Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 295147, 7 pages
http://dx.doi.org/10.1155/2013/295147

*Research Article*

# Parallel Variable Distribution Algorithm for Constrained Optimization with Nonmonotone Technique

**Congying Han,[1,2] Tingting Feng,[2] Guoping He,[2] and Tiande Guo[1]**

[1] *School of Mathematical Sciences, University of the Chinese Academy of Sciences, No. 19(A), Yuquan Road, Shijingshan District, Beijing 100049, China*
[2] *College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China*

Correspondence should be addressed to Congying Han; hancy@ucas.ac.cn

Received 6 September 2012; Accepted 28 February 2013

Academic Editor: Ching-Jong Liao



A modified parallel variable distribution (PVD) algorithm for solving large-scale constrained optimization problems is developed, which modifies quadratic subproblem $QP_l$ at each iteration instead of the $QP_l^0$ of the SQP-type PVD algorithm proposed by C. A. Sagastizábal and M. V. Solodov in 2002. The algorithm can circumvent the difficulties associated with the possible inconsistency of $QP_l^0$ subproblem of the original SQP method. Moreover, we introduce a nonmonotone technique instead of the penalty function to carry out the line search procedure with more flexibly. Under appropriate conditions, the global convergence of the method is established. In the final part, parallel numerical experiments are implemented on CUDA based on GPU (Graphics Processing unit).

## 1. Introduction

In this paper, we consider the following nonlinear programming problem:

$$\begin{aligned} \min \quad & f(x), \\ \text{s.t.} \quad & c(x) \le 0, \end{aligned} \tag{1}$$

where $f : R^n \to R$ and $c(x) : R^n \to R^m$ are all continuously differentiable. Suppose the feasible set $X$ in (1) is described by a system of inequality constraints:

$$Z = \left\{x \in R^n \mid c_i(x) \le 0,\ I = \{i \mid i = 1, 2, \ldots, m\}\right\}. \tag{2}$$

In this paper, we give a new algorithm based on the method in [1], which partitions the problem variables $x \in R^n$ into $p$ blocks $x_1, \ldots, x_p$, such that

$$x = \left(x_1, \ldots, x_p\right), \quad x_l \in R^{n_l},\ l = 1, \ldots, p,\ \sum_{l=1}^{p} n_l = n. \tag{3}$$

And assume that $X$ has block-separable structure. Specifically,

$$\begin{aligned} & c(x) = \left\{c_1(x_1), c_2(x_2), \ldots, c_p\left(x_p\right)\right\}, \\ & c_l(x_l) : R^{n_l} \longrightarrow R^{m_l}, \quad l \in \{1, 2, \ldots, p\}, \sum_{l=1}^{p} m_l = m. \end{aligned} \tag{4}$$

Parallel variable distribution (PVD) algorithm for solving optimization problems was first proposed by Ferris and Mangasarian [2], in 1994, in which the variables are distributed among $p$ processors. Each processor has the primary responsibility for updating its block of variables while allowing the remaining secondary variables to change in a restricted fashion along some easily computable directions. The distinctive novel feature of this algorithm is the presence of the "forget-me-not" term which allows for a change in "secondary" variables. This makes PVD fundamentally different from the block Jacobi [3] and coordinate descent [4] methods. The forget-me-not approach improves robustness and accelerates convergence of the algorithm and is the key to its success. In 1997, Solodov [5] proposed useful generalizations that consist, for the general unconstrained case, of replacing exact global solution of the subproblems by

a certain natural sufficient descent condition, and, for the convex case, of inexact subproblem solution in the PVD algorithm. Solodov [6] proposed an algorithm applied the PVD approach to problems with general convex constraints directly use projected gradient residual function as the direction and show that the algorithm converges, provided that certain conditions are imposed on the change of secondary variables. In 2002, Sagastizábal and Solodov [1] proposed two new variants of PVD for the constrained case. Without assuming convexity of constraints, but assuming block-separable structure, they showed that PVD subproblems can be solved inexactly by solving their quadratic programming approximations. This extends PVD to nonconvex (separable) feasible sets and provided a constructive practical way of solving the parallel subproblems. For inseparable constraints, but assuming convexity, they developed a PVD method based on suitable approximate projected gradient directions. The approximation criterion was based on a certain error bound result, and it was readily implementable. In 2011, Zheng et al. [7] gave a parallel SSLE algorithm, in which the PVD subproblems are solved inexactly by serial sequential linear equations, for solving large-scale constrained optimization with block-separable structure. Without assuming the convexity of constraints, the algorithm is proved to be globally convergent to a KKT point. Han et al. [8] proposed an asynchronous PVT algorithm for solving large-scale linearly constrained convex minimization problems with the idea of [9] in 2009, which based on the idea that a constrained optimization problem is equivalent to a differentiable unconstrained optimization problem by introducing the Fischer Function. And in particular, different from [9] the linear rate of convergence does not depend on the number of processors.

