A Parallel Descent Algorithm for Convex Programming[1]

Masao FUKUSHIMA[2], Mounir HADDOU[3], Van Hien NGUYEN[4],
Jean-Jacques STRODIOT[4] and Takanobu SUGIMOTO[5]

August 1993

[1] This work was supported in part by the Scientific Research Grant-in-Aid from the Ministry of Education, Science and Culture, Japan (No. 02680025) and by a grant from the Belgian Fonds National de la Recherche Scientifique (FNRS: B 8/5-JS-9.549).

[2] Graduate School of Information Science,
Nara Institute of Science and Technology,
Ikoma, Nara 630-01, Japan

[3] Department of Applied Mathematics,
Université Blaise Pascal,
53177 Aubière Cédex, France

[4] Department of Mathematics,
Facultés Universitaires ND de la Paix,
B-5000 Namur, Belgium

[5] Ministry of International Trade and Industry,
Tokyo 100, Japan

Abstract In this paper, we propose a parallel decomposition algorithm for solving a class of convex optimization problems, which is broad enough to contain ordinary convex programming problems with a strongly convex objective function. The algorithm is a variant of the trust region method applied to the Fenchel dual of the given problem. We prove global convergence of the algorithm and report some computational experience with the proposed algorithm on a parallel computer.

Key Words Decomposition, parallel algorithm, descent method, convex programming, trust region method.

Abbreviated Title Parallel algorithm for convex programs

1. Introduction

Recently much effort has been made to develop algorithms for solving various optimization problems on parallel computers [2, 16]. In this paper, we propose an algorithm, which is particularly suited for parallel computation, for solving the convex programming problem

$$\text{minimize} \quad f_0(x) + \sum_{i=1}^{m} f_i(x) \quad \text{over} \quad x \in R^n, \tag{1.1}$$

where $f_0 : R^n \to R$ is differentiable and strongly convex with modulus α (see, for example, Rockafellar [14]) and $f_i : R^n \to R \cup \{+\infty\}$, $i = 1, \cdots, m$, are closed proper convex functions. (For the basic definitions in convex analysis, the reader is referred to Rockafellar's book [13].) Associated with problem (1.1) is the following Fenchel dual problem:

$$\text{minimize} \quad f_0^*(y_1 + \cdots + y_m) + \sum_{i=1}^{m} f_i^*(-y_i) \quad \text{over} \quad (y_1, \cdots, y_m) \in R^N, \tag{1.2}$$

where $N = nm$ and $f_i^* : R^n \to R \cup \{+\infty\}$ are the convex conjugates of f_i, $i = 0, 1, \cdots, m$. Under the above assumptions on f_0, the conjugate function f_0^* is everywhere finite and continuously differentiable (see, for example, Han and Lou [7, §2]). Moreover its gradient ∇f_0^* is Lipschitz continuous with constant $1/\alpha$.

The algorithm to be proposed in this paper is designed to solve the dual problem (1.2) rather than the original problem (1.1). To this end, we shall first develop an algorithm for solving the following optimization problem that contains as a special case problem (1.2):

$$\text{minimize} \quad f(y) + g(y) \quad \text{over} \quad y \in R^N, \tag{1.3}$$

where $f : R^N \to R$ is convex and continuously differentiable and $g : R^N \to R \cup \{+\infty\}$ is closed proper convex.

Problems of the form (1.3) have drawn much attention in the literature under the assumption that separate minimizations of f and g are much easier than joint minimization of $f + g$. In particular, various splitting methods, which belong to the class of proximal point algorithm, have recently been studied extensively [2, 3, 15]. On the other hand, assuming differentiability of f and convexity of g, several authors [5, 10, 9] have proposed descent type methods with line search strategy. The method developed here belongs to the latter category but differs from the methods of [5, 10, 9] in the strategy to ensure global convergence. Specifically, we exploit a variant of trust region strategy, in which a quadratic term is added to the objective function of the subproblem, thereby restricting the step-size indirectly. The weight of the quadratic term, which may thus be regarded as a step-size parameter, is then adaptively controlled in a way similar to the conventional trust region strategy. This idea has

already been used by Fukushima et al. [6] to minimize an exact penalty function associated with a nonlinear programming problem. It is extended here to the more general problem (1.3).

The method developed for (1.3) will then be applied to problem (1.2). Because of the special structure of the latter problem, the subproblem to be solved at each iteration can be decomposed into m independent problems containing the individual variables y_i only. Therefore the resulting algorithm is particularly suitable for parallel computation. This algorithm shares much in common with the algorithms developed by Han and Lou [7, 8], Mouallif, Nguyen and Strodiot [11] and Tseng [15]. The main difference between the proposed algorithm and those of [7, 8, 11, 15] lies in the choice of step-size parameters. Specifically, the former algorithm automatically adjusts the values of the parameters in such a way that a descent property of the generated sequence is preserved, while the latter algorithms use a sequence of step-size parameters determined a priori so as to meet some threshold conditions and the generated sequence does not satisfy any descent property. As in [7, 8, 11, 15], we assume the strong convexity of the function f_0. But, in our method, the step-size parameters do not depend on the modulus of strong convexity of f_0, which is usually difficult to estimate for a general function. Our algorithm does not rely upon such a priori knowledge on the problem data. Moreover, it enjoys global convergence property under weaker assumptions than [8, 11]. Namely, we do not assume that the dual sequence $\{y^k\}$ is bounded and that the solution set of the Fenchel dual problem is nonempty and bounded. However, to prove that the step-size parameters are bounded above, we have to introduce a new assumption, which is satisfied by a large class of examples.

This paper is organized as follows. A descent algorithm for problem (1.3) is presented in Section 2 and its convergence is proved in Section 3. A decomposition algorithm for problem (1.1) is then developed in Section 4 and it is applied to special cases of (1.1) in Section 5. Finally in Section 6, we report some computational results with the latter algorithm on a parallel computer.

2. Basic Algorithm

Let us consider the problem

$$\text{(P)} \qquad \text{minimize} \quad f(y) + g(y) \quad \text{over} \quad y \in R^N,$$

where $f : R^N \to R$ is convex and continuously differentiable and $g : R^N \to R \cup \{+\infty\}$ is closed proper convex. It is well known that a point $y \in \operatorname{dom} g$ is a minimum of problem (P) if and only if it satisfies $0 \in \nabla f(y) + \partial g(y)$.

For each $y \in R^N$ and $d \in R^N$ we define

$$\Phi(y) = f(y) + g(y)$$

and

$$\Psi(y;d) = f(y) + \nabla f(y)^T d + g(y+d).$$

At any given point $y \in R^N$, the function $\Psi(y,\cdot)$ plays the role of a model function for Φ. The traditional trust region method [4] consists in looking to the model function only within a ball or a box centered at the current point y. Here instead of using the trust region constraint, we consider the problem

$$\text{minimize} \quad \frac{\lambda}{2}||d||^2 + \Psi(y;d) \quad \text{over} \quad d \in R^N, \tag{2.1}$$

where y is fixed and λ is a positive parameter. Intuitively it may be clear that controlling the value of parameter λ amounts to controlling the size of the trust region constraint. Thanks to the strong convexity of its objective function, problem (2.1) admits a unique optimal solution, which we denote $d(y,\lambda)$ and which is characterized by

$$0 \in \lambda d(y,\lambda) + \nabla f(y) + \partial g(y + d(y,\lambda)). \tag{2.2}$$

The algorithm can now be stated as follows:

Algorithm 1.

Step 1. Select parameters $0 < \mu_0 < \mu_1 < \mu_2 < 1$ and $\gamma > 1$. Choose an initial point $y^1 \in \text{dom}\, g$, $\lambda_1 > 0$ and set $k := 1$.

Step 2. Solve the subproblem

$$\text{minimize} \quad \frac{\lambda_k}{2}||d||^2 + \Psi(y^k;d) \quad \text{over} \quad d \in R^N$$

to obtain the solution d^k.

Step 3. Compute the actual change in the objective function Φ

$$\Delta\Phi(y^k,\lambda_k) := \Phi(y^k) - \Phi(y^k + d^k),$$

and the corresponding predicted change

$$\Delta\Psi(y^k,\lambda_k) := \Phi(y^k) - \Psi(y^k;d^k).$$

If $\Delta\Psi(y^k,\lambda_k) = 0$, then stop. (**comment:** y^k is a solution of (P).)

