Master's Thesis

A multiplier method with variable augmented Lagrangian functions

Guidance

Associate Professor Nobuo YAMASHITA

Tomohiro NIIMI

Department of Applied Mathematics and Physics

Graduate School of Informatics

Kyoto University

February 2012

A multiplier method with variable augmented Lagrangian functions

Abstract

The multiplier method is a classical solution method for nonlinear programming problems. At each iteration, it minimizes an augmented Lagrangian function that consists of the constraint functions and the corresponding Lagrange multipliers. If the Lagrange multipliers in the augmented Lagrangian function are close to the exact Lagrange multipliers at an optimal solution, the multiplier method converges steadily. Since the conventional multiplier method uses inaccurate estimated Lagrange multipliers, it sometimes converges slowly. In this paper, we propose a novel multiplier method that allows the augmented Lagrangian function and its minimization problem to have variable constraints at each iteration. This allowance enables the new method to get more accurate estimated Lagrange multipliers by exploiting KKT points of the subproblems, and consequently to converge more efficiently and steadily. Moreover, we consider applying the new method with a gradient-type method to large-scale convex programming problems with linear constraints arising in the machine learning.

Contents

1 Introduction

We consider the following nonlinear programming problem:

$$\begin{array}{ll} \min\limits_{x} & f(x) \\ \text{s.t.} & g_i(x) \leq 0 \quad (i = 1, \cdots, l) \\ & h_j(x) = 0 \quad (j = 1, \cdots, m), \end{array} \tag{1.1}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$, $g_i : \mathbb{R}^n \mapsto \mathbb{R}$ and $h_j : \mathbb{R}^n \mapsto \mathbb{R}$ are twice continuously differentiable functions. For the problem (1.1), several Newton-type methods have been proposed. These methods can get a solution accurately with a small number of iterations. Among them, the interior point method and the sequential quadratic programming method (SQP) are well-studied and used for many applications in the real world. These days, we have to solve quite large-scale problems in a field of machine learning, signal processing, statistics, and so on. For these types of problems, the above Newton-type methods are not useful since they have to solve a large-scale linear equation or a convex quadratic programming problem in each iteration. For such large-scale convex programming problems, gradient-type methods such as the mirror descent method (MD) [1] and the accelerated proximal gradient method (APG) [12], which is an acceleration of MD are useful. In particular, in the field of machine learning, a special case of MD called the exponentiated gradient method [3] and a special case of APG called the fast iterative shrinkage algorithm [2] have been drawn much attention. Since these methods use only calculations of the gradient of the objective function and the projection on the feasible region, they efficiently solve the large-scale problems when the projection can be easily calculated. Thus, they can be applied to problems with simple constraints such as the upper and lower bound constraints or the unit simplex constraint. However, when the problem has complex constraints or many constraints, these methods are not efficient. Then we may use the multiplier method with the gradient-type method. The multiplier method is a classical method for nonlinear programming problems with complex constraints [5, 9, 10]. It is to obtain a local optimal solution of the problem (1.1) by minimizing an augmented Lagrangian function. The augmented Lagrangian function $Q : \mathbb{R}^{n+2l+2m} \mapsto \mathbb{R}$ is a Lagrangian function to which quadratic penalty terms of constraints are added, that is, it is defined by

$$\begin{aligned} &Q(x, \lambda, \mu, t, r) \\ &= f(x) + \sum_{i=1}^{l} \frac{1}{2t_i} \left[\max\{0, \lambda_i + t_i g_i(x)\}^2 - {\lambda_i}^2 \right] + \sum_{j=1}^{m} \left[\mu_j h_j(x) + \frac{r_j}{2} \{h_j(x)\}^2 \right], \end{aligned}$$

where $\lambda \in \mathbb{R}^l$ and $\mu \in \mathbb{R}^m$ are Lagrange multipliers of the inequality and the equality constraints, respectively, and $t \in \mathbb{R}^l$ and $r \in \mathbb{R}^m$ are penalty parameters. Note that the above definition of the augmented Lagrangian function is for the case where all the constraints are added into the Lagrangian function. Since the minimization problem of Q, which is the subproblem solved in the multiplier method, has only simple constraints or no constraint, it can be solved by the above gradient-type methods. Therefore, we may solve quite large-scale problems by the multiplier method with the gradient-type method. The efficiency of the multiplier method in general depends on the quality of the Lagrange multipliers λ and μ involved in Q. If (λ, μ) is close to a Lagrange multiplier

(λ^*, μ^*) corresponding to a KKT point (x^*, λ^*, μ^*) of (1.1), the method can find a solution x^* steadily and rapidly. On the other hand, if it is far from such a Lagrange multiplier (λ^*, μ^*), the penalty parameters t and r involved in Q should be very large in order to get an accurate solution, which causes a numerical difficulty. Therefore, we usually use a sequence of Lagrange multiplier (λ^k, μ^k) as an estimation of (λ^*, μ^*). The sequence is called the *estimated Lagrange multiplier* and is generated at each iteration. The frequently used update rules of the estimated Lagrange multiplier (λ^k, μ^k) is related to the gradient method for the dual problem of (1.1). Thus, the large number of iterations is required for (λ^k, μ^k) to converge to the accurate Lagrange multiplier (λ^*, μ^*). Note that the number corresponds to that of minimizing Q. Consequently, we may take much time to get a reasonable solution of (1.1) by the multiplier methods with the gradient method.

In this paper, we modify the conventional multiplier method that solves a subproblem with no constraint or with simple constraints fixed at all iterations. We allow the augmented Lagrangian function and its minimization problem to have variable constraints at each iteration. Then we minimize the variable augmented Lagrange function subject to some simple constraints remaining in the subproblem. Here, let constraints $g_i(x) \leq 0$ and $h_j(x) = 0$ remain in the subproblem of the kth iteration. The Lagrange multipliers $\bar{\lambda}_i$ and $\bar{\mu}_i$ for the remaining constraints $g_i(x) \leq 0$ and $h_j(x) = 0$ are simultaneously obtained in solving the subproblem. We can expect that the Lagrange multipliers $\bar{\lambda}_i$ and $\bar{\mu}_i$ are better approximations of λ_i^* and μ_j^* of the KKT point of (1.1). Therefore, we exploit the Lagrange multipliers $\bar{\lambda}_i^k$ and $\bar{\mu}_j^k$ at the $k+1$th iteration instead of the estimated Lagrange multipliers obtained by the conventional update rule. Therefore we would get a solution of (1.1) within fewer iterations.

The total efficiency of the proposed multiplier method deeply depends on how we solve subproblems with variable constraints at each iteration. We suppose that applications of the proposed method are large-scale convex programming problems with linear constraints arising in the machine learning, signal processing, statistics, and so on. In order to make use of advantages of the proposed method, we have to choose appropriate constraints remaining in the subproblem and an appropriate algorithm to solve the subproblem with them. In this paper, we propose adopting MD or APG to solve the subproblems. For convex problems with the boxed constraints or the unit simplex constraint, special implementation techniques for MD and APG have already been proposed [1, 12]. In this paper, we also give some implementation techniques for other linear constraints. These techniques allow us to apply the proposed multiplier method to wider classes of the problem (1.1).

This paper is organized as follows. In Section 2, we introduce the existing multiplier method. In Section 3, we propose a new multiplier method and show its global convergence. In Section 4, we discuss about the implementation. In particular, we introduce MD and APG as methods to solve subproblems of the proposed method when we solve large-scale convex programming problems with linear constraints. Also, we consider how to choose parameters in these methods. In Section 5, we present results of some numerical experiments. Finally, we give concluding remarks of this paper in Section 6.

Throughout this paper, we use the following notations. For a set S, $|S|$ denotes the number of elements of S. For a vector-valued function $g : \mathbb{R}^n \mapsto \mathbb{R}^l$,

∇g denotes transposition of the Jacobian of g, i.e.

$$\nabla g(x) := \begin{pmatrix} \frac{\partial}{\partial x_1} g_1(x) & \cdots & \frac{\partial}{\partial x_1} g_l(x) \\ \vdots & & \vdots \\ \frac{\partial}{\partial x_n} g_1(x) & \cdots & \frac{\partial}{\partial x_n} g_l(x) \end{pmatrix}.$$

For a subset $A \subset \{1, \cdots, m\}$, we define its complementary set as $\bar{A} := \{1, \cdots, m\} \setminus A$. For a vector y, let y_A be a $|A|$-dimensional column vactor constituting of y_i $(i \in A)$. Moreover, $\mathbb{R}^l_+$ and $\mathbb{R}^l_{++}$ denote an l-dimensional nonnegative vector and an l-dimensional positive vector, respectively.

2 The existing multiplier method

The multiplier method is one of the solution methods for nonlinear programming problems. Since we will propose a novel method based on this method in the next section, we briefly introduce the existing multiplier method here.

For simplicity, we first consider the following equality constrained problem.

$$\begin{array}{ll} \min\limits_{x} & f(x) \\ \text{s.t.} & h_j(x) = 0 \quad (j = 1, \cdots, m). \end{array} \tag{2.2}$$

The KKT conditions of (2.2) are written as

$$\nabla f(x^*) + \sum_{j=1}^{m} \mu_j^* \nabla h_j(x^*) = 0, \quad h_j(x^*) = 0 \ (j = 1, \cdots, m). \tag{2.3}$$

The multiplier method uses the augmented Lagrangian function defined by

$$\hat{Q}(x, \mu, r) = f(x) + \sum_{j=1}^{m} \mu_j h_j(x) + \sum_{j=1}^{m} r_j \{h_j(x)\}^2. \tag{2.4}$$

Then, it solves the following unconstrained optimization problem instead of the original problem (2.2).

$$\begin{array}{ll} \min & \hat{Q}(x, \mu, r) \\ \text{s.t.} & x \in \mathbb{R}^n. \end{array} \tag{2.5}$$

The next proposition shows relations between the original problem (2.2) and the minimization problem (2.5) of the augmented Lagrangian function.

Proposition 2.1. *Let (x^*, μ^*) be a KKT point of the problem (2.2). Suppose that the second-order sufficient conditions hold at x^*. Then, we have the following properties.*

1. *There exists a positive scalar r^* such that x^* is a local minimizer of $\hat{Q}(x, \mu^*, r)$ for all $r \geq r^*$.*
2. *If a local minimizer $\hat{x}$ of $\hat{Q}(x, \hat{\mu}, \hat{r})$ for some $\hat{\mu}$ and $\hat{r}$ satisfies $h_j(\hat{x}) = 0$ $(j = 1, \cdots, m)$, then $\hat{x}$ is a local minimizer of the problem (2.2).*

This proposition implies that, when μ^* is the exact Lagrange multiplier of (2.3) and r is sufficiently large, we can expect to get a local optimal solution x^* by minimizing $\hat{Q}(x, \mu^*, r)$ with no constraint. However, we do not know μ^* and r^* in advance. Thus, we usually use an approximation of μ^* instead. We call the approximation the *estimated Lagrange multiplier*, which is denoted by μ^k. The most existing multiplier method adopts the sequence $\{\mu^k\}$ generated by

$$\mu_j^{k+1} = \mu_j^k + r_j^k h_j(x^k), \tag{2.6}$$

where x^k is a solution of

$$\begin{array}{ll} \min_x & \hat{Q}(x, \mu^k, r^k) \\ \text{s.t.} & x \in \mathbb{R}^n. \end{array} \tag{2.7}$$

Since r^* is also unknown, we simultaneously enlarge r if necessary.

Now we describe the multiplier method for (2.2) [9, 10].

The Multiplier Method

Step 0. Choose $\mu_j^0 \in \mathbb{R}_+^m, r_j^0 \in \mathbb{R}_{++}^m$, $\alpha > 1$ and $\beta \in (0, 1)$. Set $c^0 = \infty$ and $k = 0$.

