Nesterov's Smoothing and Excessive Gap Methods for an Optimization Problem in VLSI Placement

Jian-Li Chen · Yan Cui · Wen-Xing Zhu

Abstract In this paper, we propose an algorithm for a nonsmooth convex optimization problem arising in very large-scale integrated circuit placement. The objective function is the sum of a large number of Half-Perimeter Wire Length (HPWL) functions and a strongly convex function. The algorithm is based on Nesterov's smoothing and excessive gap techniques. The main advantage of the algorithm is that it can capture the HPWL information in the process of optimization, and every subproblem has an explicit solution in the process of optimization. The convergence rate of the algorithm is $O(1/k^2)$, where k is the iteration counter, which is optimal. We also present preliminary experiments on nine placement contest benchmarks. Numerical examples confirm the theoretical results.

Keywords VLSI · Global placement · Nonsmooth convex optimization · Smoothing technique · Excessive gap technique

1 Introduction

In 2005, Nesterov presented the smoothing [16] and excessive gap techniques [17] to solve nonsmooth convex minimization problems. Specifically, he solved the following minimization problem,

J.-L. Chen · Y. Cui · W.-X. Zhu (✉)
Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University,
Fuzhou 350108, China
e-mail: wxzhu@fzu.edu.cn

$$\min_{x \in X} f(x) = \widehat{f}(x) + \max_u \{ <Ax, u> \ - \ \widehat{\phi}(u) : u \in Q \}, \tag{1.1}$$

where X is a bounded closed convex set in the n-dimensional Euclidean space $\mathbb{R}^n$; $\widehat{f}(x)$ is a convex and Lipschitz continuous differentiable function with some constant $M \geqslant 0$ on X; the linear operator A maps X to Q; Q is a simple bounded closed convex set in the m-dimensional Euclidean space $\mathbb{R}^m$; and $\widehat{\phi}(u)$ is a simple continuous convex function on Q, such that the maximum function in (1.1) has a closed-form solution. The iteration complexity of Nesterov's smoothing technique is $O(1/\varepsilon)$, where ε is a user-defined absolute accuracy of approximate optimal value. The convergence rate of the excessive gap technique for the problem is $O(1/k)$ if $\widehat{f}(x)$ is a non-strongly convex function, and can reach $O(1/k^2)$ if $\widehat{f}(x)$ is a strongly convex function, where k is the iteration counter. The two methods have been widely used in applications, e.g., sparse recovery [2], resource allocation [8], risk measures for portfolio optimization [7], and multi-commodity flow and fractional packing [19].

In this paper, we aim at using the smoothing [16] and excessive gap techniques [17] to solve a convex optimization problem in the placement of a very large-scale integrated (VLSI) circuit. Placement is one of the most important steps in VLSI computer aided design, since chip performance depends heavily on the circuit placement results [1]. A modern chip often contains millions of circuit cells and nets, which must be placed in a design region legally and the objective is optimized. This has to be done using high-performance optimization techniques.

A VLSI circuit can be modeled as a hypergraph $\mathcal{N} = (V, E)$, where V denotes the set of circuit cells with possibly different widths and heights and E denotes the set of nets specifying interconnections between the circuit cells. Note that a net may contain more than two cells, i.e., a net may be a hyperedge. Given a rectangular $[0, W] \times [0, H]$, VLSI placement seeks a placement of circuit cells such that no cell overlaps with the other, and the total wirelength is minimized [1, 4, 14]. This problem is NP-complete [6], and is rather difficult to solve due to the very large scale and NP-completeness. However, up to now, there are a number of packages for the VLSI placement problem using different optimization methods [1, 4, 14].

Given the location $(x^{(i)}, y^{(i)}) \in [0, W] \times [0, H]$ of circuit cell $i \in V$, the total wirelength of the circuit is defined as the Half-Perimeter Wirelength (HPWL), i.e., $\text{HPWL}_{\mathcal{N}}(\text{x}, \text{y}) = \text{HPWL}_{\mathcal{N}}(\text{x}) + \text{HPWL}_{\mathcal{N}}(\text{y})$, where

$$\text{HPWL}_{\mathcal{N}}(\text{x}) = \sum_{e \in \text{E}} \left[\max_{i \in e} \text{x}^{(i)} - \min_{j \in e} \text{x}^{(j)} \right] = \sum_{e \in \text{E}} \max_{i, j \in e} \left\{ \text{x}^{(i)} - \text{x}^{(j)} \right\}. \tag{1.2}$$

Since the HPWL function (1.2) is continuous and convex, but not differentiable, it is usually approximated by the Bound-2-Bound (B2B) net model [21]. The B2B model is a convex quadratic function and has been widely used in VLSI placement research [1, 4, 13]. SimPL [10, 12], ComPLx [9], and Mapple [11] are among the best state-of-the-art placers in modern VLSI Computer Aided Design. They also approximate the HPWL function by the B2B net model [21], and solve the following problem as a subproblem,

$$\min_{x \in X} f(x) = \mathrm{HPWL}_{\mathcal{N}} \mathrm{B2B}(x) + \lambda \left\| x - x_0^+ \right\|_1, \tag{1.3}$$

where $X = [0, W]^n$, $\mathrm{HPWL}_{\mathcal{N}} \mathrm{B2B}(x)$ is the B2B convex quadratic approximation of the function $\mathrm{HPWL}_{\mathcal{N}}(x)$ in Eq. (1.2), $\lambda > 0$, and x_0^+ is the feasible solution of the VLSI placement problem.

The problem (1.3) is a key subproblem in VLSI placers SimPL, ComPLx, and Mapple, which was solved by the conjugate gradient method. In the problem, the B2B net model is a rough approximation of the HPWL function, which cannot capture the HPWL information exactly in the process of optimization. Moreover, the $\| \cdot \|_1$ norm is not Lipschitz continuously differentiable, which makes the conjugate gradient method for solving the problem (1.3) not theoretically sound.

In this paper, we do not use the B2B net model to approximate the function $\mathrm{HPWL}_{\mathcal{N}}(x)$ but use the HPWL function directly. The l_1-norm $\| \cdot \|_1$ is changed to the square of l_2-norm, i.e., $\| \cdot \|_2^2$. Hence, the problem we consider in this paper is

$$\min_{x \in X} f(x) = \sum_{e \in E} \max_{i,j \in e} \left\{ x^{(i)} - x^{(j)} \right\} + \lambda \left\| x - x_0^+ \right\|_2^2, \tag{1.4}$$

where $\lambda > 0$. The function $\mathrm{HPWL}_{\mathcal{N}}(x)$ is a non-differentiable convex function, so we will adopt Nesterov's smoothing and excessive gap techniques [16, 17] to solve problem (1.4).

Several approaches in the previous literature can be used to solve the minimization problem (1.4). The LSE net model and the L_P-norm net model [1, 4] approximate the function $\mathrm{HPWL}_{\mathcal{N}}(x)$ by the LSE and L_P-norm functions. Subgradient methods [3] can be used to solve the problem (1.4) directly. But it has been recognized in practice that subgradient methods are usually slow and numerically sensitive to the choice of step sizes. Recently, Nesterov [18] proposed a subgradient method to optimize huge-scale problems with sparse subgradients. This method is based on a recursive update of the results of matrix/vector products and the values of symmetric functions. The convergence rate of the algorithm is $O(1/\sqrt{k})$, where k is the iteration counter. Dinh et al. [5] proposed an algorithm which combines Lagrangian decomposition with excessive gap smoothing techniques to solve large-scale separable convex optimization problems. The convergence rate is $O(1/k)$. However, our problem is not separable, so we cannot use their technique directly. Schmidt et al. [20] proposed an algorithm to optimize the sum of a finite number of smooth convex functions by the stochastic average gradient method. The convergence rate is $O(1/k)$.

In this paper, we propose an algorithm basing on the smoothing and excessive gap techniques by Nesterov [16, 17] to solve problem (1.4). Comparing problem (1.4) with the problem (1.1) in Nesterov [16, 17], we can find that our considered problem is a direct generalization of problem (1.1) on the summation of a large number of maximum functions. Moreover, every maximum function in (1.4) contains only the cells in a net, and the number of cells in a net is almost not too big according to the characteristic of the VLSI circuit. Hence, we use Nesterov's smoothing technique on every maximum function directly. The proposed algorithm has a convergence rate $O(1/k^2)$, where k is the iteration counter. According to the

complexity theory for convex optimization by Nemirovski and Yudin [15], the proposed algorithm is optimal.