Step 4. Compute

$$q_k := \frac{\Delta\Phi(y^k,\lambda_k)}{\Delta\Psi(y^k,\lambda_k)}.$$

If $q_k \leq \mu_0$ then set $y^{k+1} := y^k$ and $\lambda_{k+1} := \gamma\lambda_k$. Otherwise set $y^{k+1} := y^k + d^k$ and

$$\lambda_{k+1} := \begin{cases} \gamma\lambda_k & \text{if } q_k \leq \mu_1 \\ \lambda_k & \text{if } \mu_1 < q_k \leq \mu_2 \\ \lambda_k/\gamma & \text{if } q_k > \mu_2 \end{cases}$$

Set $k := k + 1$ and go to Step 2.

By construction we always have $y^k + d^k \in \text{dom}\, g$, and hence the generated sequence $\{y^k\}$ is contained in dom g.

As in the trust region method, the ratio q_k determines the updating of parameter λ_k. If q_k is too small ($q_k \leq \mu_0$), then $\Psi(y^k; d^k)$ poorly approximates $\Phi(y^k + d^k)$ because $||d^k||$ is unduly large. In this case we do not accept the point $y^k + d^k$ and simply increase the value of λ_k to prevent the magnitude of the solution d^k of the subproblem from being too large. Otherwise, we accept $y^k + d^k$ as the next iterate while updating parameter λ_k according to the degree of agreement between $\Psi(y^k; d^k)$ and $\Phi(y^k + d^k)$. Specifically, if the agreement is not very good ($q_k \leq \mu_1$), then we still increase the value of λ_k, whereas if the agreement is quite satisfactory ($q_k > \mu_2$), then we decrease λ_k so as to take a larger step in the hope of achieving faster convergence to a solution.

3. Convergence of Basic Algorithm

In this section we establish global convergence of Algorithm 1. The following lemma will be useful in the sequel.

Lemma 3.1 *For each $y \in R^N$ and $\lambda > 0$, let $d(y, \lambda)$ be the unique solution of problem (2.1). Then the predicted change $\Delta\Psi(y, \lambda) \equiv \Phi(y) - \Psi(y; d(y, \lambda))$ satisfies*

$$\Delta\Psi(y, \lambda) \geq \frac{\lambda}{2}||d(y, \lambda)||^2.$$

Proof Since $d(y, \lambda)$ is the solution of (2.1), we have

$$\frac{\lambda}{2}||d(y, \lambda)||^2 + \Psi(y; d(y, \lambda)) \leq \Psi(y, 0) = \Phi(y),$$

that is,

$$\Delta\Psi(y, \lambda) = \Phi(y) - \Psi(y; d(y, \lambda)) \geq \frac{\lambda}{2}||d(y, \lambda)||^2. \qquad \square$$

Lemma 3.1 implies in particular that $\Delta\Psi(y^k, \lambda_k) \geq 0$ for each k. The next theorem justifies the stopping test in Step 3 of the algorithm.

Theorem 3.2 *For Algorithm 1, the following three conditions are equivalent:*
(a) $\Delta\Psi(y^k, \lambda_k) = 0$;
(b) $d^k = 0$;
(c) y^k *is a solution of problem* (P).

Proof Part (a) $\Leftrightarrow$ (b) is obvious from Lemma 3.1 and the definition of $\Delta\Psi(y^k, \lambda_k)$. To prove (b) $\Leftrightarrow$ (c), observe that the optimality condition (2.2) applied to d^k gives

$$0 \in \lambda^k d^k + \nabla f(y^k) + \partial g(y^k + d^k).$$

Therefore, by the uniqueness of d^k, $d^k = 0$ if and only if

$$0 \in \nabla f(y^k) + \partial g(y^k),$$

i.e., y^k is a solution of (P). $\square$

Let $h : R^N \to R \cup \{+\infty\}$ be a proper convex function. We denote by h_∞ the recession function of h (cf. [13, p. 66]), by $L_h = \{y \in R^N \mid h_\infty(y) = h_\infty(-y) = 0\}$ the constancy space of h (cf. [13, p. 69]) and by $L_h^\perp$ the orthogonal complement of L_h. Also, for any subset S of R^N, we shall denote by $\Pi_h(S)$ the orthogonal projection of S onto $L_h^\perp$. With these notations, it is obvious that each $y \in R^N$ can be uniquely decomposed into the sum $y = y_1 + y_2$ with $y_1 \in L_h^\perp$ and $y_2 \in L_h$.

The following lemmas play a fundamental role in proving convergence of Algorithm 1.

Lemma 3.3 *Let $h : R^N \to R \cup \{+\infty\}$ be a proper convex function and let L_h be its constancy space. Then, for each $y \in R^N$, we have $h(y) = h(y_1)$ and $\partial h(y) = \partial h(y_1) \subset L_h^\perp$, where $y = y_1 + y_2$ is the unique decomposition of y such that $y_1 \in L_h^\perp$ and $y_2 \in L_h$.*

Proof By [13, Theorem 8.8], we have $h(y) = h(y_1 + y_2) = h(y_1)$. Let $y^* \in \partial h(y)$. By the definition of the subdifferential, we have

$$h(w) \geq h(y) + (y^*)^T(w - y), \quad \forall w \in R^N,$$

which together with $h(y) = h(y_1)$ implies

$$h(w) \geq h(y_1) + (y^*)^T(w - y), \quad \forall w \in R^N. \tag{3.1}$$

In particular, for each w such that $w_1 = y_1$, we obtain

$$h(y_1) \geq h(y_1) + (y^*)^T(w_2 - y_2),$$

i.e.,

$$(y^*)^T(w_2 - y_2) \leq 0, \quad \forall w_2 \in L_h.$$

Since L_h is a vector subspace, we conclude that $y^* \in L_h^\perp$.

On the other hand, by (3.1), we also have

$$h(w) \geq h(y_1) + (y^*)^T(w - y_1) - (y^*)^T y_2, \quad \forall w \in R^N.$$

Since $y_2 \in L_h$ and $y^* \in L_h^\perp$, this inequality becomes

$$h(w) \geq h(y_1) + (y^*)^T(w - y_1), \quad \forall w \in R^N,$$

i.e., $y^* \in \partial h(y_1)$.

Similarly one can prove that $y^* \in \partial h(y_1)$ implies $y^* \in L_h^\perp$ and $y^* \in \partial h(y)$. $\square$

Lemma 3.4 *Let L_Φ be the constancy space of $\Phi = f + g$ and let Ω be a subset of R^N whose orthogonal projection $\Pi_\Phi(\Omega)$ onto $L_\Phi^\perp$ is bounded. Suppose there exists a selection $\{c(u)\}_{u \in \Pi_\Phi(\Omega)}$ such that $c(u) \in \partial g(u)$, $\forall u \in \Pi_\Phi(\Omega)$, and $\{c(u) \,|\, u \in \Pi_\Phi(\Omega)\}$ is bounded. Then, for every $y \in \Omega$, we have*

$$||d(y, \lambda)|| = O(1/\lambda),$$

where the right-hand side is independent of y.

Proof Let $y \in \Omega$. It follows from (2.2) that $d(y, \lambda)$ satisfies

$$-\nabla f(y) - \lambda d(y, \lambda) \in \partial g(y + d(y, \lambda)).$$

Since $\Phi = f + g$ is proper and convex, and since $y = u + v$ with $u \in \Pi_\Phi(\Omega)$ and $v \in L_\Phi$, we obtain, from Lemma 3.3, that

$$\nabla f(y) + \partial g(y) = \nabla f(u) + \partial g(u). \tag{3.2}$$

Moreover $\partial g(y)$ is nonempty because by assumption $\partial g(u)$ is nonempty. So, by the monotonicity of the subdifferential [13, Corollary 31.5.2], for each $y^* \in \partial g(y)$, we have

$$(-\nabla f(y) - \lambda d(y, \lambda) - y^*)^T (y + d(y, \lambda) - y) \geq 0,$$

which implies

$$\begin{aligned} \lambda ||d(y, \lambda)||^2 &\leq (-\nabla f(y) - y^*)^T d(y, \lambda) \\ &\leq ||\nabla f(y) + y^*|| \cdot ||d(y, \lambda)||, \end{aligned}$$

i.e.,

$$||d(y, \lambda)|| \leq \frac{1}{\lambda} ||\nabla f(y) + y^*||.$$

Hence, using (3.2), we obtain

$$||d(y,\lambda)|| \leq \frac{1}{\lambda}||\nabla f(u) + u^*||, \quad \forall u^* \in \partial g(u).$$

Since $c(u) \in \partial g(u)$, the last inequality is, in particular, true for $u^* = c(u)$.