Step 1. Obtain a minimizer x^k of the subproblem (2.7).

Step 2. Set $V = \{j|\ |h_j(x^k)| > \beta c^k\}$. If $\max_j |h_j(x^k)| > c^k$, set $\mu^{k+1} = \mu^k, c^{k+1} = c^k$ and go to Step 5.

Step 3. Set $c^{k+1} = \max_j |h_j(x^k)|$. If c^{k+1} is sufficiently small, then stop.

Step 4. Update the estimated Lagrange multipliers by

$$\mu_j^{k+1} = \mu_j^k + r_j^k h_j(x^k)\ \ (j = 1, \cdots, m).$$

Step 5. Update the penalty parameters by

$$r_j^{k+1} = \alpha r_j^k\ \ (j \in V)\ \ ; \qquad r_j^{k+1} = r_j^k\ \ (\text{otherwise}).$$

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

In Step 1, we get the next iterate x^k by minimizing the augmented Lagrangian function. In Step 4, the estimated Lagrange multiplier μ^k is updated. In Step 5, we increase values of penalty parameters if necessary. Note that the set V denotes a set of indices where the violation of the constraints are relatively large.

Since

$$\begin{aligned} 0 &\approx \nabla \hat{Q}(x^k, \mu^k, r^k) \\ &= \nabla f(x^k) + \sum_{j=1}^m \nabla h_j(x^k)\{\mu_j^k + r_j^k h_j(x^k)\} \\ &= \nabla f(x^k) + \sum_{j=1}^m \mu_j^{k+1} \nabla h_j(x^k), \end{aligned}$$

the point (x^k, μ^{k+1}) satisfies the first KKT condition in (2.3), which implies the validity of (2.6). The update rule (2.6) can be regarded as the steepest descent method for the dual problem of (2.2). As another update formula, the following one has been proposed.

$$\mu^{k+1} = \mu^k + \left(\nabla h(x^k)^\top \nabla_x^2 \hat{Q}(x^k, \mu^k, r^k)^{-1} \nabla h(x^k)\right)^{-1} h(x^k).$$

This update rule can be regarded as the Newton method for the dual problem of (2.2), and hence it is expected to converge faster than (2.6). However, it is not so practical for large-scale problems since it would take much time to calculate.

Here we briefly explain how to apply the above multiplier method to the following inequality constrained problem.

$$\begin{array}{ll} \min\limits_x & f(x) \\ \text{s.t.} & g_i(x) \le 0 \quad (i = 1, \cdots, l). \end{array}$$

By using slack variables y_i $(i = 1, \cdots, l)$, this problem can be transformed to

$$\begin{array}{ll} \min\limits_x & f(x) \\ \text{s.t.} & g_i(x) + (y_i)^2 = 0 \quad (i = 1, \cdots, l). \end{array}$$

Then, the augmented Lagrangian function corresponding to (2.4) is written as

$$\acute{Q}(x, y, \lambda, t) = f(x) + \sum_{i=1}^{l} \lambda_i \{g_i(x) + (y_i)^2\} + \frac{1}{2} \sum_{i=1}^{l} t_i \{g_i(x) + (y_i)^2\}^2.$$

We minimize this function with respect to y. From the first-order optimality condition:

$$\frac{\partial \acute{Q}(x, y, \lambda, t)}{\partial y_i} = 2y_i[\lambda_i + t_i\{g_i(x) + (y_i)^2\}] = 0,$$

we have

$$(y_i)^2 = \begin{cases} -\frac{\lambda_i}{t_i} - g_i(x) & (\lambda_i + t_i g_i(x) < 0) \\ 0 & (\lambda_i + t_i g_i(x) \ge 0), \end{cases}$$

and hence $g_i(x) + (y_i)^2 = \max\{g_i(x), -\lambda_i/t_i\}$. Therefore, the minimization of $\acute{Q}(x, y, \lambda, t)$ with respect to (x, y) is equivalent to the minimization of the following function with respect to x.

$$\check{Q}(x, \lambda, t) = f(x) + \sum_{i=1}^{l} \frac{1}{2t_i} \left[\max\{0, \lambda_i + t_i g_i(x)\}^2 - (\lambda_i)^2\right].$$

As well, the update formula (2.6) of the estimated Lagrange multiplier can be written as

$$\lambda_i^{k+1} = \lambda_i^k t_i^k \{g_i(x^k) + (y_i^k)^2\} = \max\{0, \lambda_i^k + t_i^k g_i(x^k)\}. \tag{2.8}$$

In this way, the multiplier method can be extended for problems with both equality constraints and inequality constraints, and even for the case where there exists some fixed constraints that are not involved in the augmented Lagrangian function and remain in the subproblem for all iterations.

3 A multiplier method with variable augmented Lagrangian functions

In this section, we extend the multiplier method presented in the previous section. The purpose of the extended method is to get the following KKT point (x^*, μ^*, λ^*) of (1.1).

$$\nabla f(x^*) + \nabla g(x^*)\lambda^* + \nabla h(x^*)\mu^* = 0, \tag{3.9}$$

$$g_i(x^*) \le 0, \quad \lambda_i^* \ge 0, \quad \lambda_i^* g_i(x^*) = 0 \quad (i = 1, \cdots, l), \tag{3.10}$$

$$h_j(x^*) = 0 \quad (j = 1, \cdots, m). \tag{3.11}$$

The conventional multiplier method in the previous section uses an augmented Lagrangian function involving all or fixed some constraints, and solves the subproblems with none or fixed other constraints at each iteration. Recently, for solving large-scale problems with simple constraints such as the upper and lower bound constraints or the unit simplex constraint, several efficient methods have been devised [1, 6, 7]. When we apply these methods for solving the subproblems of the multiplier method, we can make some simple constraints still remain in the subproblem. On the other hand, the multiplier method may be slow when we update the estimated Lagrange multiplier (λ^k, μ^k) by (2.6) and (2.8). In such cases, the penalty parameters may have large values, and hence numerical difficulties occur. Therefore, we exploit Lagrange multipliers at a KKT point of the subproblem, and we replace constraints of the subproblem at each iteration. The Lagrange multipliers $\bar{\lambda}_i^k$ and $\bar{\mu}_j^k$ of the constraints remaining as a constraint of the subproblem are obtained by minimizing the augmented Lagrangian function subject to these constraints at the kth iteration. The Lagrange multipliers $\bar{\lambda}_i^k$ and $\bar{\mu}_j^k$ are expected to be better approximations of λ_i^* and μ_j^* in (3.9)-(3.11) than those updated by (2.6) and (2.8). Thus, at the next iteration, we include these constraints into the augmented Lagrangian function with the Lagrange multipliers $\bar{\lambda}_i^k$ and $\bar{\mu}_j^k$.

We concretely formulate the above idea. First, we define some notations about constraints and Lagrange multipliers.

Definition 1. *1. An inequality constraint is called the involved inequality constraint at the kth iteration if the augmented Lagrangian function involves it at the kth iteration. Similarly, an equality constraint is called the involved equality constraint at the kth iteration if the augmented Lagrangian function involves it at the kth iteration.*

2. An inequality constraint is called the remaining inequality constraint at the kth iteration if it remains as a constraint of the subproblem at the kth iteration. Similarly, an equality constraint is called the remaining equality constraint at the kth iteration if it remains as a constraint of the subproblem at the kth iteration.

3. G_k and H_k denote the index sets of the remaining inequality constraints and remaining equality constraints at the kth iteration, respectively.

4. We call the Lagrange multiplier $(\lambda_{\rm opt}, \mu_{\rm opt})$ of a problem optimal Lagrange multiplier if $(x_{\rm opt}, \lambda_{\rm opt}, \mu_{\rm opt})$ is a KKT point of the problem.

Note that we change the index sets G_k and H_k at each iteration. Using G_k and H_k, the augmented Lagrangian function Q_k at the kth iteration is defined by

$$\begin{aligned} Q_k(x,\lambda,\mu,t,r) = \quad & f(x) + \sum_{i\notin G_k} \frac{1}{2t_i}\left[\max\{0,\lambda_i + t_i g_i(x)\}^2 - {\lambda_i}^2\right] \\ & + \sum_{j\notin H_k} \left[\mu_j h_j(x) + \frac{r_j}{2}\{h_j(x)\}^2\right]. \end{aligned} \tag{3.12}$$

Then, the subproblem at the kth iteration is defined by

$$\begin{array}{lll} \min\limits_{x} & Q_k(x,\lambda^k,\mu^k,t^k,r^k) & \\ \text{s.t.} & g_i(x) \le 0 & (i\in G_k) \\ & h_j(x) = 0 & (j\in H_k). \end{array} \tag{3.13}$$

Here let $\tilde{x}^k$ be an exact solution of the subproblem (3.13). Then, under some constraint qualifications, there exists the optimal Lagrange multipliers $\tilde{\lambda}_i^k$ $(i\in G_k)$ and $\tilde{\mu}_j^k$ $(j\in H_k)$ of (3.13) such that

$$\begin{aligned} & \nabla_x Q_k(\tilde{x}^k,\lambda^k,\mu^k,t^k,r^k) + \sum_{i\in G_k} \tilde{\lambda}_i^k \nabla g_i(\tilde{x}^k) + \sum_{j\in H_k} \tilde{\mu}_j^k \nabla h_j(\bar{x}^k) = 0, \\ & g_i(\tilde{x}^k) \le 0,\ \ \tilde{\lambda}_i^k \ge 0,\ \ \tilde{\lambda}_i^k g_i(x^k) = 0 \quad (i\in G_k), \\ & h_j(\tilde{x}^k) = 0 \ \ (j\in H_k). \end{aligned}$$

We can expect to obtain an optimal solution of the main problem (1.1) efficiently by using the optimal Lagrange multipliers $\tilde{\lambda}_i^k$ $(i\in G_k)$ and $\tilde{\mu}_j^k$ $(j\in H_k)$ corresponding to the remaining constraints. However, it is time consuming to get the exact KKT point $(\tilde{x}^k,\tilde{\mu}^k,\tilde{\lambda}^k)$ in each iteration. Instead, we adopt an approximate KKT point $(x^k,\bar{\mu}^k,\bar{\lambda}^k)$ satisfying the following conditions.

$$\begin{aligned} & \left\| \nabla_x Q_k(x^k,\lambda^k,\mu^k,t^k,r^k) + \textstyle\sum_{i\in G_k} \bar{\lambda}_i^k \nabla g_i(x^k) + \sum_{j\in H_k} \bar{\mu}_j^k \nabla h_j(x^k) \right\| < \epsilon^k, \\ & \left|\max\left\{g_i(x^k), -\bar{\lambda}_i^k\right\}\right| < \xi^k \quad (i\in G_k), \quad |h_j(x^k)| < \zeta^k \quad (j\in H_k), \end{aligned} \tag{3.14}$$

where $\{\epsilon^k\},\{\xi^k\}$ and $\{\zeta^k\}$ are positive scalar sequences that converge to 0 as $k\to\infty$.

We describe the proposed multiplier method as follows. We call this method the *Switching Constraints Multiplier Method* (SCM). Note that SCM is reduced to the conventional multiplier method when G_k and H_k are fixed.

The switching constraints multiplier method

Step 0. Choose initial guess $\lambda^0\in\mathbb{R}^l_+$ and $\mu^0\in\mathbb{R}^m_+$. Moreover, choose parameters $t^0\in\mathbb{R}^l_{++}, r^0\in\mathbb{R}^m_{++}$, $\alpha>1$, $\beta\in(0,1)$ and a small positive scalar ϵ. Set $c^0=\infty$ and $k=0$.