The paper is organized as follows. In Sect. 2, some notations and basic results are quoted. In Sect. 3.1, we introduce the technique of smoothing the HPWL function of every net e_k. The excessive gap condition (EGC) and some lemmas are given in Sect. 3.2. Section 4 introduces the algorithm. In this section, the theorems of convergence rate and efficiency estimate are also specified. In Sect. 5, we give a scheme to speed up the convergence of the algorithm. Finally, preliminary computational experiments are put in Sect. 6.

2 Notations and Basic Results

In this section, we review some notations and basic results proposed in Nesterov's smoothing and excessive gap techniques [16, 17]. They will be used in our paper.

Let X be a finite dimensional space. In this paper, we define X as

$$X = \left\{ x = (x^{(1)}, x^{(2)}, \cdots, x^{(|V|)})T : 0 \leqslant x^{(i)} \leqslant W \right\},$$

where W is the width of the placement region in the VLSI placement problem.

The space of linear functions on X is denoted by X^*. For $s \in X^*$, $x \in X$, the value of s at x is denoted by $\langle s, x \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the regular inner product. Denote $s, z \in S$, where S is a finite dimensional space equipped with the l_p-norm. The dual norm of s is defined as

$$||s||_p^* = \max_{||z||_p = 1} \langle s, z \rangle.$$

Let A be a linear operator which maps X to Q, i.e., $X \longrightarrow Q$. We use A_i to denote the ith row of A, and use A^j to denote the jth column of A. In this paper, we set the space Q be equipped with the l_1-norm, and $Q = \left\{ u : \sum_{i=1}^p |u^{(i)}| \leqslant 1 \right\}$.

For $x \in X$, $u \in Q$, X is equipped with the l_2-norm, and the norm of A is defined as

$$||A||_{2,1} = \max_{||x||_2 = 1} \max_{||u||_1 = 1} \langle Ax, u \rangle.$$

Clearly,

$$||A||_{2,1} = \left\|A^{\mathrm{T}}\right\|_{1,2} = \max_{||x||_2 = 1} ||Ax||_1^* = \max_{||u||_1 = 1} \left\|A^{\mathrm{T}}u\right\|_2^*.$$

It is easy to verify that

$$||Ax||_1^* \leqslant ||A||_{2,1} \cdot ||x||_2, \quad \forall x \in X, \tag{2.1}$$

and

$$\left\|A^{\mathrm{T}}u\right\|_2^* \leqslant ||A||_{2,1} \cdot ||u||_1, \quad \forall u \in Q. \tag{2.2}$$

It is easy to prove that the norm of A satisfies

$$\|A\|_{2,1} = \max_{\|x\|_2=1} \max_{1\leqslant i\leqslant p} \left\{ \langle A_i^{\mathrm{T}}, x \rangle \right\} = \max_{1\leqslant i\leqslant p} \left\{ \|A_i\|_2^* \right\}, \tag{2.3}$$

where p is the number of rows in A.

Furthermore, we define a strongly convex function $d_Q(u)$ on the convex set Q, which satisfies that

$$d_Q(u) \geqslant d_Q(u^*) + \frac{1}{2}\sigma_Q \|u - u^*\|_1^2, \tag{2.4}$$

and for all $u, v \in Q$,

$$d_Q(u) \geqslant d_Q(v) + \langle \nabla d_Q(v), u - v \rangle + \frac{1}{2}\sigma_Q \|u - v\|_1^2, \tag{2.5}$$

where $\sigma_Q > 0$ is the *strong convexity parameter* of the function $d_Q(u)$, and $u^* = \operatorname{argmin}_{u\in Q} d_Q(u)$. Without loss of generality, we assume that $d_Q(u^*) = 0$.

3 Objective and Dual Functions

In this section, we present the smoothing technique for the VLSI placement problem in detail. And then, the EGC is given for the objective function and its dual form.

3.1 Smoothing the Objective Function

Let $a_{i,j}(e_k)$ be an n-dimensional vector corresponding to net e_k and cells i and j, where n is the dimension of the vector x. And for any $i, j \in e_k$, $i \neq j$, the ith component of $a_{i,j}(e_k)$ is 1, the jth component of $a_{i,j}(e_k)$ is -1, and the other components are zeros. Let $A(e_k)$ be a matrix with respect to e_k with all possible vectors $a_{i,j}(e_k)$ as rows. To make it clear, we give the following example.

Consider a hypergraph $\mathcal{N} = (V, E)$, where $V = \{1, 2, 3\}$, $E = \{e_1, e_2\}$, and the cells $1, 2 \in e_1$, $1, 2, 3 \in e_2$. By the notation, for the net e_1, $a_{1,2}(e_1) = (1, -1, 0)$, $a_{2,1}(e_1) = (-1, 1, 0)$; and for the net e_2, we have $a_{1,2}(e_2) = (1, -1, 0)$, $a_{1,3}(e_2) = (1, 0, -1)$, $a_{2,1}(e_2) = (-1, 1, 0)$, $a_{2,3}(e_2) = (0, 1, -1)$, $a_{3,1}(e_2) = (-1, 0, 1)$, $a_{3,2}(e_2) = (0, -1, 1)$. Thus, the matrices $A(e_1)$ and $A(e_2)$ have the following forms:

$$A(e_1) = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \end{pmatrix}, \quad A(e_2) = \begin{pmatrix} 1 & 1 & -1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 1 & 0 & -1 \\ 0 & -1 & 0 & -1 & 1 & 1 \end{pmatrix}^{\mathrm{T}}.$$

Obviously, the dimension of $A(e_k)$ is $(n_k^2 - n_k) \times n$, where n_k is the number of cells on the net e_k. Moreover, any row of the matrix $A(e_k)$ has only two non-zero components. The one is 1 and the other one is -1. So by (2.3), we can get

$$\|A(e_k)\|_{2,1} = \|A_i(e_k)\|_2^* = \|A_i(e_k)\|_2 = \sqrt{2}. \tag{3.1}$$

Lemma 3.1 *By the above notations, for any net e_k with n_k cells, we have*

$$\mathrm{HPWL}_{e_k}(\mathrm{x}) = \max_{i,j \in e_k}\left\{x^{(i)} - x^{(j)}\right\} = \max_{i,j \in e_k}\left\{\langle a_{i,j}(e_k), x\rangle\right\}$$

$$= \max_{u_k}\left\{\langle A(e_k)x, u_k\rangle: \sum_{i=1}^{n_k(n_k-1)}\left|u_k^{(i)}\right| \leqslant 1\right\}.$$

Proof For cells $i, j \in e_k$, it holds that $x^{(i)} - x^{(j)} = \langle a_{i,j}(e_k), x\rangle$. Thus,

$$\max_{i,j \in e_k}\left\{x^{(i)} - x^{(j)}\right\} = \max_{i,j \in e_k}\langle a_{i,j}(e_k), x\rangle = \max_{i,j \in e_k}\left|x^{(i)} - x^{(j)}\right| = \max_{i,j \in e_k}\left|\langle a_{i,j}(e_k), x\rangle\right|. \tag{3.2}$$

Furthermore, there exist $i', j' \in e_k$ such that

$$\max_{i,j \in e_k}\left|\langle a_{i,j}(e_k), x\rangle\right| = \left|\langle a_{i',j'}(e_k), x\rangle\right|. \tag{3.3}$$

So for u_k such that $\sum_{i=1}^{n_k(n_k-1)}|u_k^{(i)}| \leqslant 1$,

$$\begin{aligned}
\langle A(e_k)x, u_k\rangle &= \sum_{i=1}^{n_k(n_k-1)} u_k^{(i)}\langle A_i(e_k), x\rangle \\
&\leqslant \sum_{i=1}^{n_k(n_k-1)}\left|u_k^{(i)}\langle A_i(e_k), x\rangle\right| \\
&\leqslant \left|\langle a_{i',j'}(e_k), x\rangle\right|\sum_{i=1}^{n_k(n_k-1)}\left|u_k^{(i)}\right| \\
&\leqslant \left|\langle a_{i',j'}(e_k), x\rangle\right|.
\end{aligned} \tag{3.4}$$