Finally from the continuity of ∇f and the boundedness of the sets $\Pi_\Phi(\Omega)$ and $\{c(u) \mid u \in \Pi_\Phi(\Omega)\}$, we conclude that there exists a constant $C > 0$ such that

$$||\nabla f(u) + c(u)|| \leq C, \quad \forall u \in \Pi_\Phi(\Omega).$$

The proof is then complete. □

Lemma 3.5 *Let L_Φ be the constancy space of $\Phi = f + g$ and let Ω be a subset of R^N which does not contain any solution of problem* (P). *Suppose that the orthogonal projection $\Pi_\Phi(\Omega)$ of Ω onto $L_\Phi^\perp$ is bounded and that there exists a selection $\{c(u)\}_{u \in \Pi_\Phi(\Omega)}$ such that $c(u) \in \partial g(u)$, $\forall u \in \Pi_\Phi(\Omega)$, and $\{c(u) \mid u \in \Pi_\Phi(\Omega)\}$ is bounded. Then, for any $\rho \in (0,1)$, there exists a constant $\overline{\lambda}$ such that $\lambda \geq \overline{\lambda}$ implies*

$$q(y,\lambda) \equiv \frac{\Delta\Phi(y,\lambda)}{\Delta\Psi(y,\lambda)} \geq \rho, \quad \forall y \in \Omega,$$

where

$$\Delta\Phi(y,\lambda) \equiv \Phi(y) - \Phi(y + d(y,\lambda)),$$

$$\Delta\Psi(y,\lambda) \equiv \Phi(y) - \Psi(y; d(y,\lambda)).$$

Proof It suffices to show that $q(y,\lambda) \to 1$ uniformly in y as $\lambda \to \infty$. By Lemma 3.4 we have

$$\Phi(y + d(y,\lambda)) - \Psi(y; d(y,\lambda)) = o(||d(y,\lambda)||) = o(1/\lambda). \tag{3.3}$$

Then it follows from the definition of $q(y,\lambda)$ and (3.3) that

$$q(y,\lambda) = 1 + \frac{o(1/\lambda)}{\Delta\Psi(y,\lambda)}, \tag{3.4}$$

where the denominator on the right-hand side is positive by Lemma 3.1 and Theorem 3.2. But Lemmas 3.1 and 3.4 imply

$$\Delta\Psi(y,\lambda) \geq \frac{\lambda}{2}||d(y,\lambda)||^2 = O(1/\lambda). \tag{3.5}$$

Combining (3.4) and (3.5), and noting that both $o(1/\lambda)$ and $O(1/\lambda)$ are independent of y, we see that $q(y,\lambda) \to 1$ uniformly as $\lambda \to \infty$. This completes the proof. □

Now we state a convergence theorem for Algorithm 1. The proof is different from that of [6], because the function g may take the value $+\infty$ in the present case.

Theorem 3.6 *Let $\{y^k\}$ be a sequence generated by Algorithm 1 and let $\{u^k\}$ be its orthogonal projection onto the orthogonal complement $L_\Phi^\perp$ of the constancy space of $\Phi = f + g$. If $0 \in \mathrm{ri}(\mathrm{dom}\,\Phi^*)$, then the sequence $\{u^k\}$ is bounded. If, in addition, there exists a bounded sequence $\{c^k\}$ such that $c^k \in \partial g(u^k)$ for each k, then the sequence $\{\lambda_k\}$ is bounded above. Finally, if ∇f is Lipschitz continuous on the level set $\Lambda(y^1) \equiv \{y \in R^N \mid \Phi(y) \le \Phi(y^1)\}$ and $\{\lambda_k\}$ is bounded away from zero, then we have*

(i) $\mathrm{dist}\,\Big(0, \nabla f(y^k) + \partial g(y^k)\Big) \to 0 \quad as \quad k \to \infty,$

(ii) $\lim_{k\to\infty} \Phi(y^k) = \inf\{\Phi(y) \mid y \in R^N\},$

(iii) $\mathrm{dist}\,(y^k, S^*) \to 0 \quad as \quad k \to \infty,$

where S^* is the set of optimal solutions of problem (P) *and* $\mathrm{dist}\,(z, S) \equiv \min\{||y - z|| \mid y \in S\}$ *denotes the distance from a point z to a closed convex subset S of R^N.*

Proof Let Ω denote the set of iterates $\{y^k\}$ generated by the algorithm and let L_Φ be the constancy space of Φ. Then each $y^k \in \Omega$ can be decomposed into the sum $u^k + v^k$ such that $u^k \in L_\Phi^\perp$ and $v^k \in L_\Phi$. Assuming $0 \in \mathrm{ri}(\mathrm{dom}\,\Phi^*)$, we prove that $\Pi_\Phi(\Omega) = \{u^k\}$ is bounded in R^N. Since, for each k, $y^k \in \Lambda(y^1)$ and $\Phi(y^k) = \Phi(u^k)$ (cf. Lemma 3.3), the set $\Pi_\Phi(\Omega)$ is a subset of $\Lambda(y^1) \cap L_\Phi^\perp$. By [13, Corollary 13.3.4], $0 \in \mathrm{ri}(\mathrm{dom}\,\Phi^*)$ implies

$$\Phi_\infty(u) > 0 \quad \text{for each } u \in L_\Phi^\perp, \;\; u \neq 0,$$

where Φ_∞ denotes the recession function of Φ. By [13, Theorem 8.5], this means that for each $u \in L_\Phi^\perp, \; u \neq 0$, there exists $\lambda > 0$ such that

$$\Phi(u^1 + \lambda u) - \Phi(u^1) > 0,$$

i.e.,

$$u^1 + \lambda u \notin \Lambda(y^1) \cap L_\Phi^\perp.$$

Consequently, the vectors $u \in L_\Phi^\perp$, $u \neq 0$, are never directions of recession of the set $\Lambda(y^1) \cap L_\Phi^\perp$ (cf. [13, p. 61]). In other words, the recession cone of $\Lambda(y^1) \cap L_\Phi^\perp$ consists of the zero vector alone and, by [13, Theorem 8.4], $\Lambda(y^1) \cap L_\Phi^\perp$ is bounded. Thus $\Pi_\Phi(\Omega)$ being a subset of $\Lambda(y^1) \cap L_\Phi^\perp$ is also bounded.

Next suppose that there exists a bounded sequence $\{c^k\}$ such that $c^k \in \partial g(u^k)$ for each k. It then follows from Lemma 3.5 that there exists a sufficiently large constant $\overline{\lambda}$ such that $\lambda_k \ge \overline{\lambda}$ implies $q_k \ge \mu_1$, where q_k is the ratio computed in Step 4 and μ_1 is the positive constant chosen in Step 1. We may assume without loss of generality that $\overline{\lambda} \ge \lambda_1$, where λ_1 is the initial value of λ_k. We prove the following inequalities by induction:

$$\lambda_k \le \gamma\overline{\lambda} \quad \text{for all } k, \tag{3.6}$$

where $\gamma > 1$ is the constant chosen in Step 1. The inequality (3.6) is obvious for $k = 1$. Then suppose (3.6) is true for k and consider λ_{k+1}. Note that we always have $\lambda_{k+1} \leq \gamma\lambda_k$. Therefore, when $\lambda_k < \overline{\lambda}$, we have $\lambda_{k+1} < \gamma\overline{\lambda}$. On the other hand, when $\lambda_k \geq \overline{\lambda}$, we have $q_k \geq \mu_1$, and hence $\lambda_{k+1} \leq \lambda_k$ by Step 4 of the algorithm. The induction hypothesis then implies that (3.6) is also true for $k+1$. So we have proven that $\{\lambda_k\}$ is bounded above.

To prove the last part of the theorem, we first show that

$$||y^{k+1} - y^k|| \longrightarrow 0 \quad \text{as} \quad k \to \infty. \tag{3.7}$$

Since $q_k \leq \mu_0$ implies $y^{k+1} = y^k$, it suffices to consider the subsequence $\{y^k\}_{k \in \hat{K}}$ such that $q_k > \mu_0$ for each $k \in \hat{K}$. Then we have

$$\begin{aligned} \Delta\Phi(y^k, \lambda_k) &> \mu_0 \Delta\Psi(y^k, \lambda_k) \\ &\geq \frac{\mu_0 \lambda_k}{2} ||y^{k+1} - y^k||^2 \quad \text{for each} \quad k \in \hat{K}, \end{aligned} \tag{3.8}$$

where the last inequality follows from Lemma 3.1. On the other hand, since the sequence $\{\Phi(y^k)\}_{k \in \hat{K}}$ is decreasing and bounded from below, we have

$$\begin{aligned} \sum_{k \in \hat{K}} \Delta\Phi(y^k, \lambda_k) &= \sum_{k=1}^{\infty} \{\Phi(y^k) - \Phi(y^{k+1})\} \\ &= \Phi(y^1) - \lim_{k \to \infty} \Phi(y^k) < \infty, \end{aligned}$$

which implies that

$$\Delta\Phi(y^k, \lambda_k) \to 0 \quad \text{as} \quad k \to \infty, \quad k \in \hat{K}.$$

Thus by (3.8) we obtain

$$\lambda_k ||y^{k+1} - y^k||^2 \to 0 \quad \text{as} \quad k \to \infty, \quad k \in \hat{K},$$

and hence (3.7), since $\{\lambda_k\}$ is bounded away from zero.