In this paper, we use [1] as our main reference on SQP-type PVD method for problems (1) and (4). Firstly, we introduce the algorithm in [1] simply. The original problem is distributed into $p$ parallel subproblems which treatment among $p$ parallel processors. The algorithm of [1] may result in that the linear constraints in quadratic programming subproblems are inconsistent or the solution of quadratic programming subproblems is unbounded, so that the algorithm may fail. This drawback has been overcome by many researchers, such as [10]. In [1], exact penalty function is used as merit function to carry out the line-search. For the penalty function, as pointed out by Fletcher and Leyffer [11], the biggest drawback is that the penalty parameter estimates could be problematic to obtain. To overcome this drawback, we use a nonmonotone technique to carry out the line search instead of penalty function.

Recent research [12–14] indicates that the monotone line search technique may have some drawbacks. In particular, enforcing monotonicity may considerably reduce the rate of convergence when the iteration is trapped near a narrow curved valley, which can result in very short steps or zigzagging. Therefore, it might be advantageous to allow the iterative sequence to occasionally generate points with nonmonotone objective values. Grippo et al. [12] generalized the Armijo rule and proposed a nonmonotone line search technique for Newton's method which permits increase in function value, while retaining global convergence of the minimization algorithm. In [15], Sun et al. give several nonmonotone line search techniques, such as nonmonotone Armijo rule, nonmonotone Wolfe rule, nonmonotone F-rule, and so on. Several numerical tests show that the nonmonotone line search technique for unconstrained optimization and constrained optimization is efficient and competitive [12–14, 16]. Recently, [17] gives a method to overcome the drawback of zigzagging with nonmonotone line search technique to determine the step length, which makes the algorithm more flexible.

In this paper, we combine [1] with the ideas of [10, 17] and propose an infeasible SQP-type Parallel Variable Distribution algorithm for constrained optimization with nonmonotone technique.

The paper is organized as follows. The algorithm is presented in Section 2. In Section 3, under mild assumptions, some global convergence results are proved. The numerical results are shown in Section 4. And conclusions are given in the last section.

## 2. A Modified SQP-Type PVD Method with Nonmonotone Technique

To describe our algorithm, we first give the modified quadratic subproblems of SQP method. Given $x^k \in R^n$, we distribute it into $p$ blocks, as the iterate $x_l^k \in R^{n_l}$, $l = 1 \ldots, p$, $H_l \in R^{n_l \times n_l}$, $l = 1, 2, \ldots, p$ denote an approximate Hessian.

We use $z_l \in R$ to perturb the subproblem of $QP_l^0$ of [1] and give a new subproblem $QP_l$ instead of it. Consider

$$QP_l\left(x_l^k, H_l^k\right) = \begin{cases} \min\limits_{(d_l, z_l) \in R^{n_l+1}} & z_l + \dfrac{1}{2} d_l^T H_l^k d_l \\ \text{s.t.} & \left(g_l^k\right)^T d_l \le z_l, \\ & c_i\left(x_l^k\right) + \nabla c_i\left(x_l^k\right)^T d_l \le z_l, \\ & \quad i \in I_l = \{1, \ldots, m_l\}. \end{cases} \tag{5}$$

For convenience, given $x$, the $l$th block of $\nabla f(x)$ will be denoted by

$$g_l = \left(I_{n_l \times n_l} \quad 0_{\overline{n}_l \times \overline{n}_l}\right) \nabla f(x). \tag{6}$$

In (5), $g_l^k$ is $g_l$ in step $k$. Note that (5) is always feasible for $d_l = 0$, $z_l = \max_{i \in I_l}\{c_i(x_l^k), 0\}$. To test whether constraints are satisfied or not, we denote the violation function $h(x)$ as follows:

$$h(x) = \left\| c(x)^+ \right\|, \tag{7}$$

where $c_i(x)^+ = \max\{c_i(x), 0\}$, $i \in I$, $\|\cdot\|$ denotes the Euclidean norm on $R^m$. It is easy to see that $h(x) = 0$ if and only if $x$ is a feasible point ($h(x) > 0$ if and only if $x$ is infeasible).

We describe our modified PVD algorithm as follows.

*Algorithm A.* Consider the following.

*Step 0.* Start with any $x_0 \in R^n$. Choose parameters $u \in (0, 1/2)$, $\beta \in (1/2, 1)$, $\gamma \in (0, 1)$, positive integer $M > 0$, and positive definite matrices $H_l^0 \in R^{n_l \times n_l}$, $l \in \{1, 2, \ldots, p\}$. Set $k = 0, m(k) = 0$.

Having $x_k$, check a stopping criterion. If it is not satisfied, compute $x_{k+1}$ as follows.

*Step 1.* Parallelization. For each processor $l \in \{1, 2, \ldots, p\}$, solve subproblem (5) to get $(d_l^k, z_l^k)$. If $d_l^k = 0, l = 1, \ldots, p$ stop, otherwise return to Step 2.

*Step 2.* Synchronization. Set

$$d_k = \left(d_1^k, \ldots, d_p^k\right), \qquad g_k^T = \left(\left(g_1^k\right)^T, \ldots, \left(g_p^k\right)^T\right),$$
$$H_k = \begin{pmatrix} H_1^k & & \\ & \ddots & \\ & & H_p^k \end{pmatrix}. \tag{8}$$

If $g_k^T d_k \le -(1/2) d_k^T H_k d_k$, return to Step 3; otherwise, set $\lambda_k = 1$ and return to Step 4.

