y_k)^2 \right\} \rightarrow 0 \end{cases} \quad (3)$$

Since

$$\begin{aligned} y_j y_k - f_{jk}^l(y) &= y_j y_k - \left\{ \frac{1}{2} [(l_j + u_j)y_k + (l_k + u_k)y_j + l_j u_j + l_k u_k - \frac{(l_j + u_j + l_k + u_k)^2}{4}] \right\} \\ &= \frac{1}{2} [(y_j + y_k)^2 - y_j^2 - y_k^2] - \left\{ \frac{1}{2} [(l_j + u_j)y_k + (l_k + u_k)y_j + l_j u_j + l_k u_k - \frac{(l_j + u_j + l_k + u_k)^2}{4}] \right\} \\ &= \frac{1}{2} \left\{ (l_j + u_j)y_j - l_j u_j - y_j^2 + \frac{1}{2} (l_k + u_k)y_k - l_k u_k - y_k^2 \right\} \\ &\quad + \frac{1}{2} \left\{ (y_j + y_k)^2 - \left[(l_j + u_j + l_k + u_k)(y_j + y_k) - \frac{(l_j + u_j + l_k + u_k)^2}{4} \right] \right\} \\ &\leq \frac{1}{2} \max[f_j^u(y_j) - y_j^2] + \frac{1}{2} \max[f_k^u(y_k) - y_k^2] \\ &\quad + \frac{1}{2} \max \left\{ (y_j + y_k)^2 - \left[(l_j + u_j + l_k + u_k)(y_j + y_k) - \frac{(l_j + u_j + l_k + u_k)^2}{4} \right] \right\} \end{aligned}$$

By equation (3), we have $y_j y_k - f_{jk}^l(y) \rightarrow 0$ as $\|u - l\| \rightarrow 0$.

Similarly, we can prove that $f_{jk}^u(y) - y_j y_k \rightarrow 0$ as $\|u - l\| \rightarrow 0$.

Let

$$\begin{aligned} G_i^L(y) &= \sum_{k=1}^n \alpha_k^i y_k + \sum_{k=1, \beta_{kk}^i > 0}^n \beta_{kk}^i f_k^l(y_k) + \sum_{k=1, \beta_{kk}^i < 0}^n \beta_{kk}^i f_k^u(y_k) \\ &\quad + \sum_{j=1}^n \sum_{k=1, k \neq j, \beta_{jk}^i > 0}^n \beta_{jk}^i f_{jk}^l(y) + \sum_{j=1}^n \sum_{k=1, \beta_{jk}^i < 0}^n \beta_{jk}^i f_{jk}^u(y) \\ G_i^U(y) &= \sum_{k=1}^n \alpha_k^i y_k + \sum_{k=1, \beta_{kk}^i < 0}^n \beta_{kk}^i f_k^l(y_k) + \sum_{k=1, \beta_{kk}^i > 0}^n \beta_{kk}^i f_k^u(y_k) \\ &\quad + \sum_{j=1}^n \sum_{k=1, k \neq j, \beta_{jk}^i < 0}^n \beta_{jk}^i f_{jk}^l(y) + \sum_{j=1}^n \sum_{k=1, \beta_{jk}^i > 0}^n \beta_{jk}^i f_{jk}^u(y) \end{aligned}$$

Obviously, by equation (3), we can derive that as $\|u - l\| \rightarrow 0$

$$G_i^U(y) \geq G_i(y) \geq G_i^L(y), \quad G_i(y) - G_i^L(y) \rightarrow 0 \quad \text{and} \quad G_i^U(y) - G_i(y) \rightarrow 0.$$

Based on the above linearizing process, we can derive the LRPP of the QPP over Y as follows

$$\text{LRPP}(Y) : \begin{cases} \min & G_0^L(y) \\ \text{s. t.} & G_i^L(y) \leq b_i, \quad i = 1, \dots, m \\ & y \in Y = \{y : l \leq y \leq u\} \subseteq Y^0 \end{cases}$$

where

$$G_i^L(y) = \sum_{k=1}^n \alpha_k^i y_k + \sum_{k=1, \beta_{kk}^i > 0}^n \beta_{kk}^i f_k^l(y_k) + \sum_{k=1, \beta_{kk}^i < 0}^n \beta_{kk}^i f_k^u(y_k) \\ + \sum_{j=1}^n \sum_{k=1, k \neq j, \beta_{jk}^i > 0} \beta_{jk}^i f_{jk}^l(y) + \sum_{j=1}^n \sum_{k=1, k \neq j, \beta_{jk}^i < 0} \beta_{jk}^i f_{jk}^u(y).$$

By the above linearizing process, for any $Y \subseteq Y^0$, every feasible point of the QPP is also feasible to the LRPP, the LRPP can provide a reliable lower bound for the global optimum value of the QPP.

