J. Appl. Numer. Optim. 6 (2024), No. 1, pp. 59-70
Available online at http://jano.biemdas.com
https://doi.org/10.23952/jano.6.2024.1.03

NUMERICAL COMPUTATION OF ENTROPY-REGULARIZED QUADRATIC OPTIMIZATION PROBLEMS

PIQIN SHI[1], CHENGJING WANG[1,*], CAN XIANG[1], PEIPEI TANG[2]

[1]*Department of Mathematics, Southwest Jiaotong University, Chengdu 611731, China*
[2]*Department of Computer and Computing Science School, Hangzhou City University, Hangzhou 310015, China*

Abstract. Entropy-regularized quadratic optimization problems are a special class of optimization problems with wide applications in various fields, such as transportation and machine learning. In this paper, we apply the augmented Lagrangian method to this problem with its subproblem solved by the block coordinate descent method. Under certain mild conditions, we analyze the global convergence of this algorithm. Numerical experiments demonstrate the effectiveness of this algorithm.

Keywords. Augmented Lagrangian method; Block coordinate descent method; Entropy regularized quadratic optimization.

2020 Mathematics Subject Classification. 90C20.

1. INTRODUCTION

In this paper, we focus on the entropy-regularized quadratic optimization problem

$$(P)\quad \min_{x\in\mathbf{R}^n}\left\{\tfrac{1}{2}\langle x,Qx\rangle+\langle c,x\rangle+\mu\textstyle\sum_i x_i(\log x_i-1)\,\middle|\,\mathscr{A}x=a,\,x\geq 0\right\},$$

where $Q\in\mathbf{S}^n$ is a positive semi-definite matrix, $\mathscr{A}:\mathbf{R}^n\to\mathbf{R}^m$ is a linear operator, and its adjoint operator is denoted as $\mathscr{A}^*$, and $\mu>0$ is the entropy regularization parameter. Here, we define $0\log 0:=0$.

This model finds widespread applications in practice. In the field of transportation, it is known as the self-penalization gravity model [1]. The introduction of the quadratic term in the objective function allows this model to capture the nonlinear characteristics of cost in transportation processes and simulate real-world traffic congestion. An extended application of the self-penalization gravity model is its incorporation into the framework of combined network equilibrium models, which encompasses the mode choice and trip distribution, and adapts to spatial correlations during calibration [2, 3]. Furthermore, in other areas of relevant modeling, the model structure of (P) is also involved. For instance, in the field of machine learning, this model can be used in probabilistic kernel regression. It is a probability extension form based on the kernel functions of support vector machines. Hence, it is sometimes referred to as a kernel extension of the probability regression model. Many existing models, such as generalized linear models and support vector machines, fall

*Corresponding author.
E-mail address: matwc@swjtu.edu.cn (C.Wang).
Received 17 October 2023; Accepted 24 December 2023; Published online 5 March 2024.

into this category, and it is widely applied to classification and regression problems in the field of machine learning. Additionally, it finds numerous applications in recognition tasks, such as the sequence estimation [4] and pattern recognition. Meanwhile, some optimization problems introduce entropy terms as barrier functions in algorithm design; see, e.g., [5].

As mentioned in [4], due to the entropy regularization term, this problem cannot be simply solved by using standard quadratic programming solvers. Hence, alternative nonlinear optimization techniques need to be explored. This makes the design of the algorithm more complicated. Recently, Fang and Tsao [1] proposed the curved search method (CSM) to solve problem (P), but this algorithm is limited to the case that Q is diagonal. The alternating direction method of multipliers (ADMM) can also be used to solve this problem. For the general case of problem (P), we propose the block coordinate descent (BCD) method based on the augmented Lagrangian method (ALM) to solve it.

In Section 2, we present the notations and fundamental background knowledge. In Section 3, we present the details of the algorithm. In Section 4, we provide the convergence analysis. In Section 5, we present the results of numerical experiments. In Section 6, the last section, we give the concluding conclusion.

2. Preliminaries

In this section, we provide some necessary background knowledge. For detailed contents, we refer to [6].