Now since $y^{k+1} = y^k + d^k$ for each $k \in \hat{K}$ and since d^k satisfies the optimality condition (2.2) for $y = y^k$ and $\lambda = \lambda_k$, we have

$$\begin{aligned} 0 &\in \lambda_k(y^{k+1} - y^k) + \nabla f(y^k) + \partial g(y^{k+1}) \\ &= \lambda_k(y^{k+1} - y^k) + \nabla f(y^k) - \nabla f(y^{k+1}) + \nabla f(y^{k+1}) + \partial g(y^{k+1}). \end{aligned}$$

Therefore, for $k \in \hat{K}$, we have

$$\begin{aligned} \text{dist}\,(0, \nabla f(y^{k+1}) + \partial g(y^{k+1})) &\leq \lambda_k ||y^{k+1} - y^k|| + ||\nabla f(y^k) - \nabla f(y^{k+1})|| \\ &\leq (\lambda_k + \sigma) ||y^{k+1} - y^k||, \end{aligned} \tag{3.9}$$

where $\sigma > 0$ is a Lipschitz constant for ∇f. Since $\{\lambda_k\}$ is bounded above (see (3.6)), it follows from (3.7) and (3.9) that

$$\text{dist } \Big(0, \nabla f(y^{k+1}) + \partial g(y^{k+1})\Big) \to 0 \quad \text{as} \quad k \to \infty, \quad k \in \hat{K}. \tag{3.10}$$

Since $y^{k+1} = y^k$ for $k \notin \hat{K}$, (3.10) implies that part (i) holds. The rest of the proof follows from [1, Theorem 2.3]. $\square$

The assumption that there exists a bounded sequence $\{c^k\}$ such that $c^k \in \partial g(u^k)$ for each k, will be verified for a large class of examples in the next section.

The assumption that the sequence $\{\lambda_k\}$ is bounded away from zero can be assured by slightly modifying the updating rule of λ_k in Step 4 of the algorithm. Specifically, by setting $\lambda_{k+1} := \max\{\underline{\lambda}, \lambda_k/\gamma\}$ when $q_k > \mu_2$, we always have $\lambda_k \geq \underline{\lambda}$, where $\underline{\lambda}$ is a sufficiently small positive constant.

Corollary 3.7 *In addition to the assumptions of Theorem 3.6, we suppose that* $0 \in \text{int}(\text{dom}\,\Phi^*)$. *Then the sequence* $\{y^k\}$ *generated by Algorithm 1 is bounded and each accumulation point of* $\{y^k\}$ *is a solution of* (P).

Proof Since the affine hull of $\text{dom}\,\Phi^*$ is R^N, the result follows directly from [1, Theorem 2.3]. $\square$

4. Parallel Decomposition Algorithm

Now let us turn our attention to the following optimization problem:

$$\text{(Q)} \qquad \text{minimize} \quad f_0(x) + \sum_{i=1}^{m} f_i(x) \quad \text{over} \quad x \in R^n,$$

where $f_0 : R^n \to R$ is differentiable and strongly convex with modulus α and $f_i : R^n \to R \cup \{+\infty\}$, $i = 1, \cdots, m$, are closed proper convex functions. In addition, we assume that $\text{ri}(\text{dom}\, f_i)$, $i = 1, \cdots, m$, have a point in common. As mentioned in Introduction, the class of problems (Q) is broad enough to contain ordinary convex programming problems with a strongly convex objective function. Further examples of problems (Q) may be found in Mouallif, Nguyen and Strodiot [11].

Associated with problem (Q) is the Fenchel dual:

$$\text{(D)} \qquad \text{minimize} \quad f_0^*(y_1 + \cdots + y_m) + \sum_{i=1}^{m} f_i^*(-y_i) \quad \text{over} \quad (y_1, \cdots, y_m) \in R^N,$$

where $N = mn$ and $f_i^* : R^n \to R \cup \{+\infty\}$ are the conjugate functions of f_i, $i = 0, 1, \cdots, m$. Denote $y = (y_1, \cdots, y_m) \in R^N$ and define the proper convex functions $F : R^N \to R$ and $G : R^N \to R \cup \{+\infty\}$ by

$$F(y) = f_0^*(y_1 + \cdots + y_m) \tag{4.1}$$

and

$$G(y) = \sum_{i=1}^{m} f_i^*(-y_i), \tag{4.2}$$

respectively. Then problem (D) consists in minimizing the sum of the functions F and G over $y \in R^N$. Moreover, under the given assumptions on f_0, the function F is strictly convex and continuously differentiable on R^N (see for example [7, §2] or [8, §3]). Thus problem (D) belongs to the class of problem (P) considered earlier, and hence can be solved using Algorithm 1 described in Section 2.

The focal point in applying Algorithm 1 to problem (D) consists in how the subproblems of Step 2 are dealt with. In the present case, the subproblem can be written as

$$\text{minimize} \quad \frac{\lambda_k}{2}||d||^2 + \Psi(y^k; d) \quad \text{over} \quad d \in R^N, \tag{4.3}$$

where $y^k = (y_1^k, \cdots, y_m^k)$ is the k-th iterate and

$$\Psi(y^k; d) = F(y^k) + \nabla F(y^k)^T d + G(y^k + d).$$

Let $d = (d_1, \cdots, d_m)$ in conformity with $y = (y_1, \cdots, y_m)$. Then the function Ψ may also be expressed as

$$\Psi(y^k; d) = F(y^k) + \sum_{i=1}^{m} \{\nabla f_0^*(y_1^k + \cdots + y_m^k)^T d_i + f_i^*(-y_i^k - d_i)\}.$$

Thus subproblem (4.3) can be split into m independent subproblems $(\text{AQ})_i$, $i = 1, 2, \cdots, m$, as follows:

$$(\text{AQ})_i \qquad \text{minimize} \quad \frac{\lambda_k}{2}||d_i||^2 + \nabla f_0^*(y_1^k + \cdots + y_m^k)^T d_i + f_i^*(-y_i^k - d_i) \quad \text{over} \quad d_i \in R^n,$$

where the constant term $F(y^k)$ has been dropped from the objective function.

Before solving $(\text{AQ})_i$ we have to compute the vector $\nabla f_0^*(y_1^k + \cdots + y_m^k)$. This can be done by solving the problem

$$\text{minimize} \quad f_0(x) - \left(\sum_{i=1}^{m} y_i^k\right)^T x \qquad \text{over} \quad x \in R^n. \tag{4.4}$$

Since f_0 is closed, co-finite and strictly convex, problem (4.4) admits a unique solution, which we denote x^k, for any given $(y_1^k, \cdots, y_m^k)$. By the optimality condition for problem (4.4), x^k satisfies

$$0 \in \partial f_0(x^k) - \sum_{i=1}^{m} y_i^k.$$

It then follows from [13, Corollary 23.5.1] and the differentiability of f_0^* that

$$x^k = \nabla f_0^*(y_1^k + \cdots + y_m^k). \tag{4.5}$$

Moreover by the definition of the conjugate function, $f_0^*(y_1^k + \cdots + y_m^k)$ is equal to the negative of the optimal value of problem (4.4).