Optimization algorithm and its global convergence

In this section, we first introduce several fundamental techniques; next, by combining these techniques, we present an efficient branch-and-bound optimization algorithm for globally solving the QPP. The detailed contents are given as follows.

Several fundamental techniques

Firstly, we introduce a bisection rule as partitioning method. This selected bisection rule is exhaustiveness, which can guarantee the global convergence of the proposed optimization algorithm. For any given subbox, $Y^k = [l^k, u^k] \subseteq Y^0$. This bisection rule is as follows.

Assume that $\eta \in \operatorname{argmax}\{u_i - l_i | i = 1, \dots, n\}$, then Y^k can be partitioned into two subboxes: $Y^{k,1} = \{y \in R^n | l_i \leq y_i \leq u_i, i \neq \eta; l_\eta \leq y_\eta \leq \frac{l_\eta + u_\eta}{2}\}$ and $Y^{k,2} = \{y \in R^n | l_i \leq y_i \leq u_i, i \neq \eta; \frac{l_\eta + u_\eta}{2} \leq y_\eta \leq u_\eta\}$.

Secondly, to improve the convergent efficiency of the proposed branch-and-bound optimization algorithm, we introduce a pruning technique for deleting a part of the investigated box Y , which does not contain any global optimum point of the QPP. Without loss of generality, for any $y \in Y = (Y_j)_{n \times 1}$ with $Y_j = [l_j, u_j]$ ($j = 1, \dots, n$), we assume that

$$G_i^L(y) = \sum_{j=1}^n \gamma_{ij} y_j + \eta_i, \quad i = 0, 1, \dots, m$$

Assume that UB_k be a currently known upper bound of the proposed optimization algorithm at k th iteration, by utilizing the structure of the branch-and-bound optimization algorithm, similar as Voorhis,¹¹ for any subbox $Y = (Y_j)_{n \times 1} \subseteq Y^0$, we can get the following conclusions.

- i. If $\sum_{j=1}^n \min\{\gamma_{0j} l_j, \gamma_{0j} u_j\} + \eta_0 > UB_k$, then the subbox Y should be deleted; else if $\sum_{j=1}^n \min\{\gamma_{0j} l_j, \gamma_{0j} u_j\} + \eta_0 \leq UB_k$, then for each $p \in \{1, 2, \dots, n\}$, we have
If $\gamma_{0p} > 0$, then the interval $\left(\frac{UB_k - (\sum_{j=1}^n \min\{\gamma_{0j} l_j, \gamma_{0j} u_j\} + \eta_0) + \min\{\gamma_{0p} l_p, \gamma_{0p} u_p\}}{\gamma_{0p}}, u_p \right) \cap Y_p$

should be deleted;

If $\gamma_{0p} < 0$, then the interval

$$\left[l_p, \frac{UB_k - (\sum_{j=1}^n \min\{\gamma_{0j} l_j, \gamma_{0j} u_j\} + \eta_0) + \min\{\gamma_{0p} l_p, \gamma_{0p} u_p\}}{\gamma_{0p}} \right) \cap Y_p$$

should be deleted.

- ii. If $\sum_{j=1}^n \min\{\gamma_{ij} l_j, \gamma_{ij} u_j\} + \eta_i > b_i$ for some $i \in \{1, \dots, m\}$, then the subbox Y should be deleted; else if $\sum_{j=1}^n \min\{\gamma_{ij} l_j, \gamma_{ij} u_j\} + \eta_i \leq b_i$ for some $i \in \{1, \dots, m\}$, then for each $p \in \{1, \dots, n\}$, we have

If $\gamma_{ip} > 0$, then the interval

$$\left(\frac{b_i - (\sum_{j=1}^n \min\{\gamma_{ij} l_j, \gamma_{ij} u_j\} + \eta_i) + \min\{\gamma_{ip} l_p, \gamma_{ip} u_p\}}{\gamma_{ip}}, u_p \right) \cap Y_p$$

should be deleted.