Step 1. Choose index sets G_k and H_k of the remaining constraints. Then, define the augmented Lagrangian function $Q_k(x,\lambda^k,\mu^k,t^k,r^k)$ by (3.12).

Step 2. Obtain the approximate KKT point $(x^k,\bar{\mu}^k,\bar{\lambda}^k)$ of (3.13) satisfying (3.14).

Step 3. Set

$$V_g^k = \left\{ i \;\middle|\; \left| \max\left\{ g_i(x^k), -\frac{\lambda_i^k}{t_i^k} \right\} \right| > \beta c^k \right\},$$
$$V_h^k = \left\{ j \;\middle|\; |h_j(x^k)| > \beta c^k, \right\}.$$

If either $\max_i \left| \max\left\{ g_i(x^k), -\frac{\lambda_i^k}{t_i^k} \right\} \right| > c^k$ or $\max_j \left| h_j(x^k) \right| > c^k$ holds, then set $c^{k+1} = c^k$ and go to Step 5.

Step 4. Set

$$c^{k+1} = \max\left\{ \max_i \left| \max\left\{ g_i(x^k), -\frac{\lambda_i^k}{t_i^k} \right\} \right|, \max_j |h_j(x^k)| \right\}.$$

If

$$\max\left\{ \left\| \nabla_x Q_k(x^k, \lambda^k, \mu^k, t^k, r^k) + \sum_{i=1}^{l} \lambda_i^k \nabla g_i(x^k) + \sum_{j=1}^{m} \mu_j^k \nabla h_j(x^k) \right\|, c^{k+1} \right\} < \epsilon,$$

then stop.

Step 5. Update the estimated Lagrange multipliers by

$$\begin{array}{llll} \lambda_i^{k+1} = \bar{\lambda}_i^k \quad (i \in G_k) & ; & \lambda_i^{k+1} = \max\{0, \lambda_i^k + t_i^k g_i(x^k)\} & (i \in \{1, \cdots, l\} \setminus G_k), \\ \mu_j^{k+1} = \bar{\mu}_j^k \quad (j \in H_k) & ; & \mu_j^{k+1} = \mu_j^k + r_j^k h_j(x^k) & (j \in \{1, \cdots, m\} \setminus H_k). \end{array} \tag{3.15}$$

Step 6. Update the penalty parameters by

$$t_i^{k+1} = \alpha t_i^k \;\; (i \in V_g^k) \;\; ; \quad t_i^{k+1} = t_i^k \;\; \text{(otherwise)},$$

$$r_j^{k+1} = \alpha r_j^k \;\; (j \in V_h^k) \;\; ; \quad r_j^{k+1} = r_j^k \;\; \text{(otherwise)}.$$

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

We choose the index sets G_k and H_k of the remaining constraints in Step 1. This is an important task since the difficulties of the subproblem and the efficiency of the proposed SCM deeply depend on the sets. We will discuss how to choose the sets in Section 4.2. In Step 2, we solve the subproblem that minimizes the augmented Lagrangian function over the remaining constraints chosen in Step 1. Let $\tilde{c}^k$ be the maximum of violations $|\max\{g_i(x^k), -\lambda_i^k/t_i^k\}|$ and $|h_j(x^k)|$ at the kth iteration. The parameter c^k in the algorithm represents the minimum value of $\tilde{c}^k$ through past k iterations. Thus, if x^k tends to be feasible, c^k goes to 0. When c^k is away from 0, then some constraints are violated. Then, we make the penalty parameters corresponding to such constraints large in Step 6. Note that the penalty parameters corresponding to the remaining constraints do not change. We update the estimated Lagrange multipliers in Step 5. As mentioned above, we exploit the Lagrange multipliers $\bar{\lambda}_i^k$ and $\bar{\mu}_j^k$ of the subproblem while we update the estimated Lagrange multipliers of the involved constraints by (2.6) and (2.8). We stop the algorithm in Step 4 if the residuals of KKT conditions are sufficiently small.

3.1 Convergence analysis

We show the global convergence of the problem (1.1), that is, we show that the sequence $(x^k, \bar{\mu}^k, \bar{\lambda}^k)$ generated by SCM converges to a KKT point satisfying (3.9)-(3.11).

To this end, we need the following usual assumptions for the multiplier method.

Assumption 1. *The parameter ϵ in the termination criterion is 0. We can calculate $(x^k, \bar{\lambda}_i^k, \bar{\mu}_j^k)$ satisfying (3.14) in each iteration. Furthermore, a subsequence $\{x^k\}_K$ converges to $\bar{x}$, and $\nabla g_i(\bar{x})$ $(i \in \Omega), \nabla h_j(\bar{x})$ $(j = 1, \cdots, m)$ are linearly independent, i.e.,*

$$\operatorname{rank} \left[\begin{array}{cc} \nabla g_\Omega(\bar{x}) & \nabla h(\bar{x}) \end{array} \right] = |\Omega| + m,$$

where $\Omega := \{i \mid g_i(\bar{x}) = 0\}$.

Under Assumption 1, we first show the following four lemmas.

Lemma 3.1. *Suppose that Assumption 1 holds. Then, we have*

$$\left\{\lambda_i^k\right\}_K \to 0$$

for all i such that $g_i(\bar{x}) < 0$.

Proof. Let i be an index satisfying $g_i(\bar{x}) < 0$. Then, there exists a positive scalar τ_i such that

$$g_i(x^k) < -\tau_i < -\xi^k \tag{3.16}$$

for sufficiently large $k \in K$. Since $\xi^k > 0$, the inequality (3.16) implies $|g_i(x^k)| > \xi^k$. Suppose that $i \in G_k$. It then follows from (3.14) that $|\max\{g_i(x^k), -\bar{\lambda}_i^k\}| < \xi^k$. Since $|g_i(x^k)| > \xi^k$, we have

$$|-\lambda_i^k| < \xi^k \quad (\forall i \in G_k). \tag{3.17}$$

On the other hand, suppose that $i \notin G_k$. Then, we have $\lambda_i^{k+1} = \max\{0, \lambda_i^k + t_i^k g_i(x^k)\}$ from the update rule (3.15) in Step 5. Since the penalty parameter t_i^k is nondecreasing, it follows from (3.16) that

$$t_i^k g_i(x^k) \le t^0 g_i(x^k) < -t^0 \tau_i < 0 \tag{3.18}$$

for sufficiently large k.

Here we prove $\lambda_i^k \to 0$ by contradiction. Assume that $\lambda_i^k \not\to 0$ holds. Then, there exists an infinite subsequence $\tilde{K} \subset K$ and a fixed scalar $\tilde{\tau}_i > 0$ such that

$$|\lambda_i^k| > \tilde{\tau}_i \quad (\forall k \in \tilde{K}).$$

Since $\xi^k \to 0$, there exists a positive integer k_1 such that $\xi^k < \tilde{\tau}_i$ $(\forall k > k_1)$. Therefore, (3.17) implies that $i \notin G_k$ for all $k > k_1$. It then follows from (3.15) and (3.18) that

$$|\lambda_i^{k+1}| \le \max\{0, \lambda_i^k - t_i^0 \tau_i\} \quad (\forall k > k_1),$$

which contradicts the hypothesis $\lambda_i^k \not\to 0$. □

Lemma 3.2. *Suppose that Assumption 1 holds. Let M^k be a matrix defined by*

$$M^k := \Big(\left[\begin{array}{cc} \nabla g_\Omega(x^k) & \nabla h(x^k) \end{array} \right]^\top \left[\begin{array}{cc} \nabla g_\Omega(x^k) & \nabla h(x^k) \end{array} \right] \Big)^{-1} \left[\begin{array}{cc} \nabla g_\Omega(x^k) & \nabla h(x^k) \end{array} \right]^\top.$$

Then, $\{\|M^k\|\}$, $\{\lambda^k\}$ and $\{\mu^k\}$ are bounded for sufficiently large k.

Proof. Assumption 1 implies that $\left[\begin{array}{cc} \nabla g_\Omega(\bar{x}) & \nabla h(\bar{x}) \end{array} \right]^\top \left[\begin{array}{cc} \nabla g_\Omega(\bar{x}) & \nabla h(\bar{x}) \end{array} \right]$ is nonsingular. Then, since $\nabla g_\Omega(x)$ and $\nabla h(x)$ are continuous, $\{\|M^k\|\}$ is well-defined and bounded for sufficiently large k.

Now we prove the boundedness of $\{\lambda^k\}$ and $\{\mu^k\}$. We have

$$\begin{aligned}
&\nabla_x Q_k(x^k, \lambda^k, \mu^k, t^k, r^k) + \sum_{i \in G_k} \bar{\lambda}_i^k \nabla g_i(x^k) + \sum_{j \in H_k} \bar{\mu}_j^k \nabla h_j(x^k) \\
&= \nabla f(x^k) + \sum_{i=1}^{l} \lambda_i^{k+1} \nabla g_i(x^k) + \sum_{j=1}^{m} \mu_j^{k+1} \nabla h_j(x^k) \\
&= \nabla f(x^k) + \nabla g_\Omega(x^k) \lambda_\Omega^{k+1} + \nabla g_{\bar{\Omega}}(x^k) \lambda_{\bar{\Omega}}^{k+1} + \nabla h(x^k) \mu^{k+1} \\
&= \nabla f(x^k) + \left[\begin{array}{cc} \nabla g_\Omega(x^k) & \nabla h(x^k) \end{array} \right] \left(\begin{array}{c} \lambda_\Omega^{k+1} \\ \mu^{k+1} \end{array} \right) + \nabla g_{\bar{\Omega}}(x^k) \lambda_{\bar{\Omega}}^{k+1}. \qquad (3.19)
\end{aligned}$$

Since $\mathrm{rank} \left[\begin{array}{cc} \nabla g_\Omega(\bar{x}) & \nabla h(x^*) \end{array} \right] = |\Omega| + m$,

$$\mathrm{rank} \left[\begin{array}{cc} \nabla g_\Omega(x^k) & \nabla h(x^k) \end{array} \right] = |\Omega| + m \qquad (3.20)$$

holds for sufficiently large k. Without loss of generality, we assume that (3.20) holds for all $k \in K$. By multiplying (3.19) by M^k, we obtain

$$\begin{aligned}
\left(\begin{array}{c} \lambda_\Omega^{k+1} \\ \mu^{k+1} \end{array} \right) = M^k \Bigg(& \nabla_x Q_k(x^k, \lambda^k, \mu^k, t^k, r^k) + \sum_{i \in G_k} \bar{\lambda}_i^k \nabla g_i(x^k) \\
& + \sum_{j \in H_k} \bar{\mu}_j^k \nabla h_j(x^k) - \nabla f(x^k) - \nabla g_{\bar{\Omega}}(x^k) \lambda_{\bar{\Omega}}^{k+1} \Bigg). \qquad (3.21)
\end{aligned}$$

Lemma 3.1 and $\|\nabla g_{\bar{\Omega}}(x^k)\| < \infty$ imply

$$\|\nabla g_{\bar{\Omega}}(x^k) \lambda_{\bar{\Omega}}^{k+1}\| \to 0. \qquad (3.22)$$

It then follows from (3.14) and $\|\nabla f(x^k)\| < \infty$ that (3.21) implies the boundedness of $\{\lambda_\Omega^k\}$ and $\{\mu^k\}$. □

Here we define index sets related to the involved inequality constraints as follows.

$$\begin{aligned}
\bar{G}^i &:= \{k \in K \mid i \notin G_k\}, \\
I_1 &:= \left\{ i \;\middle|\; |\bar{G}^k| < \infty \right\}, \qquad (3.23) \\
I_2 &:= \left\{ i \;\middle|\; |\bar{G}^k| = \infty, t_i^k \to \infty \right\}, \qquad (3.24) \\
I_3 &:= \left\{ i \;\middle|\; |\bar{G}^k| = \infty, \sup_k t_i^k < \infty \right\}. \qquad (3.25)
\end{aligned}$$

Note that $\bar{G}^i$ denotes the set of the iteration number k where the constraint $g_i(x) \leq 0$ is an involved inequality constraint. The set I_1 denotes the set of the index i such that $\bar{G}^i$ is finite, that is, the index set of the constraint that becomes the involved inequality constraint finite times. I_2 denotes the set of the constraint that infinitely becomes the involved constraint, and the penalty parameter t_i^k diverges to infinity. I_3 denotes the same as I_2 but the penalty parameter t_i^k is bounded from above.