*Step 3.* Choose $\lambda_k$ which is the largest one in the sequence $\{1, \gamma, \gamma^2, \ldots\}$ satisfying:

$$f\left(x_k + \lambda d_k\right) \le \max_{0 \le j \le m(k)} \left[f\left(x_{k-j}\right)\right] - \lambda \mu d_k^T H_k d_k. \tag{9}$$

*Step 4.* Set $x_{k+1} = x_k + \lambda_k d_k$. Let $\overline{h} = \max_{0 \le j \le m(k)}[h(x_{k-j})]$. If $h(x_{k+1}) \le \beta \overline{h}$, then set $m(k+1) = \min\{m(k) + 1, M\}$, update $H_l^k$ to $H_l^{k+1}$, $l = 1, \ldots, p$, set $k = k + 1$, and go to Step 1. Otherwise, let $k = k + 1$, call Restoration Algorithm of [17] to obtain $x_k^r$, let $x_k = x_k^r$, $m(k) = m(i)$, and go to Step 1.

*Remark 1.* In Step 3 of Algorithm A, the nonmonotone parameter $m(k)$ satisfies

$$m(0) = 0, \qquad 0 \le m(k) \le \min\{m(k-1) + 1, M\}, \quad k \ge 1. \tag{10}$$

For the convenience, we denote $f(x_{l(k)}) = \max_{0 \le j \le m(k)}[f(x_{k-j})]$, where $k - m(k) \le l(k) \le k$.

In a restoration algorithm, we aim to decrease the value of $h(x)$ more precisely, we will use a trust region type method to obtain $h(x_k) \to 0, k \to \infty$ by the help of the nonmonotone technique. Let

$$M_k^i(d) = h\left(x_k^i\right) - h\left(x_k^i + d\right),$$
$$\Psi_k^i(d) = h\left(x_k^i\right) - \left\| \left(c\left(x_k^i\right) + \nabla c\left(x_k^i\right)^T d\right)^+ \right\|, \tag{11}$$

and $r_k^i = M_k^i(d)/\Psi_k^i(d)$.

Algorithm B is similar to the restoration phase given by Su and Yu [17], we describe the Restoration Algorithm as follows.

*Algorithm B.* Consider the following.

*Step 0.* Assume $x_k^0 = x_k$, $\Delta^0 > 0$, $i = 0$, $\eta \in (0, 1)$, $m(i) = 0$.

*Step 1.* If $h(x_k^i) \le \beta \overline{h}$, then $x_k^r = x_k^i$ stop.

*Step 2.* Compute

$$\begin{aligned} \max \quad & \Psi_k^i(d) \\ \text{s.t.} \quad & \|d\| \le \Delta^i \end{aligned} \tag{12}$$

to get $d_k^i$. Calculate $r_k^i$.

*Step 3.* If $r_k^i \le \eta$, then let $x_k^{i+1} = x_k^i$, $\Delta^{i+1} = \Delta^i/2$, $i = i + 1$, $m(i) = \min\{m(i-1) + 1, M\}$ and go to Step 2.

*Step 4.* If $r_k^i > \eta$, then let $x_k^{i+1} = x_k^i + d_k^i$, $\Delta^{i+1} = 2\Delta^i$, $i = i + 1$, $m(i) = \min\{m(i-1) + 1, M\}$ and go to Step 1.

## 3. The Convergence Properties

To prove the global convergence of Algorithm A, we make the following assumptions.

*Assumptions*

(A1) The iterate $\{x_k\}$ remains in compacted subset $S \subset R^n$.

(A2) The objective function $f$ and the constraint functions $c_j$ $(j \in I)$ are twice continuously differentiable on $R^n$.

(A3) For all $x_l \in R^{n_l}$, the set $\{\nabla c_i(x_l) : i \in I_l(x_l)\}$ is linearly independent, where $I_l(x_l) = \{i \in I_l : c_i(x_l) = \Phi_l(x_l), \Phi_l(x_l) = \max_{i \in I_l}\{c_i(x_l), 0\}\}$.

(A4) There exist two constants $0 < a \le b$ such that $a\|d_l\|^2 \le d_l^T H_l^k d_l \le b\|d_l\|^2$, $l = 1, \ldots, p$ for all iteration $k$ and $d_l \in R^{n_l}$.

(A5) The solution of problem (12) satisfies

$$\Psi_k^i(d) = h\left(x_k^i\right) - \left\| \left(c\left(x_k^i\right) + \nabla c\left(x_k^i\right)^T d\right)^+ \right\| \ge \beta_2 \Delta^i \min\left\{h\left(x_k^i\right), \Delta^i\right\}, \tag{13}$$

where $\beta_2$ is a constant.

*Remark 2.* Assumptions (A1) and (A2) are the standard assumptions. (A3) is the LICQ constraint qualification. (A4) plays an important role in obtaining the convergence results. (A5) is the sufficient reduction condition which guarantees the global convergence in a trust region method. Under the assumptions, $f$ is bounded below and the gradient function $g_l$, $l = 1, \ldots, p$ is uniformly continuous in $S_l$, where $S = (S_1, \ldots, S_p)$.

*Remark 3.* We can use quasi-Newton BFGS methods to update $H_l^k$, different from [18], we can use a small modification of $y_l^k$ to make $H_l^k$ reserve positive definite according to the formula (33) of [17].