Hence, by (3.2)–(3.4),

$$\max_{u_k}\left\{\langle A(e_k)x, u_k\rangle: \sum_{i=1}^{n_k(n_k-1)}\left|u_k^{(i)}\right| \leqslant 1\right\} \leqslant \max_{i,j \in e_k}\langle a_{i,j}(e_k), x\rangle.$$

Furthermore, suppose that in Eq. (3.2), $a_{i',j'}(e_k)$ is the pth row of $A(e_k)$. Take u_k such that the pth component of u_k is $\mathrm{sgn}(\langle a_{i',j'}(e_k), x\rangle)$, and the other components of u_k are 0. Then

$$\langle A(e_k)x, u_k\rangle = \left|\langle a_{i',j'}(e_k), x\rangle\right| = \max_{i,j \in e_k}\langle a_{i,j}(e_k), x\rangle,$$

which implies that

$$\max_{i,j \in e_k}\left\{\langle a_{i,j}(e_k), x\rangle\right\} = \max_{u_k}\left\{\langle A(e_k)x, u_k\rangle: \sum_{i=1}^{n_k(n_k-1)}\left|u_k^{(i)}\right| \leqslant 1\right\}.$$

Hence, Lemma 3.1 holds.

As shown in Lemma 3.1, in this paper, we define the set

$$Q_k = \left\{ u_k \colon \sum_{i=1}^{n_k(n_k-1)} \left| u_k^{(i)} \right| \leqslant 1 \right\},$$

where n_k is the number of cells on the net e_k. Furthermore, by Lemma 3.1, for any net e_k, we write the function $\mathrm{HPWL}_{e_k}(\mathrm{x})$ of e_k as

$$f_{e_k}(x) = \max_{u_k \in Q_k} \langle A(e_k)x, u_k \rangle. \tag{3.5}$$

Let

$$\widehat{f}(x) = \left\| x - x_0^+ \right\|_2^2.$$

By Lemma 3.1, the problem in (1.4) can be written as

$$\min_{x \in X} f(x) = \lambda \widehat{f}(x) + \sum_{k=1}^{m} \max_{u_k \in Q_k} \langle A(e_k)x, u_k \rangle, \tag{3.6}$$

where $\lambda > 0$, $m = |\mathcal{N}|$, which is obviously a direct generalization of problem (1.1).

For every net e_k, the HPWL function (1.2) is convex but not Lipschitz continuously differentiable. Hence, the problem (3.6) is not easy to solve directly. We use the idea of smoothing technique [16] to transform the problem such that it is smooth.

Recall that $d_{Q_k}(u_k)$ is a strongly convex function on Q_k with a strongly convexity parameter $\sigma_{Q_k} > 0$. Let u_k^* be the optimal solution of $d_{Q_k}(u_k)$, i.e.,

$$u_k^* = \arg \min_{u_k \in Q_k} d_{Q_k}(u_k). \tag{3.7}$$

Let μ be a positive smoothness parameter. Consider the following function

$$f_{e_k, \mu}(x) = \max_{u_k \in Q_k} \left\{ \langle A(e_k)x, u_k \rangle - \mu d_{Q_k}(u_k) \right\}. \tag{3.8}$$

Since $d_{Q_k}(u_k)$ is strongly convex, the optimal solution of the above maximization problem is unique and we let it be $u_k^*(x)$. Thus, the smoothness function of the objective function in (3.6) is in the following form,

$$f_\mu(x) = \lambda \widehat{f}(x) + \sum_{k=1}^{m} \max_{u_k \in Q_k} \left\{ \langle A(e_k)x, u_k \rangle - \mu d_{Q_k}(u_k) \right\}. \tag{3.9}$$

By Theorem 1 of [16], $f_\mu(x)$ is a continuously differentiable convex function, and its gradient

$$\nabla f_\mu(x) = \lambda \nabla \widehat{f}(x) + \sum_{k=1}^{m} \langle A(e_k), u_k^*(x) \rangle,$$

is Lipschitz-continuous on X.

The dual problem of (3.6) can be formulated as

$$\max_{u_1 \in Q_1, \cdots, u_m \in Q_m} \Phi(u_1, \cdots, u_m) = \min_{x \in X} \lambda \widehat{f}(x) + \sum_{k=1}^{m} \langle A(e_k)x, u_k \rangle, \tag{3.10}$$

where $\lambda > 0$. Denote $x^*(u)$ as the optimal solution of the minimization problem in the dual problem (3.10), where $u = (u_1, u_2, \cdots, u_m)^{\mathrm{T}}$. Since $\widehat{f}(x)$ is strongly convex with strongly convex parameter 2, $x^*(u)$ is unique. Moreover, by Theorem 1 of [16], $\Phi(u_1, \cdots, u_m)$ is a continuously differentiable concave function.

Therefore, for any $\widehat{x} \in X$ and $\overline{u}_k \in Q_k$, where $k = 1, \cdots, m$, we have

$$\Phi(\overline{u}_1, \cdots, \overline{u}_m) \leqslant \lambda \widehat{f}(\widehat{x}) + \sum_{k=1}^{m} \langle A(e_k)\widehat{x}, \overline{u}_k \rangle. \tag{3.11}$$

Obviously, suppose $\widehat{x} = x^*(\overline{u})$, then we have

$$\Phi(\overline{u}_1, \cdots, \overline{u}_m) \leqslant \lambda \widehat{f}(\widehat{x}) + \sum_{k=1}^{m} \langle A(e_k)\widehat{x}, \overline{u}_k \rangle. \tag{3.12}$$

Since $u_1, \cdots, u_m$ are independent and they have different dimensions, the partial gradient of the function $\Phi(u_1, \cdots, u_m)$ with respect to u_k is denoted by

$$\nabla_{u_k} \Phi(u_1, \cdots, u_m) = \frac{\partial \Phi(u_1, \cdots, u_m)}{\partial u_k} = \frac{\partial \Phi}{\partial x^*(u)} \frac{\partial x^*(u)}{\partial u_k} + A(e_k)x^*(u).$$

Since $x^*(u)$ is the unique optimal solution of (3.10), the gradient of the function $\Phi(u_1, \cdots, u_m)$ with respect to $x^*(u)$ satisfies the following equality:

$$\frac{\partial \Phi(u_1, \cdots, u_m)}{\partial x^*(u)} = 0.$$

Hence,

$$\nabla_{u_k} \Phi(u_1, \cdots, u_m) = \frac{\partial \Phi(u_1, \cdots, u_m)}{\partial u_k} = A(e_k)x^*(u). \tag{3.13}$$

By the way, the gradient of $\Phi(u_1, \cdots, u_m)$ is defined as follows:

$$\nabla \Phi(u_1, \cdots, u_m) = (\nabla_{u_1} \Phi(u_1, \cdots, u_m), \cdots, \nabla_{u_m} \Phi(u_1, \cdots, u_m))^{\mathrm{T}}.$$

By the properties of the norms, we can get the following inequality:

$$\|\nabla \Phi(u_1, \cdots, u_m)\| \leqslant \sum_{k=1}^{m} \|\nabla_{u_k} \Phi(u_1, \cdots, u_m)\|. \tag{3.14}$$

Lemma 3.2 *The function $\Phi(u_1, \cdots, u_m)$ is Lipschitz-continuous differentiable with a Lipschitz constant*

$$L(\Phi) = \frac{m^2}{\lambda}.$$

Proof From the assumption that $\widehat{f}(x)$ is strongly convex, the function $\Phi(u_1, \cdots, u_m)$ has the unique minimal solution $x^*(u)$. Consider $u_k, v_k \in Q_k, k = 1, \cdots, m$. By the first-order optimality condition, we have

$$\langle \lambda \nabla \widehat{f}(x^*(u)) + \sum_{k=1}^{m} A(e_k)^{\mathrm{T}} u_k, \, x^*(v) - x^*(u) \rangle \geqslant 0,$$

$$\langle \lambda \nabla \widehat{f}(x^*(v)) + \sum_{k=1}^{m} A(e_k)^{\mathrm{T}} v_k, \, x^*(u) - x^*(v) \rangle \geqslant 0.$$