As well, we define index sets related to the involved equality constraints as follows.

$$\begin{aligned}
\bar{H}^j &:= \{k \in K \mid j \notin H_k\}, \\
J_1 &:= \left\{j \;\middle|\; |\bar{H}^j| < \infty\right\}, && (3.26)\\
J_2 &:= \left\{j \;\middle|\; |\bar{H}^j| = \infty, r_j^k \to \infty\right\}, && (3.27)\\
J_3 &:= \left\{j \;\middle|\; |\bar{H}^j| = \infty, \sup_k r_j^k < \infty\right\}. && (3.28)
\end{aligned}$$

Note that $\{1, \cdots, l\} = I_1 \cup I_2 \cup I_3$ and $\{1, \cdots, m\} = J_1 \cup J_2 \cup J_3$.

Lemma 3.3. *Suppose that Assumption 1 holds. Then, we have*

$$\left\{\max_{i \in I_1} \left| \max\left\{g_i(x^k), -\frac{\lambda_i^k}{t_i^k}\right\}\right|\right\}_K \to 0, \tag{3.29}$$

$$\left\{\max_{i \in I_2} \left| \max\left\{g_i(x^k), -\frac{\lambda_i^k}{t_i^k}\right\}\right|\right\}_K \to 0. \tag{3.30}$$

Furthermore, there exits a positive integer k_2 such that

$$\max_{i \in I_3} \left| \max\left\{g_i(x^k), -\frac{\lambda_i^k}{t_i^k}\right\}\right| \leq \beta c^k \quad (\forall k > k_2, k \in K). \tag{3.31}$$

Proof. Suppose that $i_1 \in I_1$. Then, the constraint $g_{i_1} \leq 0$ is a remaining constraint for sufficiently large k. Therefore, it follows from (3.14) that

$$\left\{\max\{g_{i_1}(x^k), -\bar{\lambda}_{i_1}^k\}\right\}_K \to 0,$$

which yields (3.29).

Next, suppose that $i_2 \in I_2$. Then, since $\{x^k\}_K$ and $\{\lambda^k\}_K$ are bounded, we obtain $|\max\{0, \lambda_{i_2}^k + t_{i_2}^k g_{i_2}(x^k)\}| < \infty$ for all $k \in \bar{G}^{i_2}$. Since $\bar{G}^{i_2}$ is a infinite set and $t_{i_2}^k \to \infty$, we have $g_{i_2}(\bar{x}) \leq 0$ and

$$|\max\{g_{i_2}(x^k), -\lambda_{i_2}^k / t_{i_2}^k\}| \to 0,$$

and hence (3.30) holds.

Finally, suppose that $i_3 \in I_3$. Then, since the penalty parameter t_i^k is not updated for sufficiently large k, there exists a positive integer k_3 such that $i_3 \notin {I_g}^k$ holds for all $k > k_3$. Therefore, we have $\|\max\{g_{i_3}(x^k), -\lambda_{i_3}^k / t_{i_3}^k\}\| \leq \beta c^k$ for all $k > k_3$, and hence (3.31) holds. □

Lemma 3.4. *Suppose that Assumption 1 holds. Then, we have*

$$\left\{\max_{j\in J_1}|h_j(x^k)|\right\}_K \to 0, \tag{3.32}$$

$$\left\{\max_{j\in J_2}|h_j(x^k)|\right\}_K \to 0. \tag{3.33}$$

Furthermore, there exists a positive integer k_4 such that

$$\max_{j\in J_3}|h_j(x^k)| \le \beta c^k \quad (\forall k > k_4, k \in K). \tag{3.34}$$

Proof. Suppose that $j_1 \in J_1$. Then, there exists a positive integer k_5 such that $j_1 \in H_k \ \ (\forall k > k_5, k \in K)$. Note that the constraint $h_{j_1}(x) = 0$ is a remaining constraint for all $k > k_5$. Therefore, the condition (3.14) implies $|h_j(x^k)| < \zeta^k \ \ (\forall j \in H_k)$, and hence $\{h_{j_1}(x^k)\}_K \to 0$. Thus, (3.32) holds.

Next, suppose that $j_2 \in J_2$. Then, since $\{\mu^k\}$ is bounded, we obtain

$$|\mu^k_{j_2} + r^k_{j_2} h_{j_2}(x^k)| < \infty \ \ (\forall k \in \bar{H}^{j_2}).$$

It then follows from $|\bar{H}^{j_2}| = \infty$ and $r^k_{j_2} \to \infty$ that $h_{j_2}(x^k) \to 0 \ \ (k \to \infty, k \in \bar{H}^{j_2})$. Thus, (3.33) holds.

Finally suppose that $j_3 \in J_3$. Then, there exists a positive integer k_6 such that $j_3 \notin V^k_h$ for all $k > k_6$. Therefore, we have $|h_{j_3}(x^k)| \le \beta c^k$ for all $k > k_6$, and hence (3.34) holds. □

Now we show the following main theorem which shows the global convergence of SCM.

Theorem 3.1. *Suppose that Assumption 1 holds. Then, the subsequences $\{\lambda^k\}_K$ and $\{\mu^k\}_K$ converge to some points $\bar{\lambda}$ and $\bar{\mu}$, respectively. Moreover $(\bar{x}, \bar{\lambda}, \bar{\mu})$ satisfies the first-order necessary conditions (3.9)-(3.11).*

Proof. Since $\{\|M^k\|\}_K$ is bounded from Lemma 3.2, we may suppose that $\{\|M^k\|\}_K \to \bar{M}$. It then follows from (3.14), (3.21), (3.22) and Lemma 3.1 that

$$\begin{pmatrix} \lambda^k_\Omega \\ \mu^k \end{pmatrix} \to \begin{pmatrix} \bar{\lambda}_\Omega \\ \bar{\mu} \end{pmatrix} := -\bar{M}\nabla f(\bar{x}), \quad \lambda^k_{\bar{\Omega}} \to \bar{\lambda}_{\bar{\Omega}} := 0,$$

that is, μ^k converges to $\bar{\mu}$ and λ^k converges to $\bar{\lambda}$, where

$$\bar{\lambda}_i := \begin{cases} \bar{\lambda}_\Omega & (i \in \Omega) \\ 0 & (i \notin \Omega). \end{cases}$$

By using (3.19) and (3.22) again, $(\bar{x}, \bar{\lambda}, \bar{\mu})$ satisfies the first KKT condition (3.9), that is,

$$\nabla f(\bar{x}) + \nabla h(\bar{x})\bar{\mu} + \nabla g(\bar{x})\bar{\lambda} = 0.$$

Next we show the feasibility of $\bar{x}$, i.e., $g_i(\bar{x}) \le 0 \ (i = 1, \cdots, l)$ and $h_j(\bar{x}) = 0 \ (j = 1, \cdots, m)$. Note that the update formula of c^k in Step 4 is

$$c^{k+1} = \min\left\{\max\left(c^k_g, c^k_h\right), c^k\right\}, \tag{3.35}$$

where

$$c_g^k = \max_i \left| \max\left\{ g_i(x^k), -\frac{\lambda_i^k}{t_i^k} \right\} \right|, \quad c_h^k = \max_j |h_j(x^k)|.$$

Using the sets I_1, I_2, I_3, J_1, J_2 and J_3 defined in (3.23)-(3.28), we can write

$$\begin{aligned} c_g^k = \quad & \max\left\{ \max_{i\in I_1} \left| \max\left\{ g_i(x^k), -\frac{\lambda_i^k}{t_i^k} \right\} \right|, \max_{i\in I_2} \left| \max\left\{ g_i(x^k), -\frac{\lambda_i^k}{t_i^k} \right\} \right|, \right. \\ & \left. \max_{i\in I_3} \left| \max\left\{ g_i(x^k), -\frac{\lambda_i^k}{t_i^k} \right\} \right| \right\}, \end{aligned} \tag{3.36}$$

$$c_h^k = \quad \max\left\{ \max_{j\in J_1} |h_j(x^k)|, \max_{j\in J_2} |h_j(x^k)|, \max_{j\in J_3} |h_j(x^k)| \right\}. \tag{3.37}$$

We prove $c^k \to 0$ by contradiction. Assume that $c^k \not\to 0$ holds. Then, since c^k is a nonincreasing sequence, there exists a positive constant $\bar{c}$ such that $c^k > \bar{c}$ for all k. Then, since $\beta \in (0,1)$, there exists a positive integer k_7 such that

$$c^{k+1} > \beta c^k > \beta \bar{c} \quad (\forall k > k_7). \tag{3.38}$$

From Lemmas 3.3 and 3.4,

$$\beta\bar{c} > \max_{i\in I_1} \left| \max\left\{ g_i(x^k), -\frac{\lambda_i^k}{t_i^k} \right\} \right|, \qquad \beta\bar{c} > \max_{i\in I_2} \left| \max\left\{ g_i(x^k), -\frac{\lambda_i^k}{t_i^k} \right\} \right|,$$
$$\beta\bar{c} > \max_{j\in J_1} |h_j(x^k)|, \qquad \beta\bar{c} > \max_{j\in J_2} |h_j(x^k)|$$

hold for sufficiently large k. It then follows from (3.35)-(3.37) that

$$\begin{aligned} c^{k+1} &= \min\left\{ \max\left\{ \max_{i\in I_3} \left| \max\left\{ g_i(x^k), -\frac{\lambda_i^k}{t_i^k} \right\} \right|, \max_{j\in J_3} |h_j(x^k)| \right\}, c^k \right\} \\ &\le \max\left\{ \max_{i\in I_3} \left| \max\left\{ g_i(x^k), -\frac{\lambda_i^k}{t_i^k} \right\} \right|, \max_{j\in J_3} |h_j(x^k)| \right\} \\ &\le \max\{\beta c^k, \beta c^k\} \\ &= \beta c^k \end{aligned}$$

for sufficiently large k, where the above two inequalities follow from (3.31) and (3.34). This contradicts (3.38). Consequently, c^k converges to 0, and hence

$$\max\left\{ g_{i_3}(x^k), -\frac{\lambda_{i_3}^k}{t_{i_3}^k} \right\} \to 0, \quad \max_{j\in J_3} |h_j(x^k)| \to 0.$$

Then, Lemmas 3.3 and 3.4 imply

$$g_i(\bar{x}) \le 0 \ \ (i = 1, \cdots, m), \quad h_j(\bar{x}) = 0 \ \ (j = 1, \cdots, l),$$

and hence $\bar{x}$ satisfies the third KKT condition (3.11).

Finally, we show the complementarity condition (3.10). From Lemma 3.1, we have $\lambda_i^k \to 0$. Moreover, since $\{\lambda^k\}$ is bounded, $\lambda_i^k g_i(x^k) \to 0$ for all i such that $g_i(\bar{x}) < 0$ and $\lambda_i^k g_i(x^k) \to 0$ for all i such that $g_i(\bar{x}) = 0$.