Similar to Lemma 1 in [19], we can get the following conclusions.

**Lemma 4.** *Suppose (A1)–(A4) hold, $H_l^k$, $l = 1, \ldots, p$ is a symmetric positive definite, then $d_l^k$ is well defined and $(d_l^k, z_l^k)$ is the unique KKT point of* (5)*. Furthermore, $d_l^k$ is bounded over compact subsets of $S_l \times P_l$, where $P_l$ is the set of symmetric positive definite $n_l \times n_l$ matrices.*

*Proof.* First note that the feasible set for (5) is nonempty, since $(d_l, z_l) = (0, \max_{i\in I_l}\{c_i(x_l), 0\})$ is always feasible. It is clear that, if $(d_l^k, z_l^k)$ is a solution to (5) if an only if $d_l^k$ solves the unconstrained problem

$$\min_{d_l\in R^{n_l}} \frac{1}{2} d_l^T H_l^k d_l + \max\left\{\left(g_l^k\right)^T d_l, \max_{i\in I_l}\left\{c_i\left(x_l^k\right) + \nabla c_i\left(x_l^k\right)^T d_l\right\}\right\}, \tag{14}$$

$$z_l = \max\left\{\left(g_l^k\right)^T d_l, \max_{i\in I_l}\left\{c_i\left(x_l^k\right) + \nabla c_i\left(x_l^k\right)^T d_l\right\}\right\}. \tag{15}$$

Since the function being minimized in (14) is strictly convex and radially unbounded, it follows that $d_l^k$ is well defined and unique as a global minimizer for the convex problem (5) and thus unique as a KKT point for that problem. So we have

$$H_l^k d_l^k + v g_l^k + \nabla c_l\left(x_l^k\right)^T \mu_l^k = 0, \tag{16}$$

$$1 - v - \sum_{i=1}^{m_l} u_i = 0, \tag{17}$$

due to (17) and $u_i \geq 0, v \geq 0$, we have $u_i$, $i \in I_l$ is bounded. Since $f, c_i$ is twice continuously differentiable and assumption (A1) holds, for all $x_l \in S_l$, $\| -(v g_l + \nabla c_l(x_l^k)^T \mu_l^k\|$ is bounded, set the maximum is $M$. For (16), $\|H_l^k d_l^k\| \leq M$ and combining with Assumption (A4), we have $a\|d_l^k\|^2 \leq d_l^{kT} H_l^k d_l^k \leq \|d_l^k\|\|H_l^k d_l^k\| \leq M\|d_l^k\|$, so that $\|d_l^k\| \leq M/a$; therefore, $d_l^k$ is bounded on $S_l \times P_l$. □

**Lemma 5.** *Suppose that $x_l^k \in R^{n_l}$, $l = 1, \ldots, p$, $H_l^k$ is positive definite matrix, and $(d_l^k, z_l^k)$ is the solution of $QP_l$. Then we have the results (B1) and (B2).*

(B1) *The following inequality holds:*

$$z_l^k \leq \Phi_l\left(x_l^k\right) - \frac{1}{2}\left(d_l^k\right)^T H_l^k d_l^k, \quad l = 1, \ldots, p, \tag{18}$$

*where $\Phi_l(x_l) = \max_{i\in I_l}\{c_i(x_l), 0\}$.*

(B2) *If $d_l^k = 0$ then $z_l^k = 0$, $l = 1, \ldots, p$, and $x_k = (x_1^k, \ldots, x_p^k)$ is a KKT point of problem* (1).

*Proof.* (B1) Since $\overline{d}_l = 0, \overline{z}_l = \max_{i\in I_l}\{c_i(x_l), 0\}$ is a feasible point of $QP_l$, from the optimality of $(d_l^k, z_l^k)$, we have

$$z_l^k + \frac{1}{2}\left(d_l^k\right)^T H_l^k d_l^k \leq \overline{z}_l + 0 = \max_{i\in I_l}\left\{c_i\left(x_l^k\right), 0\right\}. \tag{19}$$

Together with the defininationof $\Phi_l(x_l^k)$, implies that (18) holds.

(B2) Since $(d_l^k, z_l^k)$ is the solution of (5), there exists $v \in R$ and $\mu \in R^{m_l}$ such that the KKT condition of (5), while $d_l^k = 0$, we have

$$v g_l^k + \nabla c_l\left(x_l^k\right)^T \mu_l^k = 0, \tag{20}$$

$$1 - v - \sum_{i=1}^{m_l} u_i^k = 0, \tag{21}$$

$$-z_l^k \leq 0, \quad c_i\left(x_l^k\right) - z_l^k \leq 0, \quad i \in I_l, \tag{22}$$

$$v z_l^k = 0, \quad v \geq 0, \tag{23}$$

$$u_i^k\left[c_i\left(x_l^k\right) - z_l^k\right] = 0, \quad u_i^k \geq 0, \quad i \in I_l. \tag{24}$$

By the definition of $\Phi_l(x_l^k)$ and (22), $\Phi_l(x_l^k) \leq z_l^k$. Then, from (18) and $d_l^k = 0$, $\Phi_l(x_l^k) \geq z_l^k$. From (24), it follows that

$$u_i^k = 0, \quad \forall i \notin I_l^0 = \left\{i \in I_l : c_i\left(x_l^k\right) = \Phi_l\left(x_l^k\right)\right\}. \tag{25}$$

Hence, it follows from Assumption (A3) Together with (20) and (21) that $v > 0$. For (23) we have $z_l^k = 0$.