Adding the above inequalities and using the strong convexity of $\widehat{f}(x)$, we have

$$\sum_{k=1}^{m} \langle A(e_k)(x^*(v) - x^*(u)), \, u_k - v_k \rangle \geqslant \langle \lambda \nabla \widehat{f}(x^*(u)) - \lambda \nabla \widehat{f}(x^*(v)), \, x^*(u) - x^*(v) \rangle$$

$$\geqslant \lambda \widehat{\sigma} \|x^*(u) - x^*(v)\|_2^2.$$

(3.15)

By (2.1) and (3.15), we can get the following inequalities:

$$\left(\sum_{k=1}^{m} \|A(e_k)(x^*(u) - x^*(v))\|_1^* \right)^2$$

$$(\text{by } (2.1)) \leqslant \left(\sum_{k=1}^{m} \|A(e_k)\|_{2,1} \|(x^*(u) - x^*(v))\|_2 \right)^2$$

$$= \left(\sum_{k=1}^{m} \|A(e_k)\|_{2,1} \right)^2 \|x^*(v) - x^*(u)\|_2^2$$

$$(\text{by } (3.15)) \leqslant \frac{1}{\lambda \widehat{\sigma}} \left(\sum_{k=1}^{m} \|A(e_k)\|_{2,1} \right)^2 \sum_{k=1}^{m} \langle A(e_k)(x^*(u) - x^*(v)), \, u_k - v_k \rangle$$

$$\leqslant \frac{1}{\lambda \widehat{\sigma}} \left(\sum_{k=1}^{m} \|A(e_k)\|_{2,1} \right)^2 \sum_{k=1}^{m} \|A(e_k)(x^*(u) - x^*(v))\|_1^* \|u_k - v_k\|_1$$

$$\leqslant \frac{1}{\lambda \widehat{\sigma}} \left(\sum_{k=1}^{m} \|A(e_k)\|_{2,1} \right)^2 \sum_{k=1}^{m} \|A(e_k)(x^*(u) - x^*(v))\|_1^* \sum_{k=1}^{m} \|u_k - v_k\|_1.$$

Thus, by (3.1) and $\widehat{\sigma} = 2$, we can get the following inequality:

$$\sum_{k=1}^{m} \|A(e_k)(x^*(u) - x^*(v))\|_1^* \leqslant \frac{1}{\lambda \hat{\sigma}} \left(\sum_{k=1}^{m} \|A(e_k)\|_{2,1} \right)^2 \sum_{k=1}^{m} \|u_k - v_k\|_1$$

$$= \frac{m^2}{\lambda} \sum_{k=1}^{m} \|u_k - v_k\|_1.$$

Moreover, by (3.13), (3.14), and the above inequality, we have

$$\|\nabla \Phi(u_1, \cdots, u_m) - \nabla \Phi(v_1, \cdots, v_m)\|_1^* \leqslant \sum_{k=1}^{m} \|\nabla_{u_k} \Phi_{e_k}(u_1, \cdots, u_m) - \nabla_{v_k} \Phi(v_1, \cdots, v_m)\|_1^*$$

$$= \sum_{k=1}^{m} \|A(e_k)(x^*(u) - x^*(v))\|_1^*$$

$$\leqslant \frac{m^2}{\lambda} \sum_{k=1}^{m} \|u_k - v_k\|_1.$$

Hence, the lemma holds.

3.2 Excessive Gap Condition

Similar to [17], for some $\overline{x} \in X$ and $\overline{u}_k \in Q_k$, where $k = 1, \cdots, m$, the EGC is given as follows:

$$f_\mu(\overline{x}) \leqslant \Phi(\overline{u}_1, \cdots, \overline{u}_m). \tag{3.16}$$

Lemma 3.3 *Let vectors $\overline{x} \in X$ and $\overline{u}_k \in Q_k$ satisfying (3.16), where $k = 1, \cdots, m$. Then*

$$0 \leqslant f(\overline{x}) - \Phi(\overline{u}_1, \cdots, \overline{u}_m) \leqslant \mu \sum_{k=1}^{m} D_k, \tag{3.17}$$

where $D_k = \max_{u_k \in Q_k} d_{Q_k}(u_k)$.

Proof Clearly,

$$\lambda \widehat{f}(\overline{x}) + \sum_{k=1}^{m} \{ \langle A(e_k)\overline{x}, u_k \rangle - \mu D_k \}$$

$$\leqslant \lambda \widehat{f}(\overline{x}) + \sum_{k=1}^{m} \{ \langle A(e_k)\overline{x}, u_k \rangle - \mu d_{Q_k}(u_k) \}.$$

So we have

$$f(\overline{x}) - \mu \sum_{k=1}^{m} D_k = \lambda \widehat{f}(\overline{x}) + \sum_{k=1}^{m} \max_{u_k \in Q_k} \{ \langle A(e_k)\overline{x}, u_k \rangle - \mu D_k \}$$

$$\leqslant \lambda \widehat{f}(\overline{x}) + \sum_{k=1}^{m} \max_{u_k \in Q_k} \{ \langle A(e_k)\overline{x}, u_k \rangle - \mu d_{Q_k}(u_k) \} = f_\mu(\overline{x}).$$

Hence, we can easily get

$$f(\bar{x}) - \mu \sum_{k=1}^{m} D_k \leqslant f_\mu(\bar{x}) \leqslant \Phi(\bar{u}_1, \cdots, \bar{u}_m),$$

and Lemma 3.3 holds.

For $u_k, v_k \in Q_k$, denote the *Bregman distance* between u_k and v_k by

$$\xi(v_k, u_k) = d_{Q_k}(v_k) - d_{Q_k}(u_k) - \langle \nabla d_{Q_k}(u_k), v_k - u_k \rangle. \qquad (3.18)$$

By (2.5), we have

$$\xi(v_k, u_k) \geqslant \frac{1}{2} \sigma_{Q_k} \|v_k - u_k\|_1^2. \qquad (3.19)$$

Define the *Bregman projection* of g on the set Q_k as follows:

$$V_k(u_k, g) = \underset{v_k \in Q_k}{\arg\max} \{\langle g, v_k - u_k \rangle - \mu \xi(v_k, u_k)\}. \qquad (3.20)$$

Lemma 3.4 *The EGC holds for*

$$\mu = \max_{1 \leqslant k \leqslant m} \{\frac{1}{\sigma_{Q_k}} L(\Phi)\},$$

$$\bar{x} = x^*(u^*),$$

$$\bar{u}_k = V_k\big(u_k, \nabla_{u_k} \Phi\big(u_1^*, \cdots, u_m^*\big)\big),$$

where u_k^* is the minimal solution of $d_{Q_k}(u_k)$ and $k = 1, \cdots, m$.

Proof Indeed, for any $x \in X$ and $u_k \in Q_k$, $k = 1, \cdots, m$, by setting $g_k = \nabla_{u_k} \Phi(u_1^*, \cdots, u_m^*)$, we can get the equation

$$\Phi\big(V_1\big(u_1^*, g_1\big), \cdots, V_m\big(u_m^*, g_m\big)\big) = \Phi(\bar{u}_1, \cdots, \bar{u}_m).$$

By Lemma 3.3, the function $\Phi(u_1, \cdots, u_m)$ is Lipschitz-continuous differentiable, we have

$$\|\nabla\Phi(u_1, \cdots, u_m) - \nabla\Phi(v_1, \cdots, v_m)\|_1^* \leqslant L(\Phi) \sum_{k=1}^{m} \|u_k - v_k\|_1.$$

In review of [12] (Sect. 2.1), from the fact that the function $\Phi(u_1, \cdots, u_m)$ is concave, we have

$$-\Phi(\bar{u}_1, \cdots, \bar{u}_m)$$
$$\leqslant -\Phi\big(u_1^*, \cdots, u_m^*\big) - \sum_{k=1}^{m} \langle \nabla_{u_k} \Phi, \bar{u}_k - u_k^* \rangle + \frac{1}{2} L(\Phi) \sum_{k=1}^{m} \|\bar{u}_k - u_k^*\|_1^2.$$