For a given function $f : \mathbf{R}^n \to [-\infty, +\infty]$, the effective domain of f is

$$\mathrm{dom} f := \{x \in \mathbf{R}^n \mid f(x) < +\infty\}.$$

If $\mathrm{dom} f$ is non-empty, for any $x \in \mathrm{dom} f$, we have $f(x) > -\infty$. Then f is called proper. f is lower semicontinuous at $\bar{x}$ if $\liminf_{x\to\bar{x}} = f(\bar{x})$. In general, if f is lower semicontinuous at every point, it is called a lower semicontinuous function. For a convex function $f : \mathbf{R}^n \to (-\infty, +\infty]$, for any $\hat{x} \in \mathbf{R}^n$, its subdifferential set $\partial f(\hat{x})$ is defined as

$$\partial f(\hat{x}) := \{v \in \mathbf{R}^n \,|\, \forall x \in \mathrm{dom} f, f(x) \geq f(\hat{x}) + \langle v, x - \hat{x}\rangle\}.$$

For any set $C \subseteq \mathbf{R}^n$, its relative interior is defined as

$$\mathrm{ri} C := \{x \in \mathrm{aff} C \,|\, \exists\, \varepsilon > 0, (x + \varepsilon \mathbf{B}) \cap \mathrm{aff} C \subset C\}.$$

aff$C$ denotes the affine hull of the set C [6], and $\mathbf{B}$ is the unit ball in the n-dimensional Euclidean space.

Let $\mathscr{X}$ and $\mathscr{Y}$ be finite-dimensional Euclidean spaces, and let $\Gamma : \mathscr{X} \rightrightarrows \mathscr{Y}$ be a given set-valued function. If

$$\langle x - x', w - w'\rangle \geq 0,\ \forall x, x' \in \mathscr{X},\ w \in \Gamma(x),\ w' \in \Gamma(x'),$$

then Γ is called a monotone operator. If the graph of Γ

$$\mathrm{graph}(\Gamma) := \{(x, y) \in \mathscr{X} \times \mathscr{Y} \,|\, y \in \Gamma(x)\}$$

cannot be contained in the graph of any other monotone operator $\Gamma' : \mathscr{X} \to \mathscr{Y}$, then Γ is called a maximal monotone operator.

3. ALGORITHM

In this section, we present the details of the algorithm. For the convenience of the computation, we introduce an auxiliary variable $y \in \mathbf{R}^n$. Then problem (P) can be reformulated as

$$(P') \min_{x,y\in\mathbf{R}^n} \Big\{ \frac{1}{2}\langle x, Qx\rangle + \langle c, x\rangle + \mu \sum_{i=1}^{n} y_i(\log y_i - 1) \,\Big|\, \mathscr{A}x = a,\ x - y = 0,\ y \geq 0 \Big\}.$$

For problem (P'), its Lagrangian function is given by

$$l(x,y;u,v,z) = \frac{1}{2}\langle x, Qx\rangle + \langle c, x\rangle + \mu \sum_{i} y_i(\log y_i - 1) - \langle u, \mathscr{A}x - a\rangle - \langle v, x - y\rangle - \langle z, y\rangle.$$

The corresponding dual problem is

$$\max_{u\in\mathbf{R}^m, v,x,y,z\in\mathbf{R}^n} \Big\{ -\frac{1}{2}\langle x, Qx\rangle + \langle a, u\rangle - \mu \sum_{i=1}^{n} y_i \,\Big|\, Qx + c - \mathscr{A}^* u - v = 0, \mu \log y + v - z = 0, y, z \geq 0 \Big\}.$$

The Karush-Kuhn-Tucker (KKT) condition for problem (P') is

$$\begin{cases} Qx + c - \mathscr{A}^* u - v = 0, \\ \mu \log y + v - z = 0, \\ \mathscr{A}x - a = 0, \\ x - y = 0, \\ y, z \in \mathbf{R}^n_+. \end{cases} \tag{3.1}$$

For problem (P'), given a parameter $\sigma > 0$, the augmented Lagrangian function is

$$\begin{aligned} L_\sigma(x,y;u,v) = &\frac{1}{2}\langle x, Qx\rangle + \langle c, x\rangle + \mu \sum_{i=1}^{n} y_i(\log y_i - 1) - \langle u, \mathscr{A}x - a\rangle \\ &- \langle v, x - y\rangle + \frac{\sigma}{2}\|\mathscr{A}x - a\|^2 + \frac{\sigma}{2}\|x - y\|^2 + \delta_{\mathbf{R}^n_+}(y). \end{aligned}$$

The fundamental framework of the augmented Lagrangian method is as follows.