We may thus restate subproblems $(\mathrm{AQ})_i$ as

$$(\mathrm{AQ})_i \quad \text{minimize} \quad \frac{\lambda_k}{2}||d_i||^2 + (x^k)^T d_i + f_i^*(-y_i^k - d_i) \quad \text{over} \quad d_i \in R^n.$$

Unfortunately these problems may still be intractable, because it is usually impractical to evaluate the conjugate functions f_i^* at various points in the course of solving problems $(\mathrm{AQ})_i$. To avoid this difficulty, we shall again consider the Fenchel dual of $(\mathrm{AQ})_i$. Define the quadratic function $\phi_i : R^n \to R$ and the closed proper convex function $\psi_i : R^n \to R \cup \{+\infty\}$ by

$$\phi_i(d_i) = \frac{\lambda_k}{2}||d_i||^2 + (x^k)^T d_i$$

and

$$\psi_i(d_i) = f_i^*(-y_i^k - d_i),$$

respectively. Then the conjugates of these functions are given as

$$\phi_i^*(z_i) = \frac{1}{2\lambda_k}||z_i - x^k||^2$$

and

$$\psi_i^*(z_i) = f_i(-z_i) - (y_i^k)^T z_i.$$

Thus the Fenchel dual of $(\mathrm{AQ})_i$ is explicitly stated as

$$(\mathrm{DAQ})_i \quad \text{minimize} \quad \frac{1}{2\lambda_k}||z_i - x^k||^2 + (y_i^k)^T z_i + f_i(z_i) \quad \text{over} \quad z_i \in R^n.$$

By Fenchel's duality theorem [13, Section 31], we have

$$\min\ (\mathrm{AQ})_i + \min\ (\mathrm{DAQ})_i = 0 \tag{4.6}$$

for all i, where $\min\ (\mathrm{AQ})_i$ and $\min\ (\mathrm{DAQ})_i$ denote the optimal values of problems $(\mathrm{AQ})_i$ and $(\mathrm{DAQ})_i$, respectively. Moreover if z_i^k is the solution of $(\mathrm{DAQ})_i$, then the vector d_i^k given by

$$d_i^k = \frac{1}{\lambda_k}(z_i^k - x^k)$$

is the solution of $(\mathrm{AQ})_i$. Indeed, if z_i^k solves $(\mathrm{DAQ})_i$, it follows from the optimality condition that

$$0 \in \frac{1}{\lambda_k}(z_i^k - x^k) + y_i^k + \partial f_i(z_i^k),$$

or by the definition of d_i^k

$$0 \in d_i^k + y_i^k + \partial f_i(\lambda_k d_i^k + x^k).$$

Then by [13, Corollary 23.5.2] we have

$$\lambda_k d_i^k + x^k \in \partial f_i^*(-y_i^k - d_i^k),$$

that is,

$$0 \in \lambda_k d_i^k + x^k - \partial f_i^*(-y_i^k - d_i^k).$$

But this means that d_i^k solves problem $(\mathrm{AQ})_i$.

Before stating the algorithm, it only remains to make clear how to compute the quantities $\Delta\Phi(y^k, \lambda_k)$ and $\Delta\Psi(y^k, \lambda_k)$. First observe that

$$\Delta\Phi(y^k, \lambda_k) = f_0^*\left(\sum_{i=1}^m y_i^k\right) - f_0^*\left(\sum_{i=1}^m (y_i^k + d_i^k)\right) + \sum_{i=1}^m \left\{f_i^*(-y_i^k) - f_i^*(-y_i^k - d_i^k)\right\}$$

and

$$\Delta\Psi(y^k, \lambda_k) = \sum_{i=1}^m \left\{f_i^*(-y_i^k) - f_i^*(-y_i^k - d_i^k) - (x^k)^T d_i^k\right\}.$$

Since

$$\min\ (\mathrm{AQ})_i = \frac{\lambda_k}{2}||d_i^k||^2 + (x^k)^T d_i^k + f_i^*(-y_i^k - d_i^k),$$

it follows from (4.6) that

$$f_i^*(-y_i^k - d_i^k) = -\min\ (\mathrm{DAQ})_i - \frac{\lambda_k}{2}||d_i^k||^2 - (x^k)^T d_i^k.$$

On the other hand, $f_0^*(\sum_{i=1}^m (y_i^k + d_i^k))$ is equal to the negative of the optimal value of the problem

$$\text{minimize} \quad f_0(x) - \sum_{i=1}^m (y_i^k + d_i^k)^T x \quad \text{over} \quad x \in R^n. \tag{4.7}$$

Since $f_0^*(\sum_{i=1}^m y_i^k)$ and $f_i^*(-y_i^k), i = 1, \cdots, m$, have already been computed at the previous iteration, the two quantities $\Delta\Phi(y^k, \lambda_k)$ and $\Delta\Psi(y^k, \lambda_k)$ are thus obtained with little computational overheads. Note that solution of problem (4.7) gives as a byproduct the gradient vector $\nabla f_0^*(\sum_{i=1}^m (y_i^k + d_i^k))$, which together with the function value $f_0^*(\sum_{i=1}^m (y_i^k + d_i^k))$ will be used at the next iteration (see (4.4) and (4.5)).

To summarize, we obtain the following decomposition algorithm for problem (Q).

Algorithm 2.

Step 1. Select parameters $0 < \mu_0 < \mu_1 < \mu_2 < 1$ and $\gamma > 1$. Choose an initial point $y^1 = (y_1^1, \cdots, y_m^1) \in R^N$ and $\lambda_1 > 0$. For $i = 1, \cdots, m$, compute

$$t_i^1 = f_i^*(-y_i^1) \ \left(= -\min_{x \in R^n}\{f_i(x) + (y_i^1)^T x\}\right).$$

Solve the problem

$$\text{minimize} \quad f_0(x) - \sum_{i=1}^m (y_i^1)^T x \quad \text{over} \quad x \in R^n,$$

and let x^1 and t_0^1 be its optimal solution and the negative of the optimal value, respectively. Set $k := 1$. (**comment:** $x^1 = \nabla f_0^*(\sum_{i=1}^m y_i^1)$, $t_0^1 = f_0^*(\sum_{i=1}^m y_i^1)$.)

Step 2. For $i = 1, \cdots, m$, solve the subproblems

$$(\mathrm{DAQ})_i \quad \text{minimize} \quad \frac{1}{2\lambda_k}||z_i - x^k||^2 + (y_i^k)^T z_i + f_i(z_i) \quad \text{over} \quad z_i \in R^n,$$

and let z_i^k and v_i^k be their optimal solutions and optimal values, respectively. For $i = 1, \cdots, m$, set

$$d_i^k := \frac{1}{\lambda_k}(z_i^k - x^k)$$

and

$$\bar{t}_i^k := -v_i^k - \frac{\lambda_k}{2}||d_i^k||^2 - (x^k)^T d_i^k.$$

(**comment:** $\bar{t}_i^k = f_i^*(-y_i^k - d_i^k)$.)

Step 3. Compute

$$\Delta\Psi(y^k, \lambda_k) = \sum_{i=1}^m \{t_i^k - \bar{t}_i^k - (d_i^k)^T x^k\}.$$

If $\Delta\Psi(y^k, \lambda_k) = 0$, then stop. (**comment:** y^k and x^k are optimal solutions of (D) and (Q), respectively.)

Step 4. For $i = 1, \cdots, m$, set

$$\bar{y}_i^k := y_i^k + d_i^k.$$

Solve the problem

$$\text{minimize} \quad f_0(x) - \sum_{i=1}^m (\bar{y}_i^k)^T x \quad \text{over} \quad x \in R^n,$$

and let $\bar{x}^k$ and $\bar{t}_0^k$ be its optimal solution and the negative of the optimal value, respectively. Compute

$$\Delta\Phi(y^k, \lambda_k) = t_0^k - \bar{t}_0^k + \sum_{i=1}^m \{t_i^k - \bar{t}_i^k\}.$$

(**comment:** $\bar{x}^k = \nabla f_0^*(\sum_{i=1}^m (y_i^k + d_i^k))$, $\bar{t}_0^k = f_0^*(\sum_{i=1}^m (y_i^k + d_i^k))$.)

Step 5. Compute

$$q_k := \frac{\Delta\Phi(y^k, \lambda_k)}{\Delta\Psi(y^k, \lambda_k)}.$$

If $q_k \leq \mu_0$, then set

$$\begin{aligned} x^{k+1} &:= x^k, \\ y_i^{k+1} &:= y_i^k, \quad i = 1, \cdots, m, \\ t_i^{k+1} &:= t_i^k, \quad i = 0, 1, \cdots, m, \\ \lambda_{k+1} &:= \gamma\lambda_k. \end{aligned}$$

Otherwise, set

$$\begin{aligned} x^{k+1} &:= \bar{x}^k, \\ y_i^{k+1} &:= \bar{y}_i^k, \quad i = 1, \cdots, m, \\ t_i^{k+1} &:= \bar{t}_i^k, \quad i = 0, 1, \cdots, m, \end{aligned}$$

$$\lambda_{k+1} := \begin{cases} \gamma\lambda_k & \text{if } q_k \le \mu_1 \\ \lambda_k & \text{if } \mu_1 < q_k \le \mu_2 \\ \lambda_k/\gamma & \text{if } q_k > \mu_2 \end{cases}$$

Set $k := k + 1$ and go to Step 2.