Thus,

$$\Phi(\overline{u}_1, \cdots, \overline{u}_m) \geqslant \Phi(u_1^*, \cdots, u_m^*)$$
$$+ \sum_{k=1}^{m} \left\{ \langle \nabla_{u_k} \Phi, \overline{u}_k - u_k^* \rangle - \frac{1}{2} L(\Phi) \left\| \overline{u}_k - u_k^* \right\|_1^2 \right\}$$
$$= \lambda \widehat{f}(x^*(u^*)) + \sum_{k=1}^{m} \left\{ \langle A(e_k)x^*(u^*), u_k^* \rangle \right.$$
$$+ \langle A(e_k)x^*(u^*), \overline{u}_k - u_k^* \rangle - \frac{1}{2} L(\Phi) \left\| \overline{u}_k - u_k^* \right\|_1^2 \right\}$$
$$(\text{by } (3.19)) \geqslant \lambda \widehat{f}(x^*(u^*)) + \sum_{k=1}^{m} \left\{ \langle A(e_k)x^*(u^*), u_k^* \rangle \right.$$
$$+ \langle A(e_k)x^*(u^*), \overline{u}_k - u_k^* \rangle - \frac{L(\Phi)}{\sigma_{Q_k}} \xi(\overline{u}_k, u_k^*) \right\}$$
$$\left(\text{by } \mu = \max_{1 \leqslant k \leqslant m} \left\{ \frac{1}{\sigma_{Q_k}} L(\Phi) \right\} \right) \geqslant \lambda \widehat{f}(x^*(u^*)) + \sum_{k=1}^{m} \left\{ \langle A(e_k)x^*(u^*), u_k^* \rangle \right.$$
$$+ \langle A(e_k)x^*(u^*), \overline{u}_k - u_k^* \rangle - \mu \xi(\overline{u}_k, u_k^*) \right\}$$
$$= \lambda \widehat{f}(x^*(u^*)) + \sum_{k=1}^{m} \max_{v_k \in Q_k} \left\{ \langle A(e_k)x^*(u^*), v_k \rangle \right.$$
$$- \mu \xi(v_k, u_k^*) \right\}$$
$$(\text{by } (2.5)) \geqslant \lambda \widehat{f}(x^*(u^*))$$
$$+ \sum_{k=1}^{m} \left\{ \max_{v_k \in Q_k} \left\{ \langle A(e_k)x^*(u^*), v_k \rangle - \mu d_{Q_k}(v_k) \right\} \right\}$$
$$= f_\mu(x^*(u^*)).$$

Hence, the EGC (3.16) is satisfied.

4 Algorithm

From Lemma 3.4, we know that the EGC holds for $\mu = \max_{1 \leqslant k \leqslant m} \frac{1}{\sigma_{Q_k}} L(\Phi)$, $\overline{x} = x^*(u^*)$ and $\overline{u}_k = u_k^*$, $k = 1, \cdots, m$. The following theorem develops a scheme to choose μ, $\overline{x}$, $\overline{u} = (\overline{u}_1, \cdots, \overline{u}_m)$, and makes sure that the EGC (3.16) holds in each iteration.

Theorem 4.1 *Let vectors $\overline{x} \in X$ and $\overline{u}_k \in Q_k$, $k = 1, \cdots, m$, satisfying EGC for some positive parameters μ. Fix a parameter $\tau \in (0, 1)$ and choose $\mu_+ = (1 - \tau)\mu$. Let*

$$\widehat{u}_k = (1 - \tau)\overline{u}_k + \tau u_k^*(\overline{x}), \quad k = 1, \cdots, m,$$
$$\overline{x}_+ = (1 - \tau)\overline{x} + \tau x^*(\widehat{u}),$$
$$\widetilde{u}_k = V_k(u_k^*(\overline{x}), \quad \frac{\tau}{(1 - \tau)} A(e_k)\widehat{x}), \quad k = 1, \cdots, m, \qquad (4.1)$$
$$\overline{u}_{k+} = (1 - \tau)\overline{u}_k + \tau \widetilde{u}_k, \quad k = 1, \cdots, m.$$

Then $\bar{x}_+$ and $\bar{u}_+ = (\bar{u}_{1+}, \cdots, \bar{u}_{m+})$ *satisfy EGC with smoothness parameters* μ_+, *provided that* τ *is chosen in accordance with the following relation*:

$$\frac{\tau^2}{1-\tau} \leqslant \min_{1\leqslant k\leqslant m} \frac{\mu\sigma_{Q_k}}{L(\Phi)}.$$

Proof   Denote $\hat{x} = x^*(\hat{u})$ and $v_k = u_k^*(\bar{x})$. By line 2 of (4.1) and the convexity of $\hat{f}(x)$, we have

$$f_{\mu_+}(\bar{x}_+) = \lambda\hat{f}(\bar{x}_+) + \sum_{k=1}^m \max_{u_k\in Q_k}\left\{\langle A(e_k)\bar{x}_+, u_k\rangle - \mu_+ d_{Q_k}(u_k)\right\}$$

$$= \lambda\hat{f}((1-\tau)\bar{x} + \tau\hat{x})$$

$$+ \sum_{k=1}^m \max_{u_k\in Q_k}\left\{\langle A(e_k)[(1-\tau)\bar{x} + \tau\hat{x}], u_k\rangle - (1-\tau)\mu d_{Q_k}(u_k)\right\}$$

$$\leqslant \lambda[(1-\tau)\hat{f}(\bar{x}) + \tau\hat{f}(\hat{x})]$$

$$+ \sum_{k=1}^m \max_{u_k\in Q_k}\left\{(1-\tau)[\langle A(e_k)\bar{x}, u_k\rangle - \mu d_{Q_k}(u_k)] + \tau\langle A(e_k)\hat{x}, u_k\rangle\right\}$$

$$= (1-\tau)(\lambda\hat{f}(\bar{x})) + \tau\Phi(\hat{u}_1, \cdots, \hat{u}_m)$$

$$+ \sum_{k=1}^m \max_{u_k\in Q_k}\left\{(1-\tau)[\langle A(e_k)\bar{x}, u_k\rangle - \mu d_{Q_k}(u_k)] + \tau\langle A(e_k)\hat{x}, u_k - \hat{u}_k\rangle\right\}.$$

Since $v_k = u_k^*(\bar{x})$ is an optimal solution of (3.8), by the first-order condition, we have

$$\sum_{k=1}^m \langle A(e_k)\bar{x} - \nabla d_{Q_k}(v_k), u_k - v_k\rangle \leqslant 0. \tag{4.2}$$

Hence,

$$\lambda\hat{f}(\bar{x}) + \sum_{k=1}^m [\langle A(e_k)\bar{x}, u_k\rangle - \mu d_{Q_k}(u_k)]$$

$$\text{(by (3.18))} = \lambda\hat{f}(\bar{x}) + \sum_{k=1}^m [\langle A(e_k)\bar{x}, u_k\rangle - \mu(\xi(u_k, v_k) + d_{Q_k}(v_k)$$

$$+ \langle \nabla d_{Q_k}(v_k), u_k - v_k\rangle)]$$

$$\text{(by (4.2))} \leqslant \lambda\hat{f}(\bar{x}) + \sum_{k=1}^m [\langle A(e_k)\bar{x}, v_k\rangle - \mu(\xi(u_k, v_k) + d_{Q_k}(v_k))]$$

$$= f_\mu(\bar{x}) - \mu\sum_{k=1}^m \xi(u_k, v_k)$$

$$\text{(by (3.16))} \leqslant \Phi(\bar{u}_1, \cdots, \bar{u}_m) - \mu\sum_{k=1}^m \xi(u_k, v_k)$$

$$\text{(by (3.11))} \leqslant \lambda \widehat{f}(\widehat{x}) + \sum_{k=1}^{m} \langle A(e_k)\widehat{x}, \overline{u}_k \rangle - \mu \sum_{k=1}^{m} \xi(u_k, v_k)$$

$$= \lambda \widehat{f}(\widehat{x}) + \sum_{k=1}^{m} \{ \langle A(e_k)\widehat{x}, \overline{u}_k - \widehat{u}_k + \widehat{u}_k \rangle - \mu \xi(u_k, v_k) \}$$

$$\text{(by (3.12))} = \Phi(\widehat{u}_1, \cdots, \widehat{u}_m) + \sum_{k=1}^{m} \{ \langle A(e_k)\widehat{x}, \overline{u}_k - \widehat{u}_k \rangle \} - \mu \sum_{k=1}^{m} \xi(u_k, v_k).$$