If $\gamma_{ip} < 0$, then the interval

$$\left[l_p, \frac{b_i - (\sum_{j=1}^n \min\{\gamma_{ij} l_j, \gamma_{ij} u_j\} + \eta_i) + \min\{\gamma_{ip} l_p, \gamma_{ip} u_p\}}{\gamma_{ip}} \right) \cap Y_p$$

should be deleted.

From the above conclusions, we can construct the new pruning technique to prune a part of the investigated box which does not contain the global optimal point of the QPP, so that the computational speed of the proposed branch-and-bound optimization algorithm can be improved and accelerated.

Thirdly, the bounding process is needed to update the lower bounds and upper bounds of the optimal value of the QPP. Here, we update the lower bounds by solving the former LRPP using simple method. At the same time, we update the upper bounds by computing the objective function value of feasible points for the QPP.

Fourthly, denote $LB(Y^k)$ and $y^k = y(Y^k)$ as the global optimum value and the global optimum solution for the LRPP in the subbox Y^k , respectively. Based on the above bisection rule, the pruning technique and bounding process, we design the following branch-and-bound optimization algorithm for the QPP. The detailed steps of algorithm are given as follows.

Steps for optimization algorithm are follows:

Step 1. Initialize $k = 0$, $\Lambda_0 = \{Y^0\}$, $F = \emptyset$, $\varepsilon > 0$, $UB_0 = +\infty$.

Solve the LRPP (Y^0) to obtain $LB_0 = LB(Y^0)$ and $y^0 = y(Y^0)$. If y^0 is satisfied with the feasibility of the QPP, then let $F = \{y^0\}$ and $UB_0 = G_0(y^0)$.

If $UB_0 - LB_0 \leq \varepsilon$, then the algorithm terminates; at the same time, we get the global optimal solution y^0 of the QPP. Otherwise, continue to step 2.

Step 2. For the investigated box $Y^k \in \Lambda_k$, using the proposed bisection method, partition the box Y^k into two subboxes and denote the new subboxes set by \check{Y}^k .

Step 3. For each subbox $Y \in \check{Y}^k$, using the former pruning technique to delete a part of the investigated box which does not contain the global optimal point of the QPP¹¹, and let the remaining subbox be Y and the remaining partitioning set be \check{Y}^k .

Step 4. For each subbox $Y \in \check{Y}^k$, solve the LRPP (Y) to get $LB(Y)$ and $y(Y)$; if $LB(Y) > UB_k$, then let $\check{Y}^k = \check{Y}^k \setminus Y$. Otherwise, we detect the feasibilities of the midpoint y^{mid} and $y(Y)$ and update the feasible point set. At the same time, update $UB_k = \min_{y \in F} G_0(y)$ and denote the current best feasible solution by $y^k = \operatorname{argmin}_{y \in F} G_0(y)$. Let $\Lambda_k = (\Lambda_k \setminus Y^k) \cup \check{Y}^k$ and $LB_k = \inf_{Y \in \Lambda_k} LB(Y)$.

Step 5. If $UB_k - LB_k \leq \varepsilon$, then the algorithm terminates; UB_k and y^k are the global optimum value and the global optimum point for the QPP, respectively. Otherwise, let $k = k + 1$, and select $Y^k = \operatorname{argmin}_{Y \in \Lambda_k} LB(Y)$, and return to step 2.

Global convergence of the algorithm

The global convergence of the above branch-and-bound optimization algorithm is discussed as follows.

Theorem 1. If the proposed optimization algorithm terminates at k th iteration, then a global optimal solution y^k can be obtained. Otherwise, the proposed optimization algorithm will generate an infinite subsequence $\{y^{k_0}\}$ of $\{y^k\}$ such that its accumulation point is the global optimal solution for the QPP.

Proof. If the proposed optimization algorithm stops after finite k iteration, then from the termination condition of algorithm, it is obvious that $UB_k - LB_k \leq \varepsilon$.