Algorithm 1 (ALM):
Given a parameter $\sigma_0 > 0$, select an initial point $(x^0, y^0, u^0, v^0) \in \mathbf{R}^n \times \mathbf{R}^n_+ \times \mathbf{R}^m \times \mathbf{R}^n$, let $k = 0, 1, \cdots$, and perform the following iterations:
Step 1: Call Algorithm 2 to compute an approximate solution

$$(x^{k+1}, y^{k+1}) \approx \operatorname*{argmin}_{x\in\mathbf{R}^n, y\in\mathbf{R}^n_+} L_{\sigma_k}(x, y; u^k, v^k). \tag{3.2}$$

Step 2: Update

$$\begin{aligned} u^{k+1} &= u^k - \sigma_k(\mathscr{A}x^{k+1} - a), \\ v^{k+1} &= v^k - \sigma_k(x^{k+1} - y^{k+1}), \\ z^{k+1} &= \mu \log y^{k+1} + v^{k+1}. \end{aligned}$$

Step 3: If the convergence criterion is satisfied, then terminate the iteration; otherwise, update σ_{k+1} and return to Step 1.

Algorithm 2 (BCD):
Select an initial point $(x^0, y^0) \in \mathbf{R}^n \times \mathbf{R}^n_+$, let $j = 0, 1, \cdots$, and perform the following iterations:
Step 1: Compute an approximate solution

$$x^{j+1} \approx \underset{x \in \mathbf{R}^n}{\operatorname{argmin}} L_{\sigma_k}(x, y^j; u^k, v^k). \tag{3.3}$$

Step 2: Compute an approximate solution

$$y^{j+1} \approx \underset{y \in \mathbf{R}^n_+}{\operatorname{argmin}} L_{\sigma_k}(x^{j+1}, y; u^k, v^k). \tag{3.4}$$

Step 3: If the convergence criterion is satisfied, then terminate the iteration; otherwise, return to Step 1.

Next, we discuss the details of the above algorithm.
Firstly, for subproblem (3.3), let $\nabla_x L_{\sigma_k}(x, y^j; u^k, v^k) = 0$. Then

$$(Q + \sigma_k \mathscr{A}^* \mathscr{A} + \sigma_k I)x = \sigma_k y^j + \sigma_k \mathscr{A}^* a + v^k + \mathscr{A}^* u^k - c. \tag{3.5}$$

The linear system of equations (3.5) can be solved by a direct method or the conjugate gradient method. When Q is a diagonal matrix and $m \ll n$, the inverse of its coefficient matrix can be computed using the Sherman-Morrison-Woodbury formula [7]

$$\begin{aligned}
&(Q + \sigma_k \mathscr{A}^* \mathscr{A} + \sigma_k I)^{-1} \\
&= (Q + \sigma_k I)^{-1} - (Q + \sigma_k I)^{-1} \sigma_k \mathscr{A}^* \left(I + \sigma_k \mathscr{A} (Q + \sigma_k I)^{-1} \mathscr{A}^*\right)^{-1} \mathscr{A} (Q + \sigma_k I)^{-1}.
\end{aligned}$$

This approach can significantly improve the computational efficiency.
Secondly, for subproblem (3.4), letting $\nabla_y L_{\sigma_k}(x^{j+1}, y; u^k, v^k) = 0$, we have

$$\mu \log y_i + v_i^k + \sigma_k (y_i - x_i^{j+1}) = 0, \; i = 1, \ldots, n. \tag{3.6}$$

Notice that, for given v^k and x^{j+1}, equation (3.6) always has a solution in $[\varepsilon, \infty)$. It can be easily verified that the unique solution y_i must satisfy the following condition

$$y_i \in \begin{cases} [1, x_i^{j+1} - \frac{1}{\sigma_k} v_i^k], & \text{if } v_i^k + \sigma_k (1 - x_i^{j+1}) \le 0, \\ [\max\{\varepsilon, \exp((\sigma_k (x_i^{j+1} - 1) - v_i^k)/\mu)\}, 1], & \text{otherwise.} \end{cases}$$