This algorithm is similar to the ones of Han and Lou [8] and Mouallif, Nguyen and Strodiot [11]. But unlike those algorithms, the proposed algorithm automatically adjusts the controlling parameters $\{\lambda_k\}$. As shown in the next theorem, it enjoys global convergence property under weaker assumptions than [8, 11]. Namely, we do not assume that the dual sequence $\{y^k\}$ is bounded and that the parameters $\{\lambda_k\}$ must be chosen (or fixed) below a certain threshold value that is related to the (usually unknown) modulus of strong convexity of f_0.

Theorem 4.1 *Let $\{y^k\}$ be the dual sequence generated by Algorithm 1. Then its orthogonal projection $\{u^k\}$ onto the orthogonal complement $L_{F+G}^{\perp}$ of the constancy space of $F + G$ is a bounded sequence in R^N, where F and G are defined by (4.1) and (4.2), respectively. Moreover, if $\{\lambda_k\}$ is bounded away from zero and if there exists a bounded sequence $\{c^k\}$ such that $c^k \in \partial G(u^k)$ for each k, then*

$$\text{dist } \Big(0, \nabla F(y^k) + \partial G(y^k)\Big) \longrightarrow 0 \quad \textit{as} \quad k \to \infty \tag{4.8}$$

and the sequence $\{x^k\}$ given by

$$x^k = \nabla f_0^*(y_1^k + \cdots + y_m^k), \qquad k = 1, 2, \cdots$$

converges to the unique solution of the primal problem (Q). *In particular, if the algorithm terminates at Step 3, then the last iterates y^k and x^k are solutions of problems* (D) *and* (Q), *respectively.*

Proof In order to apply Theorem 3.6, we shall show that $0 \in \text{ri}(\text{dom } (F + G)^*)$ and that ∇F is Lipschitz continuous on the level set $\Lambda(y^1) = \{y \in R^N \mid F(y) + G(y) \le F(y^1) + G(y^1)\}$. Note that the objective function $F + G$ of the dual problem (D) is the conjugate of the perturbation function $H : R^N \to R \cup \{+\infty\}$ for the primal problem (Q), which is defined by

$$H(v) = \inf_{x \in R^n} \left\{ f_0(x) + \sum_{i=1}^{m} f_i(x - v_i) \right\}, \tag{4.9}$$

where $v = (v_1, \cdots, v_m) \in R^N$ with $v_i \in R^n, i = 1, \cdots, m$. Namely, we have

$$(F + G)(y) = H^*(y) \tag{4.10}$$

for all $y \in R^N$. Since, by assumption on problem (Q), $\mathrm{dom}\, f_0 = R^n$ and there exists $x \in R^n$ such that $x \in \mathrm{ri}\,(\mathrm{dom}\, f_i), i = 1, \cdots, m$, it follows that

$$0 \in \mathrm{ri}\,(\mathrm{dom}\, H). \tag{4.11}$$

Since by [13, Corollary 7.4.1] $\mathrm{ri}\,(\mathrm{dom}\, H) = \mathrm{ri}\,(\mathrm{dom}\, H^{**})$, (4.11) and (4.10) imply

$$0 \in \mathrm{ri}\,(\mathrm{dom}\, H^{**}) = \mathrm{ri}\,(\mathrm{dom}\, (F + G)^*).$$

On the other hand, since ∇f_0^* is Lipschitz continuous by the strong monotonicity assumption on f_0, it follows from (4.1) that ∇F is Lipschitz continuous. So, by Theorem 3.6, the sequence $\{u^k\}$ is bounded and (4.8) is satisfied.

Moreover, since f_0 is essentially strictly convex [13, page 253], we can apply the results of [1, §4.2] to show that the sequence $\{x^k\}$ converges to the optimal solution of the primal problem (Q). □

To finish this section, we demonstrate that the assumption

there exists a bounded sequence $\{c^k\}$ such that $c^k \in \partial G(u^k)$ for each k

is satisfied for a large class of problems.

Consider the problem

$$\text{minimize} \quad f_0(x) \quad \text{subject to} \quad x \in C = C_1 \cap \cdots \cap C_m, \tag{4.12}$$

where C_i are closed convex subsets of R^n such that $\cap_{i=1}^m \mathrm{ri}\, C_i \neq \emptyset$ and $f_0 : R^n \to R$ is differentiable and strongly convex. By setting $f_i(x) = \delta(x|C_i), i = 1, \cdots, m$, where $\delta(\cdot|C_i)$ are indicator functions of the convex sets C_i, problem (4.12) reduces to a problem of form (Q). Moreover each function f_i is closed proper convex and $\mathrm{ri}(\mathrm{dom}\, f_i), i = 1, \cdots, m$, have a point in common. The function $G : R^N \to R \cup \{+\infty\}$ is defined by

$$G(y_1, \cdots, y_m) = \sum_{i=1}^m f_i^*(-y_i) = \sum_{i=1}^m \delta^*(-y_i|C_i). \tag{4.13}$$

In the following, we shall denote for simplicity

$$\delta^*(-\cdot\ |C_i) = \delta^*(\cdot\,|C_i) \circ (-I),$$

where $\circ$ denotes composition and I is the identity mapping. Then we immediately have

$$\mathrm{dom}\, G = \prod_{i=1}^m \mathrm{dom}\, \delta^*(-\cdot\ |C_i) \tag{4.14}$$

and

$$\partial G(y_1, \cdots, y_m) = \prod_{i=1}^{m} \partial \delta^*(- \cdot \,|C_i)(y_i). \tag{4.15}$$

Theorem 4.2 *Let G be defined by (4.13). If for each i, either C_i is bounded or C_i is a halfspace $\{x \,|\, a_i^T x \leq b_i\}$, where $a_i \in R^n$ and $b_i \in R$, then there exists a bounded sequence $\{c^k\}$ such that $c^k \in \partial G(u^k)$ for each k.*

Proof By definition, we have

$$\delta^*(-y_i|C_i) = \sup\{(-y_i)^T x \,|\, x \in C_i\} = \sup\{(y_i)^T x \,|\, -x \in C_i\}. \tag{4.16}$$

Let J be a (possibly empty) subset of $\{1, \cdots, m\}$ and suppose that C_i is bounded for $i \in J$ and $\{x \,|\, a_i^T x \leq b_i\}$ for $i \notin J$.

For each $i \in J$, the boundedness of C_i ensures that the supremum in (4.16) is attained for all y_i, implying that

$$\operatorname{dom} \delta^*(- \cdot \,|\, C_i) = R^n.$$

Moreover it is not difficult to see that for each y_i

$$\partial \delta^*(- \cdot \,|\, C_i)(y_i) = \arg\max\{(y_i)^T x \,|\, -x \in C_i\} \subset -C_i. \tag{4.17}$$

On the other hand, for each $i \notin J$, we have

$$\delta^*(-y_i|C_i) = \begin{cases} \lambda b_i, & \text{if } y_i = -\lambda a_i,\ \lambda \geq 0 \\ +\infty, & \text{otherwise,} \end{cases}$$

i.e.,

$$\operatorname{dom} \delta^*(- \cdot \,|\, C_i) = \{y_i |\, y_i = -\lambda a_i,\ \lambda \geq 0\}. \tag{4.18}$$

In particular, we have

$$\delta^*(\lambda a_i|C_i) = \lambda b_i \quad \text{for } \lambda \geq 0$$

and

$$\partial \delta^*(- \cdot |C_i)(-\lambda a_i) = \begin{cases} \{-x \,|\, a_i^T x \leq b_i\}, & \text{if } \lambda = 0 \\ \{-x \,|\, a_i^T x = b_i\}, & \text{if } \lambda > 0. \end{cases} \tag{4.19}$$

Since $\operatorname{dom} F = R^N$ and since the sequence $\{y^k\}$ is in $\operatorname{dom} G$, its projection $\{u^k\}$ onto $L_{F+G}^{\perp}$ is also in $\operatorname{dom} G$. Consequently, by (4.14), for each k, the vector $u^k = (u_1^k, \cdots, u_m^k)$ satisfies

$$u_i^k \in \operatorname{dom} \delta^*(- \cdot |C_i), \quad i = 1, \cdots, m. \tag{4.20}$$