From the bounding process, we know that there does exist a feasible solution y^k such that $UB_k = G_0(y^k)$, so we have $G_0(y^k) - \varepsilon \leq LB_k$.

Let v^* be the global optimal value, then from the structure of the proposed optimization algorithm, we have $LB_k \leq v^*$.

Since y^k is a feasible solution for the QPP, then we follow that $G_0(y^k) \geq v^*$.

From comprehensive analysis of the above results, we can follow that $v^* \geq LB_k \geq G_0(y^k) - \varepsilon \geq v^* - \varepsilon$. Obviously, y^k is a global optimal solution of the QPP.

If the proposed optimization algorithm does not stop after finite k iterations, based on the sufficient condition of the convergence of the branch-and-bound optimization algorithm, we can draw a conclusion that the bounding operation of the proposed optimization algorithm must be consistent and its selection operation may be improved.

From the construction of the above optimization algorithm, the selected bisection method is exhaustive, so that any unfathomed box can be further partitioned by the branching process. By the deriving process of the LRPP, it is easy to follow that $\lim_{k \rightarrow \infty} (UB_k - LB_k) = 0$ must hold. Therefore, the bounding operation must be consistent.

From the proposed bisection method, the selected subbox Y^k of which the lower bound can be immediately attained from further partitioning process. Therefore, the selecting operation of the proposed optimization algorithm that satisfies the bound must be improved.

From comprehensive analysis of the above discussions, we have that the bounding operation of the proposed algorithm must be consistent and its selection operation may be improved. Finally, according to literature,¹⁵⁻¹⁷ we can follow that the proposed optimization must be globally convergent to the optimal solution of the initial problem (QPP).

Numerical experiments

In this section, to verify the feasibility and efficiency of the proposed branch-and-bound optimization algorithm, some random numerical test problems in recent literatures are solved using the proposed optimization algorithm on microcomputer, the algorithm procedure is coded in C++, and the LRPPs are solved by simplex approach. These random numerical test problems and their computational results are demonstrated as follows

$$(QPP) : \begin{cases} \min & G_0(y) = \sum_{j=1}^n \sum_{k=1}^n \beta_{jk}^0 y_j y_k + \sum_{k=1}^n \alpha_k^0 y_k \\ \text{s.t.} & G_i(y) = \sum_{j=1}^n \sum_{k=1}^n \beta_{jk}^i y_j y_k + \sum_{k=1}^n \alpha_k^i y_k \leq b_i, \quad i = 1, \dots, m \\ & y \in Y^0 = \{y \in R^n : l^0 \leq y \leq u^0\} \end{cases}$$

where β_{jk}^0 and α_k^0 are randomly generated in $[-1, 1]$, β_{jk}^i and α_k^i , $i = 1, \dots, m$ are randomly generated in $[-1, 1]$, and b_i is randomly generated in $[-300, -90]$, $l^0 = [0, \dots, 0]^T$, and $u^0 = [10, \dots, 10]^T$.

The numerical experimental results for these test problems are listed in Table 1, where n represents the number of variable and m represents the number of constraints. The numerical results indicate the feasibility of the proposed optimization algorithm.

Table I. Numerical results for test problem.

(m, n)	Our algorithm		
	Iteration	L_{\max}	Time (s)
(5, 5)	582	202	0.983
(10, 5)	625	213	1.458
(20, 5)	492	180	1.445
(30, 5)	419	189	2.985
(40, 5)	545	199	6.544
(50, 5)	595	223	12.566
(60, 5)	545	227	16.589
(70, 5)	606	265	27.743
(80, 5)	519	201	31.603
(90, 5)	548	212	46.646