In practical computations, we may set $\varepsilon = 10^{-10}$ since x and y cannot be zeros, and a lower bound must be provided. We can use a binary search method to find an approximate solution to problem (3.6). Given an error tolerance ε, the binary search method has a maximum number of iterations:

$$\max_{1 \le i \le n} \lceil \max\{\log_2(\frac{x_i^{j+1} - \frac{1}{\sigma_k} v_i^k - 1}{\varepsilon}) - 1, \log_2(\frac{1 - (\exp((\sigma_k (x_i^{j+1} - 1) - v_i^k)/\mu))}{\varepsilon}) - 1\} \rceil.$$

4. Convergence Analysis of ALM

In this section, we provide an analysis of the convergence and convergence rate of Algorithm 1. Let $\{\varepsilon_k\}$ and $\{\delta_k\}$ be two sequences that can be summed, $\varepsilon_k \ge 0$, $0 \le \delta_k < 1$, and $0 \le \delta_k' \to 0$ for

all $k \geq 0$. For inner subproblem (3.2), we have the following three stopping criteria:

$$(A) \qquad \phi_k(x^{k+1}, y^{k+1}) - \inf \phi_k(x,y) \leq \frac{\varepsilon_k^2}{2\sigma_k},$$

$$(B) \qquad \phi_k(x^{k+1}, y^{k+1}) - \inf \phi_k(x,y) \leq \frac{\delta_k^2}{2\sigma_k} \|(u^{k+1}, v^{k+1}) - (u^k, v^k)\|^2,$$

$$(C) \qquad \|\nabla \phi_k(x^{k+1}, y^{k+1})\| \leq \frac{\delta_k'}{\sigma_k} \|(u^{k+1}, v^{k+1}) - (u^k, v^k)\|,$$

where $\phi_k(x,y) := L_{\sigma_k}(x,y;u^k,v^k)$. However, $\inf \phi_k(x,y)$ is unknown, so such stopping criteria are practically infeasible. To obtain the implementable stopping criteria, we need to estimate an upper bound on $\phi_k(x^{k+1}, y^{k+1}) - \inf \phi_k(x,y)$. Since

$$\inf \phi_k(x,y) = \sup \psi_k(u,v) \geq \psi_k(u^{k+1}, v^{k+1}),$$

where $\psi_k(u,v) := \inf_{x,y}\{L(x,y;u,v) - \frac{1}{2\sigma_k}\|(u,v) - (u^k,v^k)\|\}$, we have

$$\phi_k(x^{k+1}, y^{k+1}) - \inf \phi_k(x,y) \leq \phi_k(x^{k+1}, y^{k+1}) - \psi_k(u^{k+1}, v^{k+1}).$$

where

$$\psi_k(u^{k+1}, v^{k+1}) = -\frac{1}{2}\langle x, Qx\rangle + \langle a, u^{k+1}\rangle - \mu \sum_{i=1}^{n} y_i - \frac{1}{2\sigma_k}|(u^{k+1}, v^{k+1}) - (u^k, v^k)|,$$

and x and y are the solutions of the equations $Qx + c - \mathscr{A}^* u^{k+1} - v^{k+1} = 0$ and $\mu \log y + v^{k+1} = 0$, respectively. Therefore, we can use the following three stopping criteria

$$(A') \qquad \phi_k(x^{k+1}, y^{k+1}) - \psi_k(u^{k+1}, v^{k+1}) \leq \frac{\varepsilon_k^2}{2\sigma_k},$$

$$(B') \qquad \phi_k(x^{k+1}, y^{k+1}) - \psi_k(u^{k+1}, v^{k+1}) \leq \frac{\delta_k^2}{2\sigma_k} \|(u^{k+1}, v^{k+1}) - (u^k, v^k)\|^2,$$

$$(C') \qquad \|\nabla \phi_k(x^{k+1}, y^{k+1})\| \leq \frac{\delta_k'}{\sigma_k} \|(u^{k+1}, v^{k+1}) - (u^k, v^k)\|.$$

Next, we present the convergence result based on the theory from [8] and [9].