From the updating rule of λ_i^k and the condition (3.14), we have either $|\max\{g_i(x^k), -\lambda_i^k\}| < \xi^k$ or $\lambda_i^k = \max\{0, \lambda_i^{k-1} + t_i^{k-1} g_i(x^{k-1})\}$. Thus, $\bar{\lambda}_i \ge 0$ holds. □

Note that this theory includes the case when the sets G_k and H_k are fixed for all iterations.

4 Implementations for large-scale convex programming problems with linear constraints

In this section, we discuss about the concrete implementations of SCM for large-scale programming problems with linear constraints. We assume that the following assumption holds for (1.1).

Assumption 2. *The objective function f is continuously differentiable and convex. Moreover, all of the constraint functions $g_i(x)$ $(i = 1, \cdots, l)$ and $h_j(x)$ $(j = 1, \cdots, m)$ are linear.*

There are many applications that satisfy this assumption, such as the support vector machine and other problems in the machine learning [4, 8], transportation problems [11], and so on. Concrete formulations and other details of the applications are referred to in Appendix B.

4.1 Gradient descent methods for solving the subproblems of the switching constraints multiplier method

Solving the subproblems of SCM efficiently is essential to get a good performance of the total computations of SCM. Here we introduce two types of gradient descent methods, MD and its acceleration method, APG, as solvers for large-scale convex programming problems. We then explain why they are well-suited to SCM.

MD and APG are solvers for the following convex programming problem:

$$\min_{x \in X} \tilde{Q}(x), \tag{4.39}$$

where $\tilde{Q}$ is a differentiable convex function, and X is a simple set.

MD uses a Bregman function $B_\phi : X \times X \mapsto \mathbb{R}$ defined by

$$B_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y \rangle,$$

where ϕ is a continuously differentiable strictly convex function. It is obvious that $B_\phi(x, y) \geq 0$ for all $x, y \in X$, and hence we can regard this function as a distance-like function.

The algorithm of MD for (4.39) is described as follows [1, 7].

The Mirror Descent Method

Set an initial point $x^0 \in X$. Generate a sequence $\{x^k\} \subset X$ by the following iteration.

$$x^{k+1} = \mathrm{argmin}_{x \in X} \left\{ \langle \nabla\tilde{Q}(x^k), x \rangle + \frac{1}{q_k} B_\psi(x, x^k) \right\}, \tag{4.40}$$

where q_k is an appropriate step size.

Next, we introduce APG. The algorithm of APG for (4.39) is described as follows [12].

The Accelerated Proximal Gradient Method

Set initial points $x^0, z^0 \in X$ and $\theta_0 = 1$. Generate the sequence $\{x^k\} \subset X$ by the following rules.

$$\begin{aligned} y^k &= (1-\theta_k)x^k + \theta_k z^k, \\ z^{k+1} &= \arg\min_{x\in X}\left\{\langle \nabla\tilde{Q}(y^k), x\rangle + \frac{\theta_k}{q_k}B_\psi(x, z^k)\right\}, \\ x^{k+1} &= (1-\theta_k)x^k + \theta_k z^{k+1}, \\ \theta_{k+1} &= \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}, \end{aligned} \tag{4.41}$$

where q_k is an appropriate step size.

Here we can assume by scaling if necessary that the Bregman function satisfies

$$B_\psi(x, y) \geq \frac{1}{2}\|x - y\|^2 \quad (\forall x, y \in X). \tag{4.42}$$

Then, we have the following proposition related to the convergence of MD and APG.

Proposition 4.1. *[12] Let $\{x^k\}$ be a sequence generated by MD or APG, and $\hat{x}^*$ be an optimal solution of (4.39). Moreover, suppose that $\nabla\tilde{Q}(x)$ is a Lipschitz continuous function with a Lipschitz constant $L_{\tilde{Q}}$, and the Bregman function B_ψ satisfies (4.42). Suppose also that $q_k = \frac{1}{L_{\tilde{Q}}}$. Then, we have, for MD,*

$$\tilde{Q}(x^k) \leq \tilde{Q}(\hat{x}^*) + \frac{1}{k}L_{\tilde{Q}}B_\psi(\hat{x}^*, x^0) \quad (k = 1, 2, \cdots),$$

and for APG,

$$\tilde{Q}(x^k) \leq \tilde{Q}(\hat{x}^*) + \theta_{k-1}^2 L_{\tilde{Q}}B_\psi(\hat{x}^*, z^0) \quad (k = 1, 2, \cdots).$$

□

Since $\theta_k \leq \frac{2}{k+2}$ for all k, this proposition implies the convergence of MD and APG.

However the Lipschitz constant $L_{\tilde{Q}}$ is not always given in advance. Then, we may use a reasonable step size q_k. It is easy to find an appropriate q_k for MD. In [12], this proposition for APG was proved by using the inequalities

$$\begin{aligned} \tilde{Q}(x^{k+1}) &\leq \ell_{\tilde{Q}}(x^{k+1}, y^k) + \frac{L}{2}\|x^{k+1} - y^k\|^2 \\ &= \ell_{\tilde{Q}}((1-\theta_k)x^k + \theta_k z^{k+1}, y^k) + \frac{L\theta_k^2}{2}\|z^{k+1} - z^k\|^2 \\ &\leq (1-\theta_k)\ell_{\tilde{Q}}(x^k, y^k) + \theta_k \ell_{\tilde{Q}}(z^{k+1}, y^k) + \theta_k^2 L_{\tilde{Q}}B_\psi(z^{k+1}, z^k) \\ &\leq (1-\theta_k)\ell_{\tilde{Q}}(x^k, y^k) + \theta_k\left(\ell_{\tilde{Q}}(x, y^k) + \theta_k L_{\tilde{Q}}B_\psi(x, z^k) - \theta_k L_{\tilde{Q}}B_\psi(x, z^{k+1})\right) \\ &\leq (1-\theta_k)\tilde{Q}(x^k) + \theta_k\tilde{Q}(x) + \theta_k^2 L_{\tilde{Q}}B_\psi(x, z^k) - \theta_k^2 L_{\tilde{Q}}B_\psi(x, z^{k+1}) \\ &\quad (\forall x \in X), \end{aligned}$$

where $\ell_{\tilde{Q}}(x, y)$ was defined by

$$\ell_{\tilde{Q}}(x, y) = \tilde{Q}(y) + \langle \nabla \tilde{Q}(y), x - y \rangle.$$

Then, we can ensure the convergence of APG when these inequalities hold. Therefore, when applying APG, we choose q_k such that

$$\tilde{Q}(x^{k+1}) \leq (1 - \theta_k)\ell_{\tilde{Q}}(x^k, y^k) + \theta_k \ell_{\tilde{Q}}(z^{k+1}, y^k) + \frac{\theta_k}{q_k} B_\psi(z^{k+1}, z^k).$$

4.1.1 Explicit formulas of the subproblems (4.40) and (4.41) for some special linear constraints

When using MD or APG, how to solve the subproblems (4.40) or (4.41) is the key to efficiency.

The next proposition shows that we can explicitly construct a solution of these subproblems of MD and APG when X in the problem (4.39) has some special structures. The concrete solution formulas are presented in Appendix A.

Proposition 4.2. *Suppose that X in the problem (4.39) is formulated as either the following (a), (b) or (c). Then, the solution of (4.40) and (4.41) with (a), (b) or (c) can be calculated in $O(n)$, $O(\Gamma n)$ or $O(mn)$ steps, respectively by MD or APG.*

(a) Upper and Lower constraints.

$$X := \{x \in \mathbb{R}^n \mid l_i \leq x_i \leq u_i \ \ (i = 1, \ldots, n)\}. \tag{4.43}$$

(b) Lower bound constraints and separated linear equality constraints.

$$X := X_1 \times \cdots \times X_\Gamma \tag{4.44}$$

with $\sum_{\gamma=1}^{\Gamma} n_\gamma = n$ and

$$X_\gamma := \left\{ x^\gamma \in \mathbb{R}^{n_\gamma} \;\middle|\; \begin{array}{l} \sum_{i \in P^\gamma} x_i^\gamma - \sum_{j \in N^\gamma} x_j^\gamma = d^\gamma, \\ x_i^\gamma \geq 0 \quad (i = 1, \ldots, n_\gamma) \end{array} \right\},$$

where the sets P^γ and N^γ are a partitions of $\{1, \cdots, n_\gamma\}$, that is,

$$P^\gamma \cap N^\gamma = \emptyset, \ \ P^\gamma \cup N^\gamma = \{1, \cdots, n_\gamma\}.$$

(c) Linear equality constraints.

$$X := \{x \in \mathbb{R}^n \mid Ax = b\}, \tag{4.45}$$

where $m < n$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Moreover, A can be decomposed into a nonsingular matrix $A_{\mathrm{F}} \in \mathbb{R}^{m \times m}$ and a matrix $A_{\mathrm{L}} \in \mathbb{R}^{m \times (n-m)}$, i.e., $A = \begin{bmatrix} A_{\mathrm{F}} & A_{\mathrm{L}} \end{bmatrix}$.

□

Note that some linear constraints can be regarded as (a) or (b).

(a') Lower bound constraints. By setting $u_i = \infty$, (a) can be transformed to the following lower bound constraints.

$$X := \{x \in \mathbb{R}^n \mid x_i \geq l_i \ \ (i = 1, \ldots, n)\}.$$

(b') Unit simplex constraint. We can regard the following unit simplex constraint as a special case of (b).

The set $\{1, 2, \ldots, n\}$ is decomposed into m sets $I_1, I_2, \ldots, I_m$, and

$$X := \Delta_1 \times \Delta_2 \times \cdots \times \Delta_m,$$

where Δ_j is a unit simplex defined by

$$\Delta_j = \{x \in R^{|I_j|} \mid x_j \geq 0, \textstyle\sum_{i=1}^{|I_j|} x_i = 1\}.$$

(b") Lower bound constraints and a single linear equality constraint. A general equality constraint with the lower bound constraints can be reduced into (b).

$$X := \{x \in \mathbb{R}^n \mid a^\top x = b, \ \ x_i \geq 0 \ \ (i = 1, \ldots, n)\}.$$

To see this, we reformulate the problem (4.39) with

$$\dot{x}_i := \begin{cases} |a_i| x_i & (a_i \neq 0) \\ x_i & (a_i = 0). \end{cases}$$

Then, the problem (4.39) can be write as

$$\begin{array}{ll} \min & \tilde{Q}(\dot{x}) \\ \text{s.t.} & \sum_{i \in S} \dot{x}_i - \sum_{j \in T} \dot{x}_j = d \\ & \dot{x}_i \geq 0, \ \ (i = 1, \ldots, n), \end{array}$$

where

$$S = \{i \mid a_i \geq 0\}, \ \ T = \{i \mid a_i < 0\}.$$

The reformulated problem has the constraint (b).

Now, we discuss why MD and APG is suitable for solving the subproblem of SCM. SCM can be used for the problem with many constraints. The advantage of SCM is that we can choose the remaining constraints at each iteration. On the other hand, the iterates of MD and APG can be represented explicitly for several particular constraints mentioned above. Thus, we can make use of both advantages by using MD or APG for solving the subproblems of SCM. That is to say, when we may choose the remaining constraints as the constraint given in Proposition 4.2, we can solve large-scale subproblems of SCM by MD or APG.

4.1.2 Calculation of $\bar{\lambda}$ and $\bar{\mu}$ in the mirror descent method and the accelerated proximal gradient method

SCM exploits the optimal Lagrange multiplier $(\bar{\lambda}^k, \bar{\mu}^k)$ of the remaining constraints as an estimated Lagrange multiplier. Note that the optimal Lagrange multiplier of the subproblem (4.40) of MD converges to the optimal Lagrange

multiplier of the original problem (4.39). Thus, we may use the optimal Lagrange multiplier of (4.40) as $(\bar{\lambda}^k, \bar{\mu}^k)$ in Step 2 of SCM. However, since z^k obtained in (4.41) of APG does not always converge to an optimal solution of (4.39), we cannot exploit the optimal Lagrange multiplier of (4.41) as $(\bar{\lambda}^k, \bar{\mu}^k)$ in SCM. However, APG is still useful for SCM when the subproblem has the special constraints given in Proposition 4.2. We can estimate the optimal Lagrange multipliers of the subproblem (3.13) of SCM from the approximate solution x^k obtained by APG.