So, for each k, we define the vector $c^k = (c_1^k, \cdots, c_m^k)$ in the following manner: For $i \in J$, let c_i^k be an arbitrary element of $\partial \delta^*(- \cdot |C_i)(u_i^k)$ (cf. (4.17)), while for $i \notin J$, let c_i^k be an arbitrary vector

$-x_i^*$ (independent of k) such that $a_i^T x_i^* = b_i$, which is an element of $\partial\delta^*(- \cdot |C_i)(u_i^k)$ by (4.18), (4.19) and (4.20). Then it follows from (4.15) that $c^k \in \partial G(u^k)$ for all k. Moreover the sequence $\{c^k\}$ is bounded, because $\{c_i^k\}$ is contained in the bounded set $-C_i$ for each $i \in J$ (cf. (4.17)), and $\{c_i^k\}$ stays at a constant vector $-x_i^*$ for each $i \notin J$. $\square$

5. Special Cases

To see how Algorithm 2 works, we shall consider two typical examples that belong to the class of problem (Q). The first example is the problem

$$\text{minimize} \quad \frac{1}{2}||x||^2 \quad \text{subject to} \quad x \in C = C_1 \cap \cdots \cap C_m, \tag{5.1}$$

where C_i are closed convex subsets of R^n such that $\bigcap_{i=1}^m C_i \neq \emptyset$. This problem consists in finding the vector of minimum norm in the convex set C [7]. By setting

$$f_0(x) = \frac{1}{2}||x||^2 \quad \text{and} \quad f_i(x) = \delta(x|C_i), \quad i = 1, \cdots, m,$$

where $\delta(\cdot|C_i)$ are the indicator functions of the convex sets C_i, problem (5.1) reduces to a problem of form (Q). For this problem, we can easily compute various quantities that appear in Algorithm 2. For instance, the auxiliary problem

$$\text{minimize} \quad f_0(x) - \sum_{i=1}^m (\bar{y}_i^k)^T x \quad \text{over} \quad x \in R^n,$$

in Step 4 can be solved explicitly and yields

$$\bar{x}^k = \bar{y}_1^k + \cdots + \bar{y}_m^k,$$

$$\bar{t}_0^k = \frac{1}{2}||\bar{y}_1^k + \cdots + \bar{y}_m^k||^2.$$

Moreover, subproblems $(\text{DAQ})_i, i = 1, \cdots, m$, to be solved at Step 2 become

$$\text{minimize} \quad \frac{1}{2\lambda_k}||z_i - x^k||^2 + (y_i^k)^T z_i \quad \text{subject to} \quad z_i \in C_i.$$

Each of these problems is usually much easier to solve than the original problem (5.1), because it only requires finding the point in the individual convex set C_i, which is closest to the point $x^k - \lambda_k y_i^k$.

The second example is the following separable convex programming problem with linear constraints:

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^n q_j(x_j) \\ \text{subject to} \quad & \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, \cdots, m, \\ & 0 \leq x_j \leq c_j, \quad j = 1, \cdots, n, \end{aligned} \tag{5.2}$$

where $q_j : R \to R$ are strictly convex functions and a_{ij}, b_i, c_j are given constants. Let us define the convex functions $f_{0j} : R \to R \cup \{+\infty\}$ by

$$f_{0j}(x_j) = \begin{cases} q_j(x_j) & \text{if} \quad 0 \le x_j \le c_j \\ +\infty & \text{otherwise} \end{cases}$$

for $j = 1, \cdots, n$, and the convex sets (halfspaces) $H_i \subset R^n$ by

$$H_i = \left\{ x \in R^n \mid \sum_{j=1}^{n} a_{ij} x_j \le b_i \right\}.$$

for $i = 1, \cdots, m$. Then by setting

$$f_0(x) = \sum_{j=1}^{n} f_{0j}(x_j) \quad \text{and} \quad f_i(x) = \delta(x|H_i), \quad i = 1, \cdots, m,$$

we again obtain a problem of the form (Q), to which Algorithm 2 can effectively be applied. In fact, the auxiliary problem to be solved in Step 4 can be split into the following n independent problems:

$$\text{minimize} \quad q_j(x_j) + \bar{y}_{ij}^k x_j \quad \text{subject to} \quad 0 \le x_j \le c_j, \tag{5.3}$$

where $\bar{y}_{ij}^k$ is the jth element of the vector $\bar{y}_i^k \in R^n$. Each of these problems contains an individual variable x_j only, and hence can be solved by using a suitable univariate minimization technique.

On the other hand, subproblems $(\text{DAQ})_i$, $i = 1, \cdots, m$, in Step 2 are written as

$$\text{minimize} \quad \frac{1}{2\lambda_k} ||z_i - x^k||^2 + (y_i^k)^T z_i \ \text{ subject to } \quad a_i^T z_i \le b_i, \tag{5.4}$$

where $a_i = (a_{i1}, \cdots, a_{in})^T$. Then, by straightforward calculation, the unique solution z_i^k of this problem is explicitly represented as

$$z_i^k = x^k - \lambda_k y_i^k - \frac{\max\{0, a_i^T (x^k - \lambda_k y_i^k) - b_i\}}{||a_i||^2} a_i, \tag{5.5}$$

from which the quantities d_i^k and $\bar{t}_i^k$ are obtained immediately. After all, Algorithm 2 seems very attractive for problem (5.2). In particular, we observe that not only subproblems (5.4) in Step 2 but also subproblems (5.3) in Step 4 are decomposed into smaller independent subproblems.

6. Numerical Results

In this section, we present some numerical results for the proposed algorithm (Algorithm 2) on a parallel computer called ADENA [12] at the Department of Applied Mathematics and Physics, Kyoto University. ADENA was originally designed to solve PDE's and consists of 16×16 PE array, i.e., 256

PEs. Its performance is estimated as 2.56 GFLOPS (peak) or 1.0 GFLOPS (effective). Programs used in the numerical experiments were coded in the parallel programming language ADETRAN [12], which is a modification of FORTRAN specially developed for ADENA. All computations were done in double precision arithmetic.

The test problems are randomly generated quadratic programming problems of the following form:

$$\text{minimize} \quad \frac{1}{2}x^T Dx + c^T x \quad \text{subject to} \quad a_i^T x \leq b_i, \quad i = 1, \cdots, m, \tag{6.1}$$

where D is an $n \times n$ positive diagonal matrix, c and $a_i, i = 1, \cdots, m$, are vectors in R^n and $b = (b_1, \cdots, b_m)^T$ is a vector in R^m. The elements of a_i, b, c and D were chosen from the intervals $[-5, 5]$, $[1, 10]$, $[-100, 100]$ and $[1, 10]$, respectively. Note that any problem thus generated is feasible, because $b > 0$ implies that the origin $x = 0$ satisfies the constraints.

Algorithm 2 can be applied to problem (6.1) with the following identifications:

$$\begin{aligned} f_0(x) &= \frac{1}{2}x^T Dx + c^T x \\ f_i(x) &= \begin{cases} 0 & \text{if } a_i^T x \leq b_i, \\ \infty & \text{otherwise,} \end{cases} \quad i = 1, \cdots, m. \end{aligned} \tag{6.2}$$

In this case, each subproblem $(\text{DAQ})_i$ solved in Step 2 is the singly constrained quadratic programming problem

$$\text{minimize} \quad \frac{1}{2\lambda_k}||z_i - x^k||^2 + (y_i^k)^T z_i \quad \text{subject to} \quad a_i^T z_i \leq b_i, \tag{6.3}$$

of which solution can be obtained explicitly (see (5.4) and (5.5)). On the other hand, subproblems to be solved in Step 4 (and Step 1) become the unconstrained quadratic minimization problems

$$\text{minimize} \quad \frac{1}{2}x^T Dx + \left(c - \sum_{i=1}^m \bar{y}_i^k\right)^T x, \tag{6.4}$$

for which explicit solutions can also be obtained componentwise by

$$\bar{x}_j = \frac{\left(c - \sum_{i=1}^m \bar{y}_i^k\right)_j}{D_j}, \quad j = 1, \cdots, n, \tag{6.5}$$

where D_j are the elements of the diagonal matrix D. In our computational experiments, by taking advantage of this fact, computation of (6.5) was executed in parallel as well as the solutions of subproblems (6.3).