From (22) and (24), we have

$$u_i^k c_i\left(x_l^k\right) = 0, \quad u_i^k \geq 0, \quad i \in I_l, \qquad c_i\left(x_l^k\right) \leq 0, \quad i \in I_l, \tag{26}$$

due to (20) and set $\rho_i^k = u_i^k / v$, we have

$$g_l^k + \nabla c_l\left(x_l^k\right)^T \rho_l^k = 0. \tag{27}$$

Taking into account separability of constraints $c(x)$, and let $I = \cup_{l=1}^p I_l$, then

$$\rho_i^k c_i\left(x_k\right) = 0, \quad \rho_i^k \geq 0, \quad i \in I, \qquad c_i\left(x_k\right) \leq 0, \quad i \in I, \qquad g_k + \nabla c_i\left(x_k\right)^T \rho_k = 0; \tag{28}$$

that is, $x_k$ is a KKT point of (1). □

**Lemma 6** (see [17, Lemma 3]). *In Step 3, the line search procedure is well defined.*

**Lemma 7** (see [17, Lemma 4]). *Under Assumption (A5), the Restoration Algorithm (Algorithm B) terminates finitely.*

From Lemmas 6 and 7, we know that the Algorithm A and Algorithm B are well implemented.

The following Lemma implies that every cluster point of $\{x_k\}$ which generated by Algorithm A is a feasible point of problem (1).

**Lemma 8.** *Let $\{x_k\}$ be an infinite sequence generated by Algorithm A, then $h(x_k) \to 0$ $(k \to \infty)$.*

**Theorem 9.** *Suppose $\{x_k\}$ is an infinite sequence generated by Algorithm A, $d_l^k$, $l = 1, \ldots, p$ is the solution of (5), $d_k = (d_1^k, \ldots, d_p^k)$, then $\lim_{k\to\infty}\|d_l^k\| = 0$, $l = 1, \ldots, p$.*

*Proof.* By Assumption (A1), there exists a point $x^*$ such that $x_k \to x^*$ for $k \in K$, where $K$ is an infinite index set. By Algorithm A and Lemma 8, we consider the following two possible cases. □

*Case I.* $K_0 = \{k \in K \mid g_k^T d_k \le -(1/2)d_k^T H_k d_k\}$ is an infinite index set. In this case, we have

$$\begin{aligned} f\left(x_k + \lambda_k d_k\right) &\le f\left(x_{l(k)}\right) - \lambda_k \mu d_k^T H_k d_k \\ &= f\left(x_{l(k)}\right) - \lambda_k \mu \sum_{l=1}^{p} d_l^{kT} H_l^k d_l^k. \end{aligned} \tag{29}$$

Since $m(k+1) \le m(k) + 1$, we obtain

$$\begin{aligned} f\left(x_{l(k+1)}\right) &= \max_{0\le j\le m(k+1)} \left[f\left(x_{k+1-j}\right)\right] \\ &\le \max_{0\le j\le m(k)+1} \left[f\left(x_{k+1-j}\right)\right] \\ &= \max\left\{f\left(x_{l(k)}\right), f\left(x_{k+1}\right)\right\} \\ &= f\left(x_{l(k)}\right). \end{aligned} \tag{30}$$

Hence for $m(k) \le M$, it holds

$$\begin{aligned} f\left(x_{l(k)}\right) &\le f\left(x_{l(l(k)-1)}\right) - \lambda_{l(k)-1}\mu d_{l(k)-1}^T H_{l(k)-1} d_{l(k)-1} \\ &= f\left(x_{l(l(k)-1)}\right) - \lambda_{l(k)-1}\mu \sum_{l=1}^{p} \left(d_l^{l(k)-1}\right)^T H_l^{l(k)-1} d_l^{l(k)-1}. \end{aligned} \tag{31}$$

Since $f$ is bounded below, $\{f(x_{l(k)})\}$ converges. Due to (31), we can obtain

$$\lim_{k\to\infty} \lambda_{l(k)-1}\mu \sum_{l=1}^{p} \left(d_l^{l(k)-1}\right)^T H_l^{l(k)-1} d_l^{l(k)-1} = 0. \tag{32}$$

By Lemma 6, there exists $\overline{\lambda} > 0$ such that $\lambda_{l(k)-1} \ge \overline{\lambda}$. By Assumption (A4), we obtain

$$\lim_{k\to\infty} \left\|d_l^{l(k)-1}\right\| = 0, \quad l = 1, \ldots, p. \tag{33}$$

From the uniform continuity of $f(x)$, this implies that

$$\lim_{k\to\infty} f\left(x_{l(k)-1}\right) = \lim_{k\to\infty} f\left(x_{l(k)}\right). \tag{34}$$