We give the concrete calculations of the Lagrange multipliers. Note that it is difficult to obtain the exact optimal solution of (3.13). Therefore, in actual fact, the Lagrange multipliers obtained as follows are also approximate Lagrange multipliers.

(a) Upper and Lower bound constraints. Suppose that X is given by (4.43) and $\acute{x}$ is a solution of (4.39). Then, there exists a vector λ such that

$$\begin{array}{ll} \nabla\tilde{Q}(\acute{x})_i - \lambda_i + \lambda_{i+n} = 0 & (i = 1, \cdots, n), \\ \lambda_i \geq 0, \lambda_i(\acute{x}_i - l_i) = 0 & (i = 1, \cdots, n), \\ \lambda_{i+n} \geq 0, \lambda_{i+n}(\acute{x}_i - u_i) = 0 & (i = 1, \cdots, n). \end{array}$$

Thus, λ can be written as

$$\lambda_i = \max\{0, \nabla\tilde{Q}(\acute{x})_i\}, \quad \lambda_{i+n} = \max\{0, -\nabla\tilde{Q}(\acute{x})_i\} \quad (i = 1, \cdots, n).$$

Therefore, we may adopt the following estimated Lagrange multipliers in SCM.

$$\bar{\lambda}_i^k = \max\{0, \nabla Q_k(x^k)_i\}, \quad \bar{\lambda}_{i+n}^k = \max\{0, -\nabla Q_k(x^k)_i\} \quad (i = 1, \cdots, n).$$

(b) Lower bound constraints and separated linear equality constraints. Suppose that X is given by (4.44) and $\acute{x} = (\acute{x}^1, \cdots, \acute{x}^\Gamma) \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_\Gamma}$ is a solution of (4.39). Then, for each $\gamma = 1, \cdots, \Gamma$, there exists vectors λ^γ and μ^γ such that

$$\begin{array}{ll} \nabla_{x^\gamma}\tilde{Q}(\acute{x})_i + \mu^\gamma - \lambda_i^\gamma = 0 & (i \in P^\gamma), \\ \nabla_{x^\gamma}\tilde{Q}(\acute{x})_i - \mu^\gamma - \lambda_i^\gamma = 0 & (i \in N^\gamma), \\ \sum_{i \in P^\gamma} \acute{x}_i^\gamma - \sum_{j \in N^\gamma} \acute{x}_j^\gamma = d^\gamma, & \\ \lambda_i^\gamma \geq 0, \quad \lambda_i^\gamma \acute{x}_i^\gamma = 0 & (i = 1, \cdots, n_\gamma). \end{array}$$

Case (i) $d^\gamma \neq 0$: Then, from the third and the last equality, there exists an index i' such that $\acute{x}_{i'}^\gamma \neq 0$ and $\lambda_{i'}^\gamma = 0$. For such i', we have $\mu^\gamma = -\nabla_{x^\gamma}\tilde{Q}(\acute{x})_{i'}$. Moreover, λ^γ is determined as follows.

$$\lambda_i^\gamma = \begin{cases} \nabla_{x^\gamma}\tilde{Q}(\acute{x})_i - \nabla_{x^\gamma}\tilde{Q}(\acute{x})_{i'} & (i \in P^\gamma) \\ \nabla_{x^\gamma}\tilde{Q}(\acute{x})_i + \nabla_{x^\gamma}\tilde{Q}(\acute{x})_{i'} & (i \in N^\gamma). \end{cases}$$

Case (ii) $d^\gamma = 0$: When there exists an index i' such that $\acute{x}_{i'}^\gamma \neq 0$, then we set λ^γ and μ^γ as Case (i). Thus, we consider only the case when $\acute{x}_i^\gamma = 0$ for all i. Then, we set the minimum μ such that

$$\begin{array}{ll} \nabla_{x^\gamma}\tilde{Q}(\acute{x})_i + \mu \geq 0 & (i \in P^\gamma), \\ \nabla_{x^\gamma}\tilde{Q}(\acute{x})_i - \mu \geq 0 & (i \in N^\gamma) \end{array}$$

as μ^γ. Moreover, λ^γ can be written as follows.

$$\lambda_i^\gamma = \begin{cases} \nabla_{x^\gamma}\tilde{Q}(\acute{x})_i + \mu^\gamma & (i \in P^\gamma) \\ \nabla_{x^\gamma}\tilde{Q}(\acute{x})_i - \mu^\gamma & (i \in N^\gamma). \end{cases}$$

Therefore, in SCM, we may adopt the estimated Lagrange multiplier $\bar{\mu^\gamma}^k$ as the minimum μ such that

$$\begin{aligned} \nabla_{x^\gamma} Q_k(x^k)_i + \mu \geq 0 \quad (i \in P^\gamma), \\ \nabla_{x^\gamma} Q_k(x^k)_i - \mu \geq 0 \quad (i \in N^\gamma), \end{aligned}$$

and

$$\bar{\lambda^\gamma}_i^k = \begin{cases} \nabla_{x^\gamma} Q_k(x^k)_i + \bar{\mu^\gamma}^k & (i \in P^\gamma) \\ \nabla_{x^\gamma} Q_k(x^k)_i - \bar{\mu^\gamma}^k & (i \in N^\gamma). \end{cases}$$

(c) Linear equality constraints. Suppose that X is given by (4.45) and x is decomposed into $x_{\mathrm{F}} \in \mathbb{R}^m$ and $x_{\mathrm{L}} \in \mathbb{R}^{n-m}$, i.e., $x = (x_{\mathrm{F}}, x_{\mathrm{L}})$. Suppose also that $\acute{x} = (\acute{x}_{\mathrm{F}}, \acute{x}_{\mathrm{L}})$ is a solution of (4.39). Then, there exists a μ such that

$$\begin{pmatrix} \nabla_{x_{\mathrm{F}}}\tilde{Q}(\acute{x}) \\ \nabla_{x_{\mathrm{L}}}\tilde{Q}(\acute{x}) \end{pmatrix} + \begin{pmatrix} A_{\mathrm{F}}^\top \\ A_{\mathrm{L}}^\top \end{pmatrix} \mu = 0. \tag{4.46}$$

Moreover, since A_{F} is nonsingular and $A_{\mathrm{F}}\acute{x}_{\mathrm{F}} + A_{\mathrm{L}}\acute{x}_{\mathrm{L}} = b$, we have $\acute{x}_{\mathrm{F}} = A_{\mathrm{F}}^{-1}b - A_{\mathrm{F}}^{-1}A_{\mathrm{L}}\acute{x}_{\mathrm{L}}$. Then, the problem (4.39) can be transformed to the following $(n-m)$-dimensional unconstrained minimizing problem.

$$\min_{x_{\mathrm{L}} \in \mathbb{R}^{n-m}} F(x_{\mathrm{L}}) := \tilde{Q}(A_{\mathrm{F}}^{-1}b - A_{\mathrm{F}}^{-1}A_{\mathrm{L}}x_{\mathrm{L}}, x_{\mathrm{L}}). \tag{4.47}$$

Note that $\nabla F(x_{\mathrm{L}}) = -\left(A_{\mathrm{F}}^{-1}A_{\mathrm{L}}\right)^\top \nabla_{x_{\mathrm{L}}}\tilde{Q}(\acute{x}) + \nabla_{x_{\mathrm{L}}}\tilde{Q}(\acute{x}) = 0$. By setting $\mu = -(A_{\mathrm{F}}^{-1})^\top \nabla_{x_{\mathrm{F}}}\tilde{Q}(\acute{x})$, (4.46) holds.

Therefore, we may adopt the following estimated Lagrange multipliers in SCM.

$$\bar{\mu}^k = -\left(A_{\mathrm{F}}^{-1}\right)^\top \nabla_{x_{\mathrm{F}}} Q_k(x^k).$$

4.2 Heuristic techniques for choosing G_k and H_k

The significance of switching constraints in each iteration is not only exploiting the Lagrange multipliers of the subproblems but also changing constraints to which the subproblem has good properties at each iteration. Here, we consider which constraints to choose as the remaining constraints at each iteration.

First, we consider from a feasibility point of view. Since the feasibility of the remaining constraints will be satisfied after solving the subproblem, we should choose constraints with large violations at the current point in each iteration as the remaining constraints. Then, we can expect that SCM converges within the smaller number of iterations. Moreover, we can also expect that this choosing idea will prevent the penalty parameters from increasing. Thus, we consider choosing the constraints whose violations $\left|\max\left\{g_i(x^k), -\lambda_i^k\right\}\right|$ and $\left|h_j(x^k)\right|$ are relatively large, that is, we should choose G_k and H_k such that

$$\min_{i,j \in G_k \cup H_k} \Delta_{i,j}^k \geq \max_{i,j \notin G_k \cup H_k} \Delta_{i,j}^k, \tag{4.48}$$

where $\Delta_{i,j}^k := \{|\max\{g_i(x^{k-1}), -\lambda_i^{k-1}\}|, |h_j(x^{k-1})|\}$.

Next, we consider the case where we adopt MD or APG as the solver for the subproblems (3.13) of SCM. We can reduce calculations when we choose the remaining constraints formed as given in Proposition 4.2. Therefore, we should choose constraints with relatively large violations as the remaining constraints so that they form (a), (b) or (c) (or their derivations (a'), (b') or (b")) of Proposition 4.2.

5 Numerical experiments

In this section, some numerical results are reported. We conducted two experiments in Matlab 7.1. In both experiments, we solved the following convex programming problem with SCM:

$$\begin{array}{lll} \min\limits_{x} & \frac{1}{2}x^\top Kx & \\ \text{s.t.} & -x_i \le 0 & (i = 1, \cdots, n) \\ & x_i \le \frac{1}{n} & (i = 1, \cdots, n) \\ & \sum_{i=1}^n \delta_i x_i = 0 & \\ & \sum_{i=1}^n x_i - \nu = 0, & \end{array} \tag{5.49}$$

where $K \in \mathbb{R}^{n\times n}$ is a positive definite matrix generated randomly, $\nu = 0.5$ and δ_i is given as

$$\delta_i = \begin{cases} 1 & (i = 1, 3, \cdots, \frac{n}{2}+1) \\ -1 & (i = 2, \frac{n}{2}+1, \cdots, n). \end{cases}$$

The problem (5.49) is the same as the dual problem of ν-SVM in a field of machine learning.

For convenience, let $g_i(x) = -x_i$ $(n = 1\cdots, n)$, $g_{i+n}(x) = x_i - \frac{1}{n}$ $(i = 1, \cdots, n)$, $h_1(x) = \sum_{i=1}^n \delta_i x_i$, $h_2(x) = \sum_{i=1}^n x_i - \nu$. Then, the constraints of (5.49) can be written as

$$\begin{array}{lll} g_i(x) \le 0 & (i = 1, \cdots, n), & \text{(C1)} \\ g_{i+n}(x) \le 0 & (i = 1, \cdots, n), & \text{(C2)} \\ h_1(x) = 0, & & \text{(C3)} \\ h_2(x) = 0. & & \text{(C4)} \end{array}$$

5.1 Comparison with the existing multiplier method

The first experiment was conducted for comparing the proposed SCM with the conventional multiplier method.