To see the behavior of the algorithm, we first examined how the sequence of objective values, say $\{\Phi(y^k)\}$, of the Fenchel dual (cf. problem (D)) of problem (6.1) converged to its optimal value, say Φ^*. To obtain the (approximate) value of Φ^*, we iterated the algorithm sufficiently many times until substantial improvement in the objective value was no longer observed. We solved five test

problems with $n = 100$ and $m = 100$. The results for each run are illustrated in Figure 1, where the vertical and horizontal axes represent the relative error in the objective value $|\Phi(y^k) - \Phi^*|/|\Phi^*|$ and the number of iterations, respectively. The figure indicates that the sequence converges almost linearly.

Next we made a comparison of parallel and serial implementations of the algorithm to appraise the effect of parallel computation. Programs for the serial version were coded in FORTRAN77 and run on a SUN SPARK1 IPX. By some preliminary experiments, we have found that SUN SPARK1 IPX performs floating point arithmetic operations about $1.2 \sim 1.7$ times faster than a single processor element of the parallel computer ADENA. We solved several test problems with various sizes using same termination criterion $|\Phi(y^k) - \Phi^*|/|\Phi^*| < 10^{-5}$ for the both implementations. (For each test problem, the value of Φ^* was obtained beforehand in the same way as mentioned above.) In this experiment, we fixed the number of variables as $n = 200$ and solved five test problems for various values of the number m of constraints.

Table 1. Comparison of serial and parallel implementations of the proposed algorithm[1]

(n, m)	#Iter.	Serial CPU[2]	Parallel CPU[3]	S/P ratio[4]	Ideal ratio[5]
(200, 20)	7.0	0.586	0.070	8.4	15.6
(200, 40)	10.4	1.774	0.104	17.1	30.5
(200, 60)	13.0	3.284	0.131	25.1	44.7
(200, 80)	16.0	5.318	0.163	32.6	58.3
(200, 100)	20.2	8.262	0.208	39.7	71.0
(200, 120)	25.0	12.220	0.262	46.6	83.2
(200, 140)	29.6	16.728	0.314	53.3	95.0
(200, 160)	38.8	24.964	0.418	59.7	106.5

[1] Numbers are the average over five test problems for each problem size.

[2] CPU sec by SUN SPARC1 IPX.

[3] CPU sec by ADENA.

[4] Crude ratio of Serial CPU sec and Parallel CPU sec, without taking into account the difference of the computers. (It is estimated that the performance of SPARC1 IPX is $1.2 \sim 1.7$ times faster than a single processor element of ADENA.)

[5] Ideal ratio indicates the theoretical maximum speedup attainable by using a parallel computer, assuming equal performance of a single processor element for serial and parallel implementations.

The results are summarized in Table 1, where numbers in the columns "#Iter.", "Serial CPU

sec" and "Parallel CPU sec" are the average over five test problems. The column "S/P ratio" shows the ratio of Serial CPU sec and Parallel CPU sec, which does not take into account the difference of the computers used. The last column "Ideal ratio" indicates the theoretical maximum speedup attainable by using a parallel computer, assuming equal performance of a single processor element for serial and parallel implementations. Numbers in this column were obtained by enumerating the number of basic operations $+, -, *, /$, with the assumption that those operations required the same time unit. In view of the above-mentioned difference in the performance of the two computers, we may conclude that the proposed algorithm is particularly suited for parallel computation, at least for the type of test problems (6.1).

Table 2. Comparison with Han and Lou's algorithm[1]

	Han and Lou's algorithm										Proposed algorithm	
	$\rho = 0.2$		$\rho = 0.4$		$\rho = 0.6$		$\rho = 0.8$		$\rho = 1.0$			
(n, m)	CPU sec	#Iter.	CPU sec	#Iter.	CPU sec	#Iter.	CPU sec	#Iter.	CPU sec	#Iter.	CPU sec	#Iter.
(100, 20)	–	–	0.071	16.6	0.107	25.2	0.139	29.0	0.172	33.0	0.075	10.0
(100, 40)	–	–	–	–	0.179	41.8	0.234	54.8	0.287	67.2	0.104	13.8
(100, 60)	–	–	–	–	0.271	62.4	0.351	81.0	0.426	98.4	0.167	21.8
(100, 80)	–	–	–	–	0.324	73.4	0.480	109.0	0.399	90.4	0.244	31.4
(100, 100)	–	–	–	–	–	–	0.480	107.4	0.540	120.8	0.568	72.0

[1] Numbers are the average over five test problems for each problem size. Symbol $(-)$ indicates that the algorithm failed to solve at least one of the five test problems.

Finally we compared the proposed algorithm with the method of Han and Lou [8]. The latter algorithm also solves in parallel two types of subproblems, which are similar to subproblems (6.3) and (6.4) of the proposed algorithm. The principal difference between the two algorithms is that Han and Lou's algorithm uses a fixed parameter ρ, which corresponds to λ_k in the proposed algorithm, while parameter λ_k is controlled at each iteration in the proposed algorithm. Table 2 summarizes the results for several choices of parameter ρ in the algorithm of Han and Lou. The termination criterion was $|\Phi(y^k) - \Phi^*|/|\Phi^*| < 10^{-5}$. The numbers in the table represent the average over five problem instances for each problem size. Note that Han and Lou's algorithm assumes that the function f_0 is strongly convex, which is particularly the case for our test problems by the positive definiteness of D (see (6.2)). Its convergence is guaranteed, however, only when the parameter ρ is set greater than a certain problem-dependent threshold value which is usually difficult to know beforehand. The symbol $(-)$ in the table indicates that the algorithm failed to solve at least one of the five test

problems, probably due to inappropriate choice of parameter ρ. When the algorithm failed, the generated sequence resulted in either oscillation or divergence. Although the use of larger ρ prevents nonconvergence, it then usually causes slow convergence as shown in Table 2. The table also contains the results for the proposed algorithm with the same convergence criterion for the same set of test problems. It can be seen that the proposed algorithm not only succeeded in solving all test problems but also required less computational time than the algorithm of Han and Lou in most cases. This is probably because the automatic adjustment of parameter λ_k effectively accelerates convergence of the proposed algorithm, in spite of some computational overheads.

References

[1] A. Auslender, R. Cominetti and J.-P. Crouzeix, Convex functions with unbounded level sets and applications to duality theory, *SIAM Journal on Control and Optimization*, to appear.

[2] D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall: Englewood Cliffs, N.J., 1989.

[3] J. Eckstein and D.P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293-318, 1992.

[4] R. Fletcher, *Practical Methods of Optimization, Second Edition*, John Wiley: Chichester, 1987.

[5] M. Fukushima and H. Mine, "A generalized proximal point algorithm for certain non-convex minimization problems," *International Journal of Systems Science*, vol. 12, pp. 989-1000, 1981.

[6] M. Fukushima, K. Takazawa, S. Ohsaki and T. Ibaraki, "Successive linearization methods for large-scale nonlinear programming problems," *Japan Journal of Industrial and Applied Mathematics*, vol. 9, pp. 117-132, 1992.

[7] S.-P. Han and G. Lou, "A parallel algorithm for a class of convex programs," *SIAM Journal on Control and Optimization*, vol. 26, pp. 345-355, 1988.

[8] S.-P. Han and G. Lou, "Some parallel decomposition algorithms for convex programming," Technical Report, Department of Mathematics, University of Illinois, Urbana, Illinois, 1987.

[9] K.C. Kiwiel, "A method for minimizing the sum of a convex function and a continuously differentiable function," *Journal of Optimization Theory and Applications*, vol. 48, pp. 437-449, 1986.

[10] H. Mine and M. Fukushima, "A minimization method for the sum of a convex function and a continuously differentiable function," *Journal of Optimization Theory and Applications*, vol. 33, pp. 9-23, 1981.

[11] K. Mouallif, V.H. Nguyen and J.-J. Strodiot, "A perturbed parallel decomposition method for a class of nonsmooth convex minimization problems," *SIAM Journal on Control and Optimization*, vol. 29, pp. 829-847, 1991.

[12] T. Nogi, "Parallel computation on ADENA," in *Parallel Computing '91*, D.J. Evans, G.R. Joubert and H. Liddell, eds., Elsevier, New York: pp. 619-626, 1992.

[13] R.T. Rockafellar, *Convex Analysis*, Princeton University Press: Princeton, N.J., 1970.

[14] R.T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM Journal on Control and Optimization*, vol. 14, pp. 877-898, 1976.

[15] P. Tseng, "Application of a splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM Journal on Control and Optimization*, vol. 29, pp. 119-138, 1991.

[16] S. Zenios, "Parallel numerical optimization: Current status and an annotated bibliography," *ORSA Journal on Computing*, vol. 1, pp. 20-43, 1989.