Theorem 4.1. *Assume that the solution set of the KKT system (3.1) is non-empty. Then, under the stopping criterion (A'), the sequence of points $\{(x^k, y^k, u^k, v^k)\}$ generated by the ALM is bounded, and (x^k, y^k) converges to the optimal solution $(\bar{x}, \bar{y})$ of the original problem (P'), while (u^k, v^k) converges to the optimal solution $(\bar{u}, \bar{v})$ of its dual problem.*

For the convenience of later discussions, we define the maximal monotone operator $\mathscr{T}_l$ and $\mathscr{T}_g$ as follows

$$\begin{aligned} \mathscr{T}_g(u,v) &:= \left\{(u',v') \;\middle|\; (-u',-v') \in \partial g(u,v)\right\}. \\ \mathscr{T}_l(x,y,u,v) &:= \left\{(x',y',u',v') \;\middle|\; (x',y',-u',-v') \in \partial l(x,y,u,v)\right\}. \end{aligned}$$

Based on the theory from [9] and [10], we can provide the following conclusion regarding the convergence rate.

Theorem 4.2. *Assume that there exists $\kappa_1 > 0$ such that, for any (u,v) satisfying $dist(0, \mathscr{T}_g(u,v)) < \delta$, $dist((u,v), \mathscr{T}_g^{-1}(0)) \leq \kappa_1 dist(0, \mathscr{T}_g(u,v))$. Let δ be a positive real number. Under the stopping criterion (B'), for sufficiently large k, let $\{(u^k, v^k)\}$ be generated by Algorithm 1 with*

$$dist((u^{k+1}, v^{k+1}), \mathscr{T}_g^{-1}(0)) \leq \mu_k dist((u^k, v^k), \mathscr{T}_g^{-1}(0)),$$

where $\mu_k = [\kappa_1(\kappa_1^2 + \sigma_k^2)^{-\frac{1}{2}} + 2\delta_k](1-\delta_k)^{-1} \to \mu_\infty = \kappa_1(\kappa_1^2 + \sigma_\infty^2)^{-\frac{1}{2}} < 1$. Moreover, under the stopping criterion (C'), assume that there exists $\kappa_2 > 0$ such that, for any (x,y,u,v) satisfying $dist(0, \mathscr{T}_l(x,y,u,v)) < \delta'$, where $\delta' > 0$, $dist((x,y,u,v), \mathscr{T}_l^{-1}(0)) \leq \kappa_2 dist(0, \mathscr{T}_l(x,y,u,v))$. Then, for the sequence $\{(x^k, y^k, u^k, v^k)\}$ generated by Algorithm 1,

$$dist((x^{k+1}, y^{k+1}, u^{k+1}, v^{k+1}), \mathscr{T}_l^{-1}(0)) \leq \theta_k |(u^{k+1}, v^{k+1}) - (u^k, v^k)|,$$

where $\theta_k = \kappa_2(1 + \delta_k'^2)^{\frac{1}{2}} \sigma_k^{-1} \to \theta_\infty = \kappa_2 \sigma_\infty^{-1}$.

5. Numerical Experiment

In this section, we compare the performance of ALM, CSM, and ADMM (where the inner loop of Algorithm 1 iterates only once) in solving problem (P). All numerical experiments in this paper are conducted on a 64-bit Windows 10 laptop with the following specifications: Intel(R) Core(TM) i5-4210M CPU @2.60GHz 2.60GHz, 8GB RAM, and the operating environment ss MATLAB 2016a.

In our experiments, we measure the quality of computed solutions by using the relative primal infeasibility R_P, dual infeasibility R_D, and relative duality gap R_G as follows

$$\begin{aligned} R_P &= \frac{\|\mathscr{A}x - a\| + \|x - y\|}{1 + \|a\| + \|x\|}, \\ R_D &= \frac{\|Qx + c - \mathscr{A}^* u - z + \mu \log x\|}{1 + \|x\| + \|u\|}, \\ R_G &= \frac{|\text{pobj} - \text{dobj}|}{1 + |\text{pobj}| + |\text{dobj}|}, \end{aligned}$$

where

$$\begin{aligned} \text{pobj} &= \frac{1}{2}\langle x, Qx\rangle + \langle c, x\rangle + \mu \sum_{i=1}^{n} x_i(\log x_i - 1), \\ \text{dobj} &= -\frac{1}{2}\langle x, Qx\rangle + \langle a, u\rangle - \mu \sum_{i=1}^{n} x_i. \end{aligned}$$

We stop the iteration of the algorithm when the following conditions are satisfied

$$\eta_{kkt} := \max\{R_P, R_G, R_D\} < \texttt{Tol},$$

where $\texttt{Tol} = 10^{-5}$ is the default value. Additionally, when the number of iterations of ALM and CSM reaches 200, the number of iterations of ADMM reaches 10,000, or the running time exceeds 6 hours, we terminate the iteration.