In both SCM and the conventional multiplier method, the dimension n is 300, and the number of remaining constraints is fixed to 60 for all iterations. That is, the main problem (5.49) has 300-dimensional problem with 602 constraints, and each subproblem has always 60 remaining constraints in both SCM and the conventional multiplier method. The difference between these two methods is that SCM switches the remaining constraints G_k and H_k at each iteration by the rule (4.48) while the conventional multiplier method has always the fixed remaining constraints that are chosen randomly in advance. In addition, SCM updates the estimated Lagrange multipliers by (3.15) while the conventional multiplier method updates them by (2.6) and (2.8). In the first experiment,

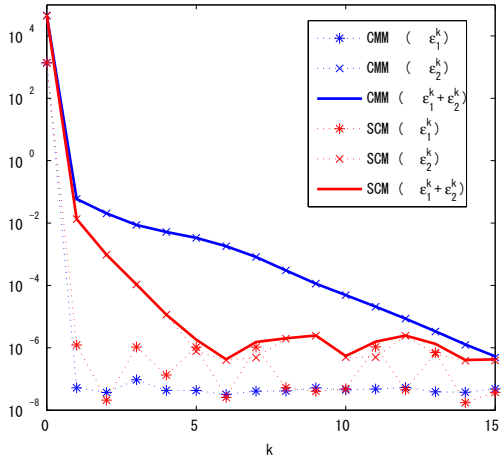

Figure 1: Residuals of KKT conditions

each subproblem (3.13) was solved by the command $fmincon$ in Matlab, that is, it is solved very accurately. The results are described in Figures 1-3. Note that CMM in each figure stands for the conventional multiplier method.

Figure 1 illustrates the residuals $\epsilon_1^k, \epsilon_2^k$ and $\epsilon_1^k+\epsilon_2^k$ of KKT conditions defined by

$$\epsilon_1^k := \frac{1}{n}\left\|\nabla f(x^k) + \sum_{i=1}^{l}\lambda_i^k \nabla g_i(x^k) + \sum_{j=1}^{m}\mu_j^k \nabla h_j(x^k)\right\|,$$

$$\epsilon_2^k := \frac{1}{l}\sum_{i=1}^{l}\left|\max\{g_i(x^k), -\lambda_i^k\}\right| + \frac{1}{m}\sum_{j=1}^{m}\left|h_j(x^k)\right|.$$

We see that SCM converges within the smaller number of iteration than the conventional multiplier method.

Figure 2 illustrates the residual ϵ_3^k of the obtained Lagrange multiplier defined by

$$\epsilon_3^k := \frac{1}{l}\|\lambda^k - \lambda^*\| + \frac{1}{m}\|\mu^k - \mu^*\|.$$

We see that the Lagrange multipliers obtained by SCM are closer to the optimal Lagrange multipliers of (5.49) than those obtained by the conventional multiplier method. From Figures 1 and 2, we see that the convergence of the multiplier method depends on the quality of the estimated Lagrange multipliers.

Figure 3 illustrates the total amount $\sum_{i=1}^{l} t_i^k + \sum_{j=1}^{m} r_j^k$ of the penalty parameters. We see that the choosing technique (4.48) prevents the penalty parameters from increasing. Therefore, we can expect that SCM is more stable than the conventional multiplier method.

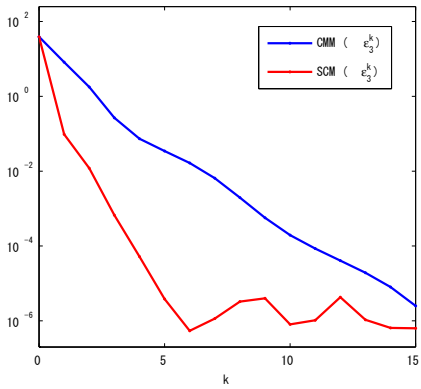

Figure 2: Residuals of Lagrange multipliers

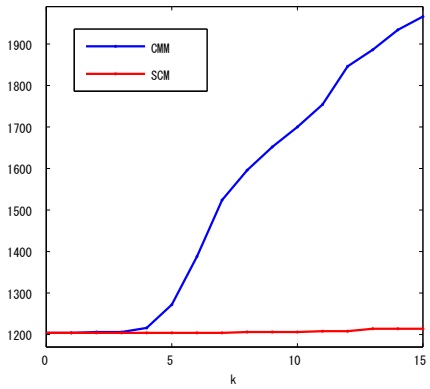

Figure 3: Increasing process of penalty parameters

5.2 The switching constraints multiplier method with the mirror descent method

As the second experiment, we investigate SCM with MD. As mentioned in Section 4, it is to be desired that the remaining constraints G_k and H_k have the form (a), (b) or (c) (or their derivations (a'), (b') or (b")) in Proposition 4.2. For the problem (5.49), we have the following four choices to make each subproblem suitable for MD.

- Constraints (C1) and (C2) form (a). Then, $G_k = \{1, \cdots, 2n\}$ and $H_k = \emptyset$. Note that it is not necessary to choose all of the constraints in (C2).
- Constraints (C1) form (a'). Then, $G_k = \{1, \cdots, n\}$ and $H_k = \emptyset$.
- Constraints (C1) and (C3), or constraints (C1) and (C4) form (b). Then, $G_k = \emptyset$ and $H_k = \{1\}$ or $H_k = \{2\}$.
- Constraints (C3) and (C4) form (c). Then, $G_k = \emptyset$ and $H_k = \{1, 2\}$.

We investigate the behaviors of SCM with MD for the following three cases.

Case 1. (C1) and (C2) are the remaining constraints for all iterations, that is, $G_k = \{1, \cdots, 2n\}$ and $H_k = \emptyset$.

Case 2. (C3) and (C4) are the remaining constraints for iterations, that is, $G_k = \emptyset$ and $H_k = \{1, 2\}$.

Case 3. We mix the above two cases, that is, we choose the constraints as Case 1 and Case 2 by turns, that is, $G_k = \{1, \cdots, 2n\}$ and $H_k = \emptyset$ if k is odd, and, $G_k = \emptyset$ and $H_k = \{1, 2\}$ if k is even.

The problems with various dimensions, i.e., $n = 500, 1000, 2000, 3000, 4000$ were solved by SCM with MD. We adopted the termination criterion $\epsilon = 10^{-2}$.

We adopted the following termination criterion of the each subproblem:

$$\epsilon_\Delta^k < 0.9\epsilon_\Delta^{k-1},$$

Table 1: The number of iterations by SCM with MD

	Case 1		Case 2		Case 3	
Dimension (n)	# of Outer iterations	# of Total iterations	# of Outer iterations	# of Total iterations	# of Outer iterations	# of Total iterations
500	23.8	694.4	19.8	558.8	4.0	472.4
1000	18.2	1409.0	19.0	990.8	4.0	528.6
2000	20.0	2347.0	19.6	2095.0	4.0	945.8
3000	20.2	2926.0	21.8	3220.6	4.0	1191.4
4000	21.2	2452.4	23.2	4937.8	4.0	1395.8

where ϵ_{Δ}^{k} denotes the KKT residual of the subproblem (3.13) at the kth iteration. We set the number of iterations of the subproblem at the first iteration of SCM as 100.

The results are listed in Table 1. Table 1 lists the number of iterations until the termination criterion holds. Each result listed in Table 1 is the average value of five attempts. Note that there are two types of iteration in Table 1, that is, one is the outer iteration of SCM, and the other includes the inner iteration (MD iteration).

We see that SCM with Case 3 converges more efficiently than those with Cases 1 and 2. This result implies that the Lagrange multipliers of the subproblem are more accurate than those of calculated by (2.6) and (2.8) only.

6 Concluding remarks

In this paper, we proposed a novel multiplier method. The proposed method has the following advantages.

- By exploiting the Lagrange multipliers of the subproblems, the proposed method can approximate the optimal Lagrange multipliers more efficiently than the existing multiplier method, which results in the efficient convergence.
- By choosing the remaining constraints optionally at each iteration, we can prevent the penalty parameters from diverging. Therefore, we can avoid the numerical difficulty.

On the other hand, the behaviors of SCM must be investigated for many more problems with more complex constraints arisen in practical situation. Moreover, although we gave only one idea of choosing the remaining constraints in this paper, we can suggest other ideas. For example, when using MD or APG as the algorithm to solve the subproblems of SCM, we can take into consideration the fact that the smaller the Lipschitz constant of the objective function is, the faster MD or APG converges. Therefore, we can expect that the subproblem will converge within the smaller number of iterations when we choose the remaining constraints such that the augmented Lagrangian function has a smaller Lipschitz constant.

Acknowledgements

I would like to express my sincere appreciation to Associate Professor Nobuo Yamashita for his kind guidance, suggestion and support. I could have never written up this paper without them. As well, I am deeply indebted to Professor Masao Fukushima for giving me countless useful comments on my research, and Assistant Professor Shunsuke Hayashi not only for his teaching but for his good management of Fukushima Laboratory. In addition, I would like to thank every member of this laboratory. They encouraged and entertained me inside and outside the laboratory. Finally, special thanks are due to my family for their constant support.

Appendix A

The concrete solution formulas of the problem (4.40) and (4.41) with the constraint given in Proposition 4.2.

(a) Upper and Lower bound constraints. Suppose that X is given by (4.43). Then, the iteration (4.41) of APG can be represented as

$$z_i^{k+1} = \mathrm{mid}\left\{l_i, z_i^k - \frac{q_k}{\theta_k}\nabla\tilde{Q}(y^k)_i, u_i\right\}.$$

Similarly, the iteration (4.40) of MD can be represented, by substituting y^k, z^k and θ_k for x^k, x^k and 1, respectively, as

$$x_i^{k+1} = \mathrm{mid}\left\{l_i, x_i^k - q_k\nabla\tilde{Q}(x^k)_i, u_i\right\}.$$

(b) Lower bound constraints and separated linear equality constraints. For simplicity, we consider the following constraints as X in the problem (4.39) instead of (4.44).

$$X := \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \sum_{i\in P} x_i - \sum_{j\in N} x_j = d, \\ x_i \geq 0 \quad (i = 1, \ldots, n) \end{array} \right\}.$$

For the case when the problem has nonnegative constraints, the following function, called *entropy function*, has been proposed as ψ in the definition of Bregman function [1]. The entropy function is defined by

$$\psi(x) = \sum_{i=1} x_i \ln x_i,$$

where we adopt the convention $0 \ln 0 := 0$. We use the Bregman function with the entropy function. Then, we obtain

$$\begin{aligned} B_\psi(x, y) &= \sum_{i=1}^{n} (x_i \ln x_i - y_i \ln y_i + (1 + \ln y_i)(y_i - x_i)) \\ &= \sum_{i=1}^{n} \left(x_i \ln \frac{x_i}{y_i} - x_i + y_i \right). \end{aligned}$$

Thus, we have

$$\nabla_x B_\psi(x,y) = \begin{pmatrix} \vdots \\ 1 + \ln x_i - \ln y_i - 1 \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \ln \frac{x_i}{y_i} \\ \vdots \end{pmatrix}.$$

Then, KKT conditions of the subproblem (4.41) of APG can be written as

$$\begin{aligned} &\nabla \tilde{Q}(y^k) + \frac{\theta_k}{q_k} \ln \frac{z_i^{k+1}}{z_i^k} + \mu = 0 \quad (i \in P), \\ &\nabla \tilde{Q}(y^k) + \frac{\theta_k}{q_k} \ln \frac{z_i^{k+1}}{z_i^k} - \mu = 0 \quad (i \in N), \\ &\sum_{i \in P} z_i^{k+1} - \sum_{j \in N} z_j^{k+1} = d. \end{aligned}$$