Let $\bar{l}(k) = l(k + M + 2)$, for any given $j \ge 1$, we can have

$$\lim_{k\to\infty} \left\|d_l^{\bar{l}(k)-j}\right\| = 0, \quad l = 1, \ldots, p, \tag{35}$$

$$\lim_{k\to\infty} f\left(x_{\bar{l}(k)-j}\right) = \lim_{k\to\infty} f\left(x_{l(k)}\right). \tag{36}$$

If $j = 1$, since $\{\bar{l}(k)\} \subset \{l(k)\}$, (35) and (36) follow from (33) and (34). For a given number $j > 1$, we have

$$\begin{aligned} &f\left(x_{\bar{l}(k)-j}\right) \\ &\quad \le f\left(x_{l(\bar{l}(k)-j-1)}\right) - \lambda_{\bar{l}(k)-j-1}\mu d_{\bar{l}(k)-j-1}^T H_{\bar{l}(k)-j-1} d_{\bar{l}(k)-j-1}. \end{aligned} \tag{37}$$

Using the same arguments above, we obtain

$$\begin{aligned} &\lim_{k\to\infty} \left\|d_l^{\bar{l}(k)-j-1}\right\| = 0, \quad l = 1, \ldots, p, \\ &\lim_{k\to\infty} f\left(x_{\bar{l}(k)-j-1}\right) = \lim_{k\to\infty} f\left(x_{l(k)}\right). \end{aligned} \tag{38}$$

Therefore (35) and (36) hold for any given $j \ge 1$.

Now for any $k$, $x_{k+1} = x_{\bar{l}(k)} - \sum_{j=1}^{\bar{l}(k)-k-1} \lambda_{\bar{l}(k)-j} d_{\bar{l}(k)-j}$, from Remark 1, we obtain $\bar{l}(k) = l(k + M + 2) < k + M + 2$, that is $\bar{l}(k) - k - 1 \le M + 1$. Since $d_{\bar{l}(k)-j} = \{d_1^{\bar{l}(k)-j}, \ldots, d_p^{\bar{l}(k)-j}\}$ together with (35), we can obtain $\lim_{k\to\infty}\|d_{\bar{l}(k)-j}\| = 0$, $1 \le j \le \bar{l}(k) - k - 1$. Therefore,

$$\lim_{k\to\infty} \left\|x_{k+1} - x_{\bar{l}(k)}\right\| = 0. \tag{39}$$

Since $\{f(x_{l(k)})\}$ admits a limit, by the uniform continuity of $f$ on $S$, it holds

$$\lim_{k\to\infty} f\left(x_k\right) = \lim_{k\to\infty} f\left(x_{\bar{l}(k)-1}\right) = \lim_{k\to\infty} f\left(x_{l(k)}\right). \tag{40}$$

Then by the relation (29), we have

$$\lim_{k\to\infty} \sum_{l=1}^{p} \lambda_k \mu d_l^{kT} H_l^k d_l^k = 0. \tag{41}$$

Using the same arguments for deriving (33), we obtain that

$$\lim_{k\to\infty} \left\|d_l^k\right\| = 0, \quad l = 1, \ldots, p. \tag{42}$$

*Case II.* $K_0$ is a finite index set, which implies $K_1 = \{k \in K \mid g_k^T d_k > -(1/2)d_k^T H_k d_k\}$ is an infinite index set.

If $\lim_{k\to\infty}\|d_l^k\| = 0$, $l = 1, \ldots, p$ does not hold, then there exists a positive number $\varepsilon > 0$ and an infinite index set $K_2$, such that $\|d_l^k\| > \varepsilon$, for all $k \in K_2 \subset K_1$. Since $(d_l^k, z_l^k)$ is the solution of $QP_l$. By the KKT condition of (5), we have

$$H_l^k d_l^k + v g_l^k + \nabla c_l\left(x_l^k\right)^T \mu_l^k = 0, \tag{43}$$

$$1 - v - \sum_{i=1}^{m_l} u_i = 0, \tag{44}$$

$$g_l^{kT} d_l^k - z_l^k \le 0, \tag{45}$$

$$u_i \left[c_i\left(x_l^k\right) + \nabla c_i\left(x_l^k\right)^T d_l^k - z_l^k\right] = 0, \quad u_i \ge 0,\ i \in \mathrm{I}_l. \tag{46}$$

Since $v \ge 0$, $u_i \ge 0$ and (44), we obtain $0 \le u_i \le 1$, therefore there exists $\overline{M}_l$ satisfied $\|u_l^k\| \le \overline{M}_l$.

By Lemma 8, $h(x_k) \to 0$ implies $h_l(x_l^k) \to 0$. Hence there exists $k_0$, such that for $k > k_0$, $k \in K_2$, it holds

$$h_l\left(x_l^k\right) \le \frac{a\varepsilon^2}{2\overline{M}_l} \le \frac{a\left\|d_l^k\right\|^2}{2\overline{M}_l} \le \frac{d_l^{kT} H_l^k d_l^k}{2\overline{M}_l}, \quad l = 1, \ldots, p. \tag{47}$$