By the above inequalities, (3.19) and (4.1), we can get the following relations:

$$f_{\mu_+}(\overline{x}_+) \leqslant (1 - \tau)(\lambda \widehat{f}(\overline{x})) + \tau \Phi(\widehat{u}_1, \cdots, \widehat{u}_m)$$

$$+ \sum_{k=1}^{m} \max_{u_k \in Q_k} \{ (1 - \tau)[\langle A(e_k)\overline{x}, u_k \rangle - \mu d_{Q_k}(u_k)]$$

$$+ \tau \langle A(e_k)\widehat{x}, u_k - \widehat{u}_k \rangle \}$$

$$\leqslant \Phi(\widehat{u}_1, \cdots, \widehat{u}_m) + \sum_{k=1}^{m} \max_{u_k \in Q_k} \{ A(e_k)\widehat{x}, (1 - \tau)\overline{u}_k + \tau u_k - \widehat{u}_k \rangle$$

$$- (1 - \tau)\mu \xi(u_k, v_k) \}$$

$$\text{(by line 1 of (4.1))} = \Phi(\widehat{u}_1, \cdots, \widehat{u}_m) + (1 - \tau)$$

$$\sum_{k=1}^{m} \max_{u_k \in Q_k} \left\{ \frac{\tau}{1 - \tau} \langle A(e_k)\widehat{x}, u_k - v_k \rangle - \mu \xi(u_k, v_k) \right\}$$

$$\text{(by line 3 of (4.1))} = \Phi(\widehat{u}_1, \cdots, \widehat{u}_m)$$

$$+ (1 - \tau) \sum_{k=1}^{m} \left\{ \frac{\tau}{1 - \tau} \langle A(e_k)\widehat{x}, \widetilde{u}_k - v_k \rangle - \mu \xi(\widetilde{u}_k, v_k) \right\}$$

$$\text{(by (3.19))} \leqslant \Phi(\widehat{u}_1, \cdots, \widehat{u}_m) + (1 - \tau)$$

$$\sum_{k=1}^{m} \left\{ \frac{\tau}{1 - \tau} \langle A(e_k)\widehat{x}, \widetilde{u}_k - v_k \rangle - \frac{\sigma_{Q_k}}{2} \|\widetilde{u}_k - v_k\|_1^2 \right\}$$

$$\leqslant \Phi(\widehat{u}_1, \cdots, \widehat{u}_m)$$

$$+ \sum_{k=1}^{m} \left\{ \tau \langle A(e_k)\widehat{x}, \widetilde{u}_k - v_k \rangle - \frac{\tau^2}{2} L(\Phi) \|\widetilde{u}_k - v_k\|_1^2 \right\}.$$

Moreover, according to lines 1 and 4 of (4.1), we have

$$f_{\mu_+}(\overline{x}_+) \leqslant \Phi(\widehat{u}_1, \cdots, \widehat{u}_m) + \sum_{k=1}^{m} \left\{ \tau \langle A(e_k)\widehat{x}, \widetilde{u}_k - v_k \rangle - \frac{\tau^2}{2} L(\Phi) \|\widetilde{u}_k - v_k\|_1^2 \right\}$$

$$= \Phi(\widehat{u}_1, \cdots, \widehat{u}_m) + \sum_{k=1}^{m} \left\{ \langle A(e_k)\widehat{x}, \overline{u}_{k+} - \widehat{u}_k \rangle - \frac{1}{2} L_2(\Phi) \|\overline{u}_{k+} - \widehat{u}_k\|_1^2 \right\}$$

$$\leqslant \Phi(\overline{u}_{1+}, \cdots, \overline{u}_{m+}).$$

The proof is completed.

According to the above theorem and lemmas, the following algorithm is given. Denote u^i as the ith iteration value of u, x_i as the ith iteration value of x, and u_k as the kth component of u.

Algorithm 1

Input: $u^* = (u_1^*, \cdots, u_m^*)^T$, x_0^+, desired accuracy ε and the set of nets $\mathcal{N}$.

Initialization: set $i = 0$, $\mu_0 = \frac{1}{\sigma} L_2(\phi_{e_k})$, $\overline{x}_0 = x^*(u^*)$ and $\overline{u}_{k,0} = V_k(u_k^*, A(e_k)\overline{x}_0)$.

Iterations:

1: **repeat**

2:     $\tau_i = \frac{2}{i+3}$;

3:     $\widehat{u}^i = (1 - \tau_i)\overline{u}^i + \tau_i u^{i*}(\overline{x}_i)$;

4:     $\overline{x}_{i+1} = (1 - \tau_i)\overline{x}_i + \tau_i x^*(\widehat{u})$;

5:     **for** $(k = 1; k \leqslant m; k + +)$ **do**

6:        $\widetilde{u}_k^i = V_k(u_k^{i*}(\overline{x}_i), \frac{\tau_i}{(1-\tau_i)} A(e_k)x^*(\widehat{u}))$;

7:     **end for**

8:     $\overline{u}^{i+1} = (1 - \tau_i)\overline{u}^i + \tau_i \widetilde{u}^i$;

9:     $\mu_{i+1} = (1 - \tau_i)\mu_i$;

10:    $i = i + 1$;

11: **until** $f(\overline{x}_i) - \Phi(\overline{u}_1^i, \cdots, \overline{u}_m^i) \leqslant \varepsilon$.

Theorem 4.2 *Let the pairs $\overline{x}_i$ and $\overline{u}^i = \{\overline{u}_1^i, \cdots, \overline{u}_m^i\}$ be generated by the above algorithm. Then the following inequality holds*:

$$f(\overline{x}_i) - \Phi(\overline{u}_1^i, \cdots, \overline{u}_m^i) \leqslant \sum_{k=1}^{m} \{\mu_i D_k\} = \sum_{k=1}^{m} \left\{ \frac{2L(\Phi)D_k}{(i+1)(i+2)\sigma} \right\}.$$

Proof According to Theorem 4.1 and Lemma 3.4, we have that the sequences $\{\mu_1^i, \cdots, \mu_m^i\}_{i=0}^{\infty}$ and $\{\tau_i\}_{i=0}^{\infty}$ satisfy the EGC (3.16), and

$$\mu_i = \frac{1}{\sigma} L(\Phi) \frac{1}{3} \times \frac{2}{4} \times \frac{5-2}{5} \times \cdots \times \frac{i}{i+2} = \frac{2L(\Phi)}{(i+1)(i+2)\sigma}.$$

From Lemma 3.3, we have

$$0 \leqslant f(\overline{x}_i) - \Phi(\overline{u}_1^i, \cdots, \overline{u}_m^i) \leqslant \sum_{k=1}^{m} \{\mu_i D_k\} = \sum_{k=1}^{m} \left\{ \frac{2L(\Phi)D_k}{(i+1)(i+2)\sigma} \right\}.$$

This completes the proof.

By Theorem 4.2, we can see that the convergence rate of our algorithm is $O(1/i^2)$, where i is the iteration counter.

Next we will introduce some implementation details of the algorithm. Choose the smoothness function $d_{Q_k}(u_k)$ as the following form:

$$d_{Q_k}(u_k) = \ln n_k(n_k - 1) + \sum_{i=1}^{n_k^2 - n_k} u_k^{(i)} \ln u_k^{(i)}.$$

It is easy to verify that

$$D_k = \max_{u_k \in Q_k} \{d_{Q_k}(u_k)\} = \ln n_k(n_k - 1),$$

and the strong convexity parameter $\sigma_{Q_k} = 1$.

We also need to compute the following objects at each iteration.