The first and the second equality equation yield

$$z_i^{k+1} = \begin{cases} z_i^k \exp\left(-\frac{q_k}{\theta_k} \nabla \tilde{Q}(y^k)_i\right) \exp\left(-\frac{q_k}{\theta_k}\mu\right) & (i \in P) \\ z_i^k \exp\left(-\frac{q_k}{\theta_k} \nabla \tilde{Q}(y^k)_i\right) \exp\left(\frac{q_k}{\theta_k}\mu\right) & (i \in N). \end{cases}$$

By substituting them for the third equation, we obtain

$$\exp\left(-\frac{q_k}{\theta_k}\mu\right) \sum_{i \in P} z_i^k \exp\left(-\frac{q_k}{\theta_k} \nabla \tilde{Q}(y^k)_i\right) - \exp\left(\frac{q_k}{\theta_k}\mu\right) \sum_{j \in N} z_j^k \exp\left(-\frac{q_k}{\theta_k} \nabla \tilde{Q}(y^k)_j\right) = d.$$

By setting $Z_1 := \sum_{i \in P} z_i^k \exp(-\frac{q_k}{\theta_k} \nabla \tilde{Q}(y^k)_i)$, $Z_2 := \sum_{j \in N} z_j^k \exp(-\frac{q_k}{\theta_k} \nabla \tilde{Q}(y^k)_j)$ and $\pi := \exp(\frac{q_k}{\theta_k}\mu)$, this equation can be written as

$$Z_2 \pi^2 + d\pi - Z_1 = 0.$$

Note that $\pi > 0$. Then the solution of this equation π^* is given as follows.[1]

$$\pi^* = \frac{-d + \sqrt{d^2 + 4Z_1 Z_2}}{2Z_2}.$$

Therefore we obtain $\mu = \frac{\theta_k}{q_k} \ln \pi^*$, and hence

$$z_i^{k+1} = \begin{cases} \frac{z_i^k}{\pi^*} \exp\left(-\frac{q_k}{\theta_k} \nabla \tilde{Q}(y^k)_i\right) & (i \in P) \\ z_i^{k+1} = z_i^k \pi^* \exp\left(-\frac{q_k}{\theta_k} \nabla \tilde{Q}(y^k)_i\right) & (i \in N). \end{cases}$$

Similarly, the iteration (4.40) of MD can be represented, by substituting y^k, z^k and θ_k for x^k, x^k and 1, respectively, as

$$x_i^{k+1} = \begin{cases} \frac{x_i^k}{\pi^*} \exp\left(-q_k \nabla \tilde{Q}(x^k)_i\right) & (i \in P) \\ \pi^* x_i^k \exp\left(-q_k \nabla \tilde{Q}(x^k)_i\right) & (i \in N). \end{cases}$$

[1] Since $Z_1, Z_2 > 0$, we have $-d - \sqrt{d^2 + 4Z_1 Z_2} < 0$, which implies that the positive root is unique.

(c) Linear equality constraints. Suppose that X is given by (4.45). As mentioned in Section 4.1.2, since the problem (4.39) can be described as the nonconstrained problem (4.47). Then, we adopt the following Bregman function with $\psi(x) = \frac{1}{2}\|x\|^2$.

$$B_\psi(x, y) = \frac{1}{2}\|x - y\|^2.$$

Then, the KKT condition of the subproblem (4.41) of APG can be written as

$$\nabla F(y_{\mathrm{L}}^k) + \frac{\theta_k}{q_k}(z_{\mathrm{L}}^{k+1} - z_{\mathrm{L}}^k) = 0,$$

where F is given in (4.47). Therefore, we have

$$z^{k+1} = \begin{pmatrix} z_{\mathrm{F}}^{k+1} \\ z_{\mathrm{L}}^{k+1} \end{pmatrix} = \begin{cases} A_{\mathrm{F}}^{-1}b - A_{\mathrm{F}}^{-1}A_{\mathrm{L}}z_{\mathrm{L}}^k \\ -\frac{q_k}{\theta_k}\nabla F(y_{\mathrm{L}}^k) + z_{\mathrm{L}}^k. \end{cases}$$

Similarly, the iteration (4.40) of MD can be represented, by substituting y^k, z^k and θ_k for x^k, x^k and 1, respectively, as

$$x^{k+1} = \begin{pmatrix} x_{\mathrm{F}}^{k+1} \\ x_{\mathrm{L}}^{k+1} \end{pmatrix} = \begin{cases} A_{\mathrm{F}}^{-1}b - A_{\mathrm{F}}^{-1}A_{\mathrm{L}}x_{\mathrm{L}}^k \\ -q_k\nabla F(x_{\mathrm{L}}^k) + x_{\mathrm{L}}^k. \end{cases}$$

Appendix B : Applications

In this paper, we assume that SCM with MD or APG will be applied to large-scale convex programming problems such as the problems in the field of machine learning [4, 8]. We introduce some examples where SCM with MD or APG is applicable.

Support Vector Machine (SVM) : The dual problem of the hard margin SVM can be written as

$$\begin{array}{ll} \min_\alpha & \frac{1}{2}\alpha^T K\alpha - \sum_{i=1}^T \alpha_i \\ \text{s.t.} & \sum_{i=1}^T y_i\alpha_i = 0 \\ & 0 \le \alpha_i \ \ (i = 1, \cdots, T), \end{array}$$

where $y_i \in \{-1, 1\}$. These constraints correspond to (b) of Proposition 4.2. Thus, the iteration of each subproblem of MD or APG is explicitly given.

$C-$SVM : The dual problem of the soft margin SVM can be written as

$$\begin{array}{ll} \min_\alpha & \frac{1}{2}\alpha^T K\alpha - \sum_{i=1}^T \alpha_i \\ \text{s.t.} & \sum_{i=1}^T y_i\alpha_i = 0 \\ & 0 \le \alpha_i \le C \ \ (i = 1, \cdots, T), \end{array}$$

where $y_i \in \{-1, 1\}$. Since the constraints do not form any of Proposition 4.2, it is impossible to represent the iteration explicitly by MD or APG. However, if we choose remaining constraints as (a), (b), (c) or (a') of Proposition 4.2, we can solve the subproblems efficiently by applying SCM with MD or APG.

$\nu-$SVM： The dual problem of the $\nu-$SVM can be written as

$$\begin{array}{ll} \min_\alpha & \frac{1}{2}\alpha^T K\alpha \\ \text{s.t.} & \sum_{i=1}^T y_i\alpha_i = 0 \\ & \sum_{i=1}^T \alpha_i \geq \nu \\ & 0 \leq \alpha_i \leq \frac{1}{T} \ \ (i = 1, \cdots, T), \end{array}$$

where $y_i \in \{-1, 1\}$. Note that this model is meaningful only when the second inequality holds with equality. [2] Thus, we consider the following problem.

$$(\nu - \text{SVM}') \quad \begin{array}{ll} \min_\alpha & \frac{1}{2}\alpha^T K\alpha \\ \text{s.t.} & \sum_{i=1}^T y_i\alpha_i = 0 \\ & \sum_{i=1}^T \alpha_i = \nu \\ & 0 \leq \alpha_i \leq \frac{1}{T} \ \ (i = 1, \cdots, T). \end{array}$$

By defining two index sets $P = \{i \mid y_i = 1\}$ and $N = \{j \mid y_j = -1\}$, we can rewrite the first and the second constraints of $\nu-$SVM' as

$$\textstyle\sum_{i\in P} \alpha_i - \sum_{j\in N} \alpha_j = 0, \quad \sum_{i\in P} \alpha_i + \sum_{j\in N} \alpha_j = \nu,$$

respectively, which are equivalent to

$$\textstyle\sum_{i\in P} \alpha_i = \frac{\nu}{2}, \ \ \sum_{j\in N} \alpha_j = \frac{\nu}{2}.$$

Therefore, $\nu-$SVM can be write as

$$\begin{array}{ll} \min_\alpha & \frac{1}{2}\alpha^T K\alpha \\ \text{s.t.} & \sum_{i\in P} \alpha_i = \frac{\nu}{2} \\ & \sum_{j\in N} \alpha_j = \frac{\nu}{2} \\ & 0 \leq \alpha_i \leq \frac{1}{T} \ \ (i = 1, \cdots, T). \end{array}$$

Unfortunately, it is impossible to represent the iteration explicitly by MD or APG for this problem. However, we can choose remaining constraints as (a), (b), (c) or (a') of Proposition 4.2 and solve by applying SCM with MD or APG.

Support Vector Regression (SVR): The dual problem of the SVR can be written as

$$\begin{array}{ll} \min_{\alpha,\beta} & \frac{1}{2}(\alpha-\beta)^\top K(\alpha-\beta) + \varepsilon\sum_{i=1}^T(\alpha_i+\beta_i) + \sum_{i=1}^T(\alpha_i-\beta_i) \\ \text{s.t.} & \sum_{i=1}^T \alpha_i - \sum_{i=1}^T \beta_i = 0 \\ & 0 \leq \alpha_i \leq \frac{1}{C} \ \ (i = 1, \cdots, T) \\ & 0 \leq \beta_i \leq \frac{1}{C} \ \ (i = 1, \cdots, T). \end{array}$$

In a way similar to $C-$SVM and $\nu-$SVM, we can solve explicitly by applying SCM with MD or APG.

[2] $\nu-$SVM does not make sense if the second constraint is not active at the original problem, that is, the Lagrange multiplier corresponding to the second constraint is negative at the KKT point of $\nu-$SVM'.

Hitchcock Transportation Problem: The hitchcock transportation problem can be defined as follows [11].

$$\begin{array}{lll} \min_x & \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} & \\ \text{s.t.} & \sum_{j\in J} x_{ij} = a_i & (i \in I) \\ & \sum_{i\in I} x_{ij} = b_j & (j \in J) \\ & x_{ij} \geq 0 & (i \in I,\ j \in J). \end{array}$$

For this problem, we may choose remaining constraints as (a), (b), (c) or (b') of Proposition 4.2 and apply MD or APG.

References

[1] A. Beck and M. Teboulle, *Mirror descent and nonlinear projected subgradient methods for convex optimization*, Operations Research Letters, Vol. 31, pp. 167-175, 2003.

[2] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences, Vol. 2, pp. 183-202, 2009.

[3] P. Bartlett, M. Collins, B. Taskar, and D. McAllester, *Exponentiated gradient algorithms for large-margin structured classification*, In NIPS, 2004.

[4] C. J. C. Burges, *A tutorial on support vector machines for pattern recognition*, Knowledge Discovery and Data Mining, Vol. 2, pp. 121-167, 1998.

[5] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1999.

[6] A. Ben-Tal and M. Zibulevski, *Penalty/Barrier multiplier methods for convex programming problems*, SIAM Journal on Optimization, Vol. 7, pp. 347-366, 1997.

[7] A. Ben-Tal and T. Margalit and A. Nemirovski, *The ordered subsets mirror descent optimization method with applications to tomography*, SIAM Journal on Optimization, Vol. 12, pp. 79-108, 2001.

[8] N. Cristianini and J. Shawe-Taylor, *Support vector machines and other kernel-based learning methods*, Cambridge University Press, 2000.

[9] M. R. Hestenes, *Multiplier and gradient methods*, Journal of Optimization Theory and Applications, Vol. 4, pp. 303-320, 1969.

[10] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, Optimization, R. Fletcher, ed., Academic Press, New York, pp. 283-298, 1969.

[11] M. Sun, J. Aronson, P. Mckeown and M. Drinka, *A tabu search heuristic procedure for the fixed charge transportation problem*, European Journal of Operation Research, Vol. 106, pp. 441-456, 1998.

[12] P. Tseng, *Approximation accuracy gradient methods, and error bound for structured convex optimization*, Mathematical Programming, Ser. B, Vol. 125, pp. 263-295, 2010.