Consequently, from (43), we deduce

$$\begin{aligned} \nu g_l^{kT} d_l^k &= -d_l^{kT} H_l^k d_l^k - \mu_l^{kT} \nabla c_l\left(x_l^k\right) d_l^k \\ &= -d_l^{kT} H_l^k d_l^k + \sum_{i=1}^{m_l} u_i \left(c_i\left(x_l^k\right) - z_l^k\right) \\ &\le -d_l^{kT} H_l^k d_l^k + \overline{M}_l h_l\left(x_l^k\right) - \sum_{i=1}^{m_l} u_i z_l^k \\ &\le -d_l^{kT} H_l^k d_l^k + \overline{M}_l h_l\left(x_l^k\right) - \sum_{i=1}^{m_l} u_i g_l^{kT} d_l^k. \end{aligned} \tag{48}$$

Together with (44), (47), and transpose

$$\begin{aligned} g_l^{kT} d_l^k &= \left(\sum_{i=1}^{m_l} u_i + \nu\right) g_l^{kT} d_l^k \\ &\le -d_l^{kT} H_l^k d_l^k + \overline{M}_l h_l\left(x_l^k\right) \\ &\le -d_1^{kT} H_l^k d_l^k + \frac{1}{2} d_l^{kT} H_l^k d_l^k \\ &= -\frac{1}{2} d_l^{kT} H_l^k d_l^k. \end{aligned} \tag{49}$$

Taking into account separability of constraints $c(x)$, then

$$\begin{aligned} g_k^T d_k &= \sum_{l=1}^{p} g_l^{kT} d_l^k \\ &\le -\sum_{l=1}^{p} \frac{1}{2} d_l^{kT} H_l^k d_l^k \\ &= -\frac{1}{2} d_k^T H_k d_k, \end{aligned} \tag{50}$$

which contradicts the definition of $K_1$. The proof is complete.

**Theorem 10.** *Suppose $\{x_k\}$ is an infinite sequence generated by Algorithm A and the assumptions in Theorem 9 hold. Then every cluster point of $\{x_k\}$ is a KKT point of problem* (1).

*Proof.* By Assumption (A2), there exists $x^*$, such that $x_k \to x^*$, $k \in K$. From Lemma 8, we have $h(x_k) \to 0$, $k \in K$, which means that $x^*$ is a feasible point. From Theorem 9, we have $\lim_{k\to\infty} \|d_l^k\| = 0$, $l = 1, \ldots, p$. By Lemma 5, $d_l^k \to 0$ implies that $z_l^k \to 0$, $l = 1, \ldots, p$, that is, $d_l^* = 0$, $l = 1, \ldots, p$, $z_l^* = 0$, $l = 1, \ldots, p$.

Compare the KKT condition of (5) with the KKT condition of problem (1), and let $\rho^* = (\rho_1^*, \ldots \rho_p^*)$, $\rho_l^* = (u_i^*/\nu,\ i \in I_l)$, $I = \cup_{l=1}^{p} I_l$.

Table 1: Results for Algorithm A.

| Problem 1 $n, \nu$ $\nu = 3$ | $P$ | Algorithm A Iteration number | Iteration number of $QP_l$ subproblem | Running times |
|---|---|---|---|---|
| $n = 6$ | 2 | 5 | 52 | 0.5 |
| $n = 12$ | 4 | 10 | 66 | 0.5 |
| $n = 24$ | 8 | 15 | 78 | 1 |
| $n = 48$ | 16 | 9 | 46 | 2 |
| $n = 96$ | 32 | 6 | 50 | 9 |
| $n = 192$ | 64 | 7 | 24 | 13 |

Taking into account separability of constraints $c(x)$ and $x$, we conclude

$$\begin{gathered} g^* + \rho^* \nabla c^* = 0, \qquad c\left(x^*\right) \le 0, \qquad \rho_i^* \ge 0, \\ \rho_i^{*T} c_i\left(x^*\right) = 0, \quad i \in I. \end{gathered} \tag{51}$$

Therefore $x^*$ is a KKT point of problem (1). □

## 4. Numerical Results

To get some insight into computational properties of our approach in Section 2, we considered the same test problems taken from [1]. Choose Problem 1 of [1] as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \sum_{t=1}^{\nu} x_{(i-1)\nu+t} x_{(j-1)\nu+t} \\ \text{s.t.} \quad & \sum_{t=1}^{\nu} x_{(i-1)\nu+t}^2 - 1 = 0, \quad i = 1, \ldots, p. \end{aligned} \tag{52}$$

In our test, we only verify our convergence theory without comparing with serial algorithms. In the future, we will propose the convergent rate of the parallel algorithm.

Not having access to a parallel computer, we have carried out on Graphics Processing Unit (GPU) to solve them and implement the algorithm on Compute Unified Device Architecture (CUDA) [20]. However, the input and output are implemented by CPU. And all the subproblems are solved on blocks which are constructed into many threads. If there are $p$ blocks in GPU, which is equivalent to $p$ processors in a parallel computer. Many threads of the block can achieve a large number of vector calculations in current block.

We have implemented Algorithm A in C language. All codes were run under Visual Studio 2008 and Cuda 4.0 on the DELL Precision T7400. In Table 1, we report the number of iterations and the running times for Algorithms A on Problem (52). The results in Table 1 confirm that the proposed approach certainly makes sense. We solve the subproblem (5) which is a positive semidefinite quadratic programs by the modified interior-point methods. And the Restoration Algorithm of [17] is solved by Trust-region Approach combined with local pattern search methods. All algorithms are coded by C language without using any function from optimization Toolbox; hence the results are not the best.