Concluding remarks

Path planning for the mobile robot is still a challenging problem because the inherent constraints arising from the robot body and the exterior environment cannot be solved. Thus, the major difficulties are that most of existing methods can only get local optimal value instead of global one. To address these difficulties, in this article, we present an efficient optimization algorithm for globally solving the QPP. Based on the convexity of univariate quadratic function, we derive the LRPP of the initial problem, which is embedded into a branch-and-bound framework, so that we can construct a global optimization branch-and-bound algorithm. In addition, a new pruning technique is inserted into the branch-and-bound optimization algorithm to improve the global convergent speed of the proposed optimization algorithm. Finally, the global convergence of the proposed algorithm is proved, and compared with the known algorithms, numerical experiment demonstrates the higher computational efficiency of the proposed algorithm. Therefore, the bounding operation of the proposed algorithm must be consistent and its selection operation may be improved, and it provides an efficient approach to solve the problems of path planning for the mobile robot.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (61703143, 61503123, U1504616), Scientific and Technological Innovation Talents in Xinxiang (CXRC17004), Science and Technology Project of Henan Province (162102210294), and the Key Scientific Research Project of Universities in Henan Province (17B413002).

References

- Cai L and Zhang X. Second order central difference filtering algorithm for SINS/GPS integrated navigation in guided munitions. In: *AICI 2010, Part I, LNAI 6319* (eds Wang FL, Deng H, Gao Y and Lei J), 2010, pp. 193–200. Heidelberg, Berlin: Springer.
- Cai L, Kong F, Chang F, et al. Wavelet multi-resolution analysis aided adaptive Kalman filter for SINS/GPS integrated navigation in guided munitions. In: *AICI 2011, Part II, LNAI 7003*, 2011, pp. 455–462.
- Hoy M, Matveev AS, and Savkin A. Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey. *Robotica* 2015; 33(3): 463–497.
- Chen Y, Han J, and Wu H. Quadratic programming-based approach for autonomous vehicle path planning in space. *Chin J Mech Eng-en* 2012; 25(4): 655–673.
- Lin S, Zhao L, Li F, et al. A nonintrusive load identification method for residential applications based on quadratic programming. *Electr Pow Syst Res* 2016; 133: 241–248.
- Wang ZJ and Li KW. Group decision making with incomplete intuitionistic preference relations based on quadratic programming models. *Comput Ind Eng* 2016; 93: 162–170.
- Wen C, Wang Z, Geng T, et al. Event-based distributed recursive filtering for state-saturated systems with redundant channels. *Inform Fusion* 2018; 39: 96–107.
- Wen C, Cai Y, Liu Y, et al. A reduced-order approach to filtering for systems with linear equality constraints. *Neurocomputing* 2016; 193(12): 219–226.
- Bose S, Gayme DF, Chandy KM, et al. Quadratically constrained quadratic programs on acyclic graphs with application to power flow. *IEEE Trans Control NetSyst* 2015; 2(3): 278–287.
- Sherali HD and Tuncbilek CH. A reformulation-convexification approach for solving nonconvex quadratic programming problems. *J Glob Optim* 1995; 7: 1–31.
- Voorhis TV. A global optimization algorithm using lagrangian underestimates and the interval newton method. *J Glob Optim* 2002; 24: 349–370.
- Burer S and Vandembussche D. A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. *Math Program* 2008; 113: 259–282.
- Zheng XJ, Sun XL, and Li D. Convex relaxations for nonconvex quadratically constrained quadratic programming: matrix cone decomposition and polyhedral approximation. *Math Program* 2011; 129: 301–329.
- Thoai NV. Duality bound method for the general quadratic programming problem with quadratic constraints. *J Optim Theory Appl* 2000; 107(2): 331–354.
- Shen P, Duan Y, and Ma Y. A robust solution approach for nonconvex quadratic programs with additional multiplicative constraints. *Appl Math Comput* 2008; 201: 514–526.
- Jiao H, Liu S, and Lu N. A parametric linear relaxation algorithm for globally solving nonconvex quadratic programming. *Appl Math Comput* 2015; 250: 973–985.

17. Jiao H, Liu S, Yin J, et al. Outcome space range reduction method for global optimization of sum of affine ratios problem. *Open Math* 2016; 14: 736–746.
18. Jiao H and Chen Y. A global optimization algorithm for generalized quadratic programming. *J Appl Math* 2013. DOI: 10.1155/2013/215312.
19. Jiao H, Liu S, and Zhao Y. Effective algorithm for solving the generalized linear multiplicative problem with generalized polynomial constraints. *Appl Math Model* 2015; 39: 7568–7582.
20. Horst R and Tuy H. *Global optimization: deterministic approaches*, 2nd ed. Berlin: Springer Verlag, 1993.