(1) *Computation of $u_k^*(\overline{x})$, $k = 1, \cdots, m$. $u_k^*(\overline{x})$ is the* solution of the following problem:

$$\max_{u_k \in Q_k} \left\{ \langle A(e_k)\overline{x}, u_k \rangle - \mu d_{Q_k}(u_k) \right\}.$$

The solution of the above problem is

$$u_k^{(i)*}(\overline{x}) = \frac{\exp \left(\langle A_i(e_k)^{\mathrm{T}}, \overline{x} \rangle / \mu \right)}{\sum_{i=1}^{n_k(n_k - 1)} \exp \left(\langle A_i(e_k)^{\mathrm{T}}, \overline{x} \rangle / \mu \right)}.$$

(2) *Computation of $x^*(u)$. $x^*(u)$ is the* solution of the problem as follows:

$$\min_{x \in X} \lambda \widehat{f}(x) + \sum_{k=1}^{m} \{ \langle A(e_k)x, u_k \rangle \}.$$

The solution of the above problem can be written as

$$x^*(u) = x_0^+ - \frac{1}{2\lambda} \sum_{k=1}^{m} \langle A(e_k), u_k \rangle.$$

(3) *Computation of $V_k(u_k, g)$, $k = 1, \cdots, m$. $V_k(u_k, g)$ is the* solution of the following problem:

$$\max_{v_k \in Q_k} \{ \langle g, v_k - u_k \rangle - \xi(v_k, u_k) \}.$$

The solution of the above problem can be written as

$$V_k^{(i)}(u_k, g) = \frac{u_k^{(i)} \exp \left(g^{(i)} / \mu \right)}{\sum_{i=1}^{n_k(n_k - 1)} u_k^{(i)} \exp \left(g^{(i)} / \mu \right)}.$$

Thus, all variables of the algorithm can be computed directly.

5 Speeding Up the Convergence

By the above lemmas and theorems, we have found that the *Lipschitz* constant $L(\Phi)$ may be too large. So we want to take the following strategies to reduce the constant and speed up the convergence.

For any $x \in X$, denote x_{e_k} as an n-dimensional vector, and for any $i \in e_k$, the ith component of x_{e_k} is $x^{(i)}$, and the other components are 0.

Obviously, for every net e_k, x_{e_k} can be generated by the information of e_k, and we have the equalities as follows:

$$
\begin{aligned}
\langle a_{i,j}(e_k), x \rangle &= \langle a_{i,j}(e_k), x_{e_k} \rangle, \\
\langle A(e_k)x, u_k \rangle &= \langle A(e_k)x_{e_k}, u_k \rangle.
\end{aligned}
\tag{5.1}
$$

Let $\mathcal{A}$ be a matrix corresponding to a net with all the n cells. Then it is obvious that

$$
\|\mathcal{A}x_{e_k}\|_2 \geqslant \|A(e_k)x_{e_k}\|_2.
$$

By denoting $\mathcal{X}^*(u) = (x_1^*(u), x_2^*(u), \cdots, x_m^*(u))$, and $g_k(u) = A(e_k)x^*(u)$, we can get

$$
\|\nabla\Phi(u_1, u_2, \cdots, u_m)\|_1 = \|(g_1(u), g_2(u), \cdots, g_m(u))T\|_1 \leqslant \|\mathcal{A}\mathcal{X}^*(u)\|_1. \tag{5.2}
$$

Denote $\deg_i$ as the degree of the vertex i in the hypergraph. The following relations hold:

$$
\|\bar{x}^*(u)\|_2^2 = \sum_{i=1}^m \|x_{e_k}\|_2^2 = \sum_{i=1}^n \deg_i \|x^{(i)}\|_2^2. \tag{5.3}
$$

Note that the function $\Phi(u_1, \cdots, u_m)$ defined by (3.10) is concave and differentiable, and $x^*(u)$ is its minimal solution. Moreover, the function $\widehat{f}(x)$ is strongly convex with strong convexity parameter $\widehat{\sigma} = 2$. Thus, we have the following lemma.

Lemma 5.1 *The function $\Phi(u_1, \cdots, u_m)$ is Lipschitz-continuous differentiable with a constant*

$$
L(\Phi) = \frac{1}{\lambda} \max_{1 \leqslant i \leqslant m} \deg_i.
$$

Proof From the function $\Phi(u_1, \cdots, u_m)$ defined by (3.10) and the fact that $\widehat{f}(x)$ is strongly convex, the minimization problem in (3.10) has the unique solution $x^*(u)$. Consider $u_k, v_k \in Q_k, k = 1, \cdots, m$. We have

$$
\left(\|\nabla\Phi(u_1, u_2, \cdots, u_m) - \nabla\Phi(v_1, v_2, \cdots, v_m)\|_1^* \right)^2
$$

$$
(\text{by } (5.2)) \leqslant \left(\|\mathcal{A}\mathcal{X}^*(u) - \mathcal{A}\mathcal{X}^*(v)\|_1^* \right)^2
$$

$$
(\text{by } (2.1)) \leqslant \|\mathcal{A}\|_{2,1}^2 \|\mathcal{X}^*(u) - \mathcal{X}^*(v)\|_2^2
$$

$$(\text{by } (3.1)) = 2 \sum_{k=1}^{m} \left\| x_{e_k}^*(u) - x_{e_k}^*(v) \right\|_2^2$$

$$(\text{by } (5.3)) \leqslant 2 \max_{1 \leqslant i \leqslant n} \deg_i \| x^*(u) - x^*(v) \|_2^2$$

$$(\text{by } (3.15)) \leqslant \frac{2}{\lambda \widehat{\sigma}} \max_{1 \leqslant i \leqslant n} \deg_i \sum_{k=1}^{m} \langle A(e_k)(x^*(v) - x^*(u)), u_k - v_k \rangle$$

$$\leqslant \frac{1}{\lambda} \max_{1 \leqslant i \leqslant n} \deg_i \| \nabla \Phi(u_1, u_2, \cdots, u_m) - \nabla \Phi(v_1, v_2, \cdots, v_m) \|_1^* \sum_{k=1}^{m} \| u_k - v_k \|_1.$$

Hence, we have

$$\| \nabla \Phi(u_1, \cdots, u_m) - \nabla \Phi(v_1, \cdots, v_m) \|_1 \leqslant \frac{1}{\lambda} \max_{1 \leqslant i \leqslant n} \deg_i \sum_{k=1}^{m} \| u_k - v_k \|_1,$$

and the lemma holds.

6 Experiments

In this section, we test the algorithm on nine benchmarks of the 2004 International Symposium on Physical Design (ISPD04) placement contest benchmark suites. Our implementation is written in *Matlab* 7.0, and is run on a personal computer with Intel Core2 Duo CPU E7500 (2.9 GHZ) and 2 GB internal memory, under Windows XP. The information of the nine benchmarks are put in Table 1. In the experiments, we take different values $\lambda = 0.1$, 0.5, and 1, and we take the value $\varepsilon = 200$ for every benchmark.

In Fig. 1, we plot the primal function value and the dual function value of the algorithm on the benchmark ibm01 with $\lambda = 1.0$ at each iteration.

From Fig. 1, we can find that the objective function value decreases very fast in the first 10 iterations, and then the value changes very small. Hence, we terminate