5.1. **CSM.** In this subsection, we briefly introduce the details of CSM. Consider the following optimization problem

$$\min_{x\in\mathbb{R}^n}\left\{\frac{1}{2}\sum_{i=1}^{n}d_ix_i^2+\langle c,x\rangle+\mu\sum_{i=1}^{n}x_i(\log x_i-1),|Ax=a,x\geq 0\right\},\tag{5.1}$$

where $c\in\mathbf{R}+^n$, $d_i\geq 0$ for $i=1,2,\cdots,n$, $A\in\mathbf{R}^{m\times n}$, $a\in\mathbf{R}^m$ are given, and $\mu>0$ is a regularization parameter derived from the real problem. Note that this problem is actually a special form of (P) with $Q=\mathrm{diag}(d)$, where $d:=(d_1,d_2,\cdots,d_n)$.

The Lagrangian function for problem (5.1) is given by

$$L(x;u)=\frac{1}{2}\sum_{i=1}^{n}d_ix_i^2+\langle c,x\rangle+\mu\sum_{i=1}^{n}x_i(\log x_i-1)-\langle u,\mathscr{A}x-a\rangle+\delta_{\mathbf{R}^n_+}(x).$$

Its dual problem is defined as

$$\min_{u\in\mathbb{R}^m}\left\{f(u):=\frac{1}{2}\sum_{i=1}^{n}d_ih_i(u)^2+\mu\sum_{i=1}^{n}h_i(u)-\langle a,u\rangle\right\}.\tag{5.2}$$

This is an unconstrained optimization problem, where $h(u)=(h_1(u),h_2(u),\cdots,h_n(u))^T$ is expressed implicitly as follows

$$\mu\log h(u)+Qh(u)=A^Tu-c.$$

Note that the objective function of problem (5.2) is twice continuously differentiable with respect to the optimization variable u, and therefore, it can be solved using CSM [11]. The solution framework is as follows.

Algorithm 3 (CSM):
Given parameters $\varepsilon\geq 0$, $\sigma\geq 0$, $\gamma>0$, two selected sequence of $\{\alpha_k\}$ and $\{\beta_k\}$ satisfying $\bar{\alpha}\geq\alpha_k\geq 0$ and $\bar{\beta}\geq\beta_k\geq\underline{\beta}$, where $\bar{\alpha}>0$ and $\bar{\beta}\geq\underline{\beta}>0$, we start with an initial point $u_0\in\mathbf{R}^m$, and iterate as follows:
Step 1: Compute the gradient g_k and the Hessian matrix H_k

$$g_k=\nabla f(u_k)\text{ and }H_k=\nabla^2 f(u_k).$$

If $\|g_k\|\leq\varepsilon$, terminate the iteration; otherwise, proceed with the following steps.
Step 2: Solve $H_kv_k=g_k$ for v_k, and calculate

$$G_k=g_k^Tv_k,\ \gamma_k=\frac{|G_k|}{\|v_k\|\|g_k\|^2},\ \sigma_k=(\det H_k)^2.$$

Step 3: If $\sigma_k>\sigma$ and $\gamma_k>\gamma$, then

$$d_k=-\beta_k\frac{\|g_k\|^2}{G_k}v_k,\ z_k=-\alpha_k\|g_k\|g_k;$$

otherwise,

$$d_k=-g_k,\ z_k=0.$$

Step 4: Calculate the step length

$$t_k\in\arg\min_{t>0}f\left(u_k+td_k+\tfrac{1}{2}t^2z_k\right).$$

Step 5: Update u_{k+1}:

$$u_{k+1}=u_k+t_kd_k+\tfrac{1}{2}t_k^2z_k.$$

5.2. **Numerical Results.** In this subsection, we perform numerical computations for both the general entropy-regularized quadratic optimization problem and the trip assignment problem.
I: The general entropy-regularized quadratic optimization problem. Let us first consider the following problem, which is derived from [4].

$$\begin{aligned}
\min_{\lambda_1^1,\lambda_1^2,\cdots,\lambda_S^N} \quad & \left\{\sum_{k=1}^{S}\left[\sum_{i=1}^{N}\left(\frac{1}{2}\sum_{j=1}^{N}\lambda_k^i Q_{ij}\lambda_k^j + C\left(y_k^i-\frac{\lambda_k^i}{C}\right)\log\left(y_k^i-\frac{\lambda_k^i}{C}\right)\right)\right]\right\} \\
\text{s.t.} \quad & \sum_{i=1}^{N}\lambda_k^i = 0,\ k=1,2,\cdots,S, \\
& \sum_{k=1}^{S}\lambda_k^i = 0,\ i=1,2,\cdots,N, \\
& Cy_k^i \geq \lambda_k^i, k=1,2,\cdots,S, i=1,2,\cdots,N,
\end{aligned} \tag{5.3}$$

where $Q_{ij} = K(x_i,x_j)$, and the function $K(.,.)$ is the Gaussian kernel function, and S and N are given constants. For each $i \in 1,2,\cdots,N$, $x_i \in \mathbf{R}^d$, and $y^i \in \mathbf{R}^S$ with the probabilities for the k-th class denoted as y_k^i satisfying $\sum_{k=1}^{S} y_k^i = 1$. Also, $C > 0$ is a regularization parameter.

Let $w = (w_1^1, w_1^2, \cdots, w_S^N)^T$ with $w_k^i = y_k^i - \frac{\lambda_k^i}{C}$ for $k = 1,2,\cdots,S$ and $i = 1,2,\cdots,N$, and $y = (y_1^1, y_1^2, \cdots, y_S^N)^T$. Let $Q = \mathrm{diag}(Q^0,\cdots,Q^0)$ be an S-block diagonal matrix, where Q^0 is a positive semi-definite matrix with its (i,j)-th element being $Q_{ij}^0 = K(x_i,x_j)$. It is evident that problem (5.3) can be written in the following form

$$\begin{aligned}
\min_{w,\lambda} \quad & \left\{\frac{C}{2}\langle w, Qw\rangle - C\langle w, Qy\rangle + \sum_{i=1}^{N} w_i \log w_i\right\} \\
\text{s.t.} \quad & \mathscr{A}(y-w) = 0, \\
& \mathscr{B}(y-w) = 0, \\
& \lambda - c(y-w) = 0, \\
& w \geq 0,
\end{aligned} \tag{5.4}$$

where $\mathscr{A}(y-w) = 0$ and $\mathscr{B}(y-w) = 0$ represent the constraints $\sum_{i=1}^{N}\lambda_k^i = 0$ and $\sum_{k=1}^{S}\lambda_k^i = 0$, respectively, in their equivalent linear transformation form.

In problem (5.3), for each column attribute $x_k, k \in 1,2,\cdots,d$, they are a set of random vectors following a normal distribution with mean 0 and variance 1. For each $i \in 1,2,\cdots,N$, $y^i \in \mathbf{R}^S$ is a set of random vectors uniformly distributed in an S-dimensional probability density space. The parameter S is set to be 3, the bandwidth of the Gaussian kernel function $K(.,.)$ is chosen from the set $\{0.01, 0.1, 1\}$, and d is set to be 10.

For problem (5.3), we only use ALM and ADMM since Q is not a diagonal matrix, and CSM cannot be used as it cannot compute the curved directions. The results are demonstrated in Table 1. To compare different algorithms, we record the objective value (obj), the number of iterations (iter), accuracy (η_{kkt}), and computational time (time) in different node sizes.

From Table 1, we can see that both ALM and ADMM can achieve the required accuracy for small scaled problems. However, for larger problems, ADMM fails to reach the desired accuracy. ALM is more efficient compared to ADMM, especially for larger problems. The computational

TABLE 1. Numerical results for problem (5.3) solved by ALM and ADMM

N	(C,width)	method	iter	η_{kkt}	obj	time
1000	(0.1, 0.01)	ALM	18	5.87e-06	-1.9650e+03	00:00:04
		ADMM	28	2.18e-06	-1.9651e+03	00:00:05
1000	(10, 0.01)	ALM	55	7.42e-06	-6.2111e+03	00:00:11
		ADMM	126	1.18e-05	-6.2115e+03	00:00:24
1000	(0.1, 0.01)	ALM	54	8.57e-06	-8.5907e+03	00:00:37
		ADMM	151	1.36e-05	-8.5910e+03	00:01:45
3000	(0.1, 0.01)	ALM	18	5.85e-06	-5.8957e+03	00:01:03
		ADMM	28	1.24e-06	-5.8959e+03	00:01:37
3000	(0.1, 1)	ALM	17	9.66e-06	-5.7980e+03	00:03:15
		ADMM	28	1.38e-06	-5.7982e+03	00:05:24
3000	(1, 0.01)	ALM	22	8.52e-06	-6.8320e+03	00:01:17
		ADMM	29	1.33e-06	-6.8322e+03	00:01:55
3000	(10, 0.01)	ALM	55	7.63e-06	-1.8704e+04	00:03:18
		ADMM	156	1.24e-05	-1.8706e+04	00:09:26
6000	(0.1, 0.01)	ALM	18	5.88e-06	-1.1792e+04	00:08:37
		ADMM	28	8.57e-07	-1.1793e+04	00:13:54
6000	(0.1, 0.1)	ALM	18	5.91e-06	-1.1792e+04	00:33:06
		ADMM	28	8.55e-07	-1.1793e+04	00:54:48
6000	(10, 0.01)	ALM	55	7.46e-06	-3.7333e+04	00:35:53
		ADMM	159	1.26e-05	-3.7339e+04	01:05:05
6000	(10, 0.1)	ALM	55	7.39e-06	-3.7452e+04	01:30:42
		ADMM	159	1.27e-05	-3.7459e+04	05:14:27
6000	(10, 1)	ALM	54	8.87e-06	-4.6714e+04	01:44:21
		ADMM	149	1.33e-05	-4.6721e+04	05:11:51

time is positively correlated with the dimension N and the parameter C. Additionally, the performance of the algorithm is influenced by the parameter "width". Figure 1 illustrates the influence of different "width" values on ALM. The figure shows the relationship between η_{kkt} and the computational time for different "width" values. It is evident from the figure that the 'width' value has a noticeable impact on the overall convergence of the algorithm. A smaller 'width' value makes the matrix Q more close to an identity matrix, which leads to the faster convergence.

II: Trip assignment problems. This problem originates from [1], and its formulation is as follows

$$\min_{x\in\mathbf{R}^n}\left\{\frac{1}{2}\langle x,Qx\rangle+\langle c,x\rangle+\mu\sum_{i=1}^{n}x_i(\log x_i-1)\,\middle|\,Ax=a,x\geq 0\right\}, \tag{5.5}$$

where $A\in\mathbf{R}^{m\times n}$ is a matrix derived from the trip distribution problem, with elements consisting of 0s and 1s (see [1]). Additionally, $Q\in\mathbf{S}^n_+$ is a diagonal matrix, $c\in\mathbf{R}^n_+$, $a\in\mathbf{R}^m$ are given vectors, and $\mu>0$ is the regularization parameter.

For problem (5.5), we compare three algorithms: ALM, CSM, and ADMM. The parameter settings follow from those in [1]. The diagonal elements of the matrix Q are obtained by multiplying

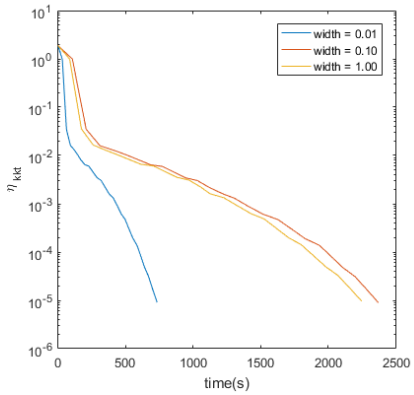

FIGURE 1. The impact of different "width" values on ALM.

random numbers which are uniformly distributed in the interval $(0, 1)$ by corresponding scaling factors, denoted as "scale".

The computing results are presented in the Table 2 and Figure 2. In Table 2, we use "O" to represent the number of origins and "D" to represent the number of destinations. From the results in the table, it can be seen that ALM, CSM, and ADMM are all capable of solving the trip distribution problem effectively. But ALM is obviously more efficient than the other two algorithms. Additionally, Figure 2 provides a representation of the convergence curves for the three algorithms. It is evident from the figure that CSM requires fewest iterations (as it can be understood as a modified Newton method). However, CSM takes much more time at each iteration compared to ALM and ADMM since when applying CSM, we need to compute the values of the gradient and Hessian matrix of the implicit function, which, in turn, requires solving a system of nonlinear equations. Moreover, ALM exhibits more stable and efficient compared to ADMM.