In Table 1, $n$ denotes the dimensions of test problem and $\nu$ denotes the variable number of one block. There is less iteration number of $QP_l$ subproblem with more numbers of blocks for high-dimensional problem, which shows the algorithm is reliable.

## 5. Conclusion

The PVD algorithm which was proposed in 1994 is used to solve unconstrained optimization problems or has a special case of convex block-separable constraints. In 1997, M. V. Solodov proposed useful generalizations that consist, for the general unconstrained case, of replacing exact global solution of the subproblems by a certain natural sufficient descent condition, and, for the convex case, of inexact subproblem solution in the PVD algorithm. M. V. Solodov proposed a PVD approach in 1998 which applied to problems with general convex constraints directly and show that the algorithm converges. In 2002, C. A. Sagastizábal et al. propose two variants of PVD for the constrained case: without assuming convexity of constraints, but assuming block-separable structure and for inseparable constraints, but assuming convexity. In this paper, we propose the modified algorithm of [1] used to the general constraints but with block-separable structure.

In a word, the algorithm above is a special structure of the objective function or constraints with a special structure. In the further, we will study the parallel algorithm with general inseparable constraints.

## Acknowledgments

This paper was supported by National Natural Science Foundation of China (11101420, 71271204) and other foundations (2010BSE06047). The authors would like to thank Dr. Jiang Zhipeng for valuable work on numerical experiments.

## References

[1] C. A. Sagastizábal and M. V. Solodov, "Parallel variable distribution for constrained optimization," *Computational Optimization and Applications*, vol. 22, no. 1, pp. 111–131, 2002.

[2] M. C. Ferris and O. L. Mangasarian, "Parallel constraint distribution," *SIAM Journal on Optimization*, vol. 1, no. 1, pp. 487–500, 1994.

[3] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*, Prentice Hall, Englewood Cliffs, NJ, USA, 1989.

[4] P. Tseng, "Dual coordinate ascent methods for non-strictly convex minimization," *Mathematical Programming*, vol. 59, no. 2, pp. 231–247, 1993.

[5] M. V. Solodov, "New inexact parallel variable distribution algorithms," *Computational Optimization and Applications*, vol. 7, no. 2, pp. 165–182, 1997.

[6] M. V. Solodov, "On the convergence of constrained parallel variable distribution algorithms," *SIAM Journal on Optimization*, vol. 8, no. 1, pp. 187–196, 1998.

[7] F. Zheng, C. Han, and Y. Wang, "Parallel SSLE algorithm for large scale constrained optimization," *Applied Mathematics and Computation*, vol. 217, no. 12, pp. 5377–5384, 2011.

[8] C. Han, Y. Wang, and G. He, "On the convergence of asynchronous parallel algorithm for large-scale linearly constrained minimization problem," *Applied Mathematics and Computation*, vol. 211, no. 2, pp. 434–441, 2009.

[9] M. Fukushima, "Parallel variable transformation in unconstrained optimization," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 658–672, 1998.

[10] J. Mo, K. Zhang, and Z. Wei, "A variant of SQP method for inequality constrained optimization and its global convergence," *Journal of Computational and Applied Mathematics*, vol. 197, no. 1, pp. 270–281, 2006.

[11] R. Fletcher and S. Leyffer, "Nonlinear programming without a penalty function," *Mathematical Programming*, vol. 91, pp. 239–269, 2002.

[12] L. Grippo, F. Lampariello, and S. Lucidi, "A nonmonotone line search technique for Newton's method," *SIAM Journal on Numerical Analysis*, vol. 23, no. 4, pp. 707–716, 1986.

[13] L. Grippo, F. Lampariello, and S. Lucidi, "A truncated Newton method with nonmonotone line search for unconstrained optimization," *Journal of Optimization Theory and Applications*, vol. 60, no. 3, pp. 401–419, 1989.

[14] G. Liu, J. Han, and D. Sun, "Global convergence of the BFGS algorithm with nonmonotone linesearch," *Optimization*, vol. 34, no. 2, pp. 147–159, 1995.

[15] W. Sun, J. Han, and J. Sun, "Global convergence of nonmonotone descent methods for unconstrained optimization problems," *Journal of Computational and Applied Mathematics*, vol. 146, no. 1, pp. 89–98, 2002.

[16] P. L. Toint, "An assessment of nonmonotone linesearch techniques for unconstrained optimization," *SIAM Journal on Scientific Computing*, vol. 17, no. 3, pp. 725–739, 1996.

[17] K. Su and Z. Yu, "A modified SQP method with nonmonotone technique and its global convergence," *Computers & Mathematics with Applications*, vol. 57, no. 2, pp. 240–247, 2009.

[18] Y.-X. Yuan and W. Sun, *Optimization Theory and Methods*, Springer, New York, NY, USA, 2006.

[19] C. T. Lawrence and A. L. Tits, "A computationally efficient feasible sequential quadratic programming algorithm," *SIAM Journal on Optimization*, vol. 11, no. 4, pp. 1092–1118, 2001.

[20] I. Buck, Parallel Programming with CUDA, 2008, http://gpgpu.org/sc2008.