Table 1 ICCAD'04 IBM benchmark

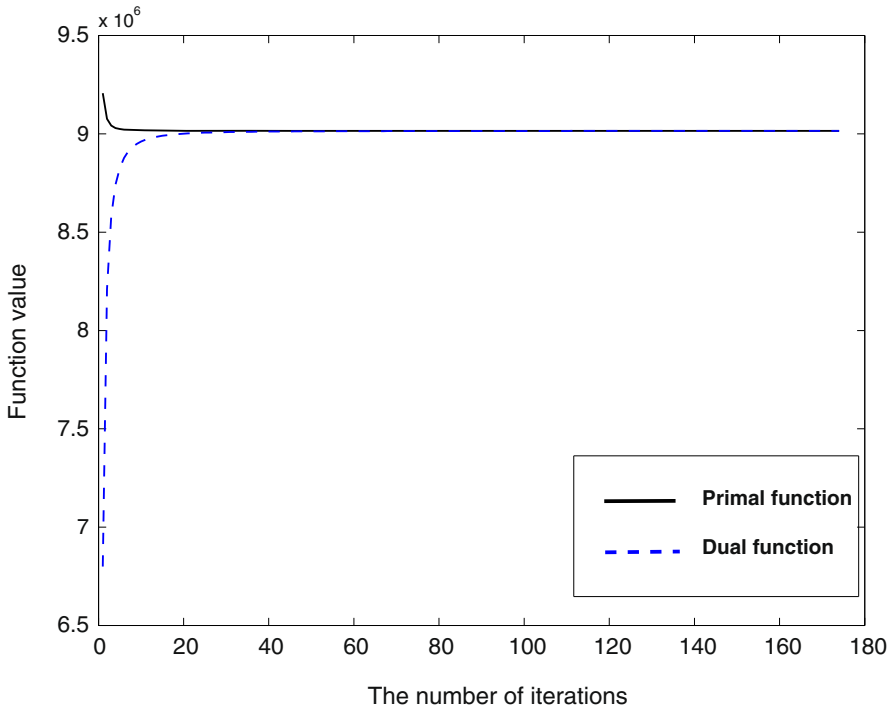| Ckt	#Cells	#Pads	#Nets	#Pins	#Rows
ibm01	12 506	246	14 111	50 566	96
ibm02	19 342	259	19 584	81 199	109
ibm03	22 853	283	27 401	93 573	121
ibm04	27 220	287	31 970	105 859	136
ibm05	28 146	1 201	28 446	126 308	139
ibm06	32 332	166	34 826	128 182	126
ibm07	45 639	287	48 117	175 639	166
ibm08	51 023	286	50 513	204 890	170
ibm09	53 110	285	60 902	222 088	183

Fig. 1 Primal and dual function values of ibm01, $\lambda = 1.0$

our algorithm in the experiments when the dual gap is less than 200. The results of the algorithm on the nine benchmarks are put in Table 2.

Table 2 shows the runtime and the final value of $f(x)$ for different λ and benchmark. In general, the computation of $u_k^*(x)$ takes 25 % of the total runtime, the computation of $V_k(u_k, g)$ takes 27 % the computation of $x^*(u)$ takes 14 % the computation of dual function $\Phi(u_1, \cdots, u_m)$ spends 26 % the computation of primal function $f(x)$ takes 5 % and the others take 3 % of the total runtime, respectively.

The runtime of our algorithm is not only related to the scale of the benchmark, but also related to λ and the number of cells on the nets. From Fig. 1 and Table 2, we can find that with the increase of the value of λ, the algorithm costs less time. This is because L is inversely proportional to λ, and with the increase of L, the improvement of each iteration will be smaller. From the experiments, we also find that computations of $u_k^*(x)$, $V_k(u_k, g)$, and $\Phi(u_1, \cdots, u_m)$ take most of the runtime, and these computations depend on the number of cells on the nets. When the number of cells on a net is larger, the more runtime of these computations will cost.

Finally, it must be remarked that the problem considered in this paper is only a subproblem in the software packages SimPL [10, 12], ComPLx [9], and Mapple [11], for VLSI placement. Our future research will improve the performance of the algorithm, and implement it for real VLSI placement.

Table 2 Runtime and primal and dual function values

λ	1			0.5			0.1		
	f	Φ	CPU	f	Φ	CPU	f	Φ	CPU
ibm01	9 187 923.85	9 187 726.63	251.59	9 168 560.15	9 168 219.08	328.03	901 528.66	901 436.66	721.73
ibm02	13 371 112.07	13 370 917.72	863.03	13 344 941.81	13 344 122.67	1 189.56	13 140 627.68	13 139 423.49	2 554.43
ibm03	7 220 386.37	17 220 186.88	687.53	17 175 462.44	17 174 569.97	930.34	16 837 384.89	16 834 217.65	2 129.6
ibm04	20 216 985.81	20 214 125.84	1 217.16	20 159 463.89	20 155 820.2	1 731.17	19 753 781.35	19 748 371.43	4 129.6
ibm05	19 849 576.21	19 849 382.19	308.95	19 815 674.79	19 815 479.58	543.14	19 546 489.04	19 546 291.48	1 646.1
ibm06	22 851 707.23	22 851 126.44	968.48	22 810 613.28	22 809 397.24	1 367.81	22 485 385.98	22 482 463.09	3 244.34
ibm07	31 859 043.41	31 858 190.12	1 756.23	31 793 025.95	31 791 499.79	2 179.03	31 289 564.73	31 285 852.15	4 966.1
ibm08	33 392 694.60	33 387 835.58	5 876.59	33 265 160.05	33 258 491.98	8 979.86	32 612 493.46	32 602 997.95	18 797.83
ibm09	39 973 390.54	39 971 847.55	2 730.41	39 872 424.57	39 869 726.35	3 263.63	39 114 849.23	39 109 763.17	7 920.87

References

[1] Alpert, C.J., Mehta, D.P.: Handbook of Algorithm for Physical Design Automation. Auerbach Publications, New York (2008)

[2] Becker, S., Bobin, J., Candes, E.J.: NESTA: a fast and accurate first-order method for sparse recovery. SIAM J. Imaging Sci. **4**(1), 1–39 (2011)

[3] Boyd, S., Xiao, L., Mutapcic, A.: Subgradient Methods. Lecture notes for EE364b. Stanford University (2007)

[4] Chu, C.: In: Wang, L.T., Chang, Y.W., Cheng, K.T. (eds.) Chapter 11: Placement in Electronic Design Automation: Synthesis, Verification, and Testing. Elsevier/Morgan, Kaufmann, San Francisco (2008)

[5] Dinh, Q.T., Savorgnan, C., Diehl, M.: Combining Lagrangian decomposition and excessive gap smoothing technique for solving large-scale separable convex optimization problems. Comput. Optim. Appl. **55**, 75–111 (2013)

[6] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York (1979)

[7] Hans-Jakob, L., Jörg, D.: Convex risk measures for portfolio optimization and concepts of flexibility. Math. Program. **104**(2–3), 541–559 (2005)

[8] Hariharan, L., Pucci, F.D.: Decentralized resource allocation in dynamic networks of agents. SIAM J. Imaging Sci. **19**(2), 911–940 (2008)

[9] Kim, M.C., Markov, I.L.: ComPLx: a competitive primal–dual Lagrange optimization for global placement. In: Design Automation Conference, pp. 747–752, 2012

[10] Kim, M.C., Lee, D.J., Markov, I.L.: SimPL: an effective placement algorithm. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **31**(1), 50–60 (2012a)

[11] Kim, M.C., Viswanathan, N., Alpert, C.J., Markov, I.L., Ramji, S.: MAPLE: multilevel adaptive placement for mixed-size Designs. In: International Symposium on Physical Design, pp. 193–200, 2012b

[12] Kim, M.C., Lee, D.J., Markov, I.L.: SimPL: an effective placement algorithm. Commun. ACM **55**(6), 105–113 (2013)

[13] Lin, T., Chu, C., Shinnerl, J.R., Bustany, I., Nedelchev, I.: POLAR: placement based on novel rough legalization and refinement. In: International Conference on Computer-Aided Design, pp. 357–362, 2013

[14] Markov, I.L., Hu, J., Kim, M.C.: Progress and challenges in VLSI placement research. In: International Conference on Computer-Aided Design, pp. 275–282, 2011

[15] Nemirovski, A.S., Yudin, D.: Problem complexity and method efficiency in optimization. In: Wiley-Interscience Series in Discrete Mathematics, vol. XV. Wiley, New York (1983)

[16] Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**(1), 127–152 (2005a)

[17] Nesterov, Y.: Excessive gap technique in nonsmooth convex minimization. SIAM J. Optim. **16**(1), 235–249 (2005b)

[18] Nesterov, Y.: Subgradient methods for huge-scale optimization problems. Math. Program. **146**(1–2), 275–297 (2014)

[19] Neveen, G., Jochen, K.: Faster and simpler algorithms for multicommodity flow and other fractional packing problems. SIAM J. Imaging Sci. **37**(2), 630–652 (2007)

[20] Schmidt, M., Roux, N.L., Bach, F.: Minimizing finite sums with the stochastic average gradient. Technical report, HAL 00860051 (2013)

[21] Spindler, P., Schlichtmann, U., Johannes, F.M.: Kraftwerk2—a fast force-directed quadratic placement approach using an accurate net model. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **27**(8), 1398–1411 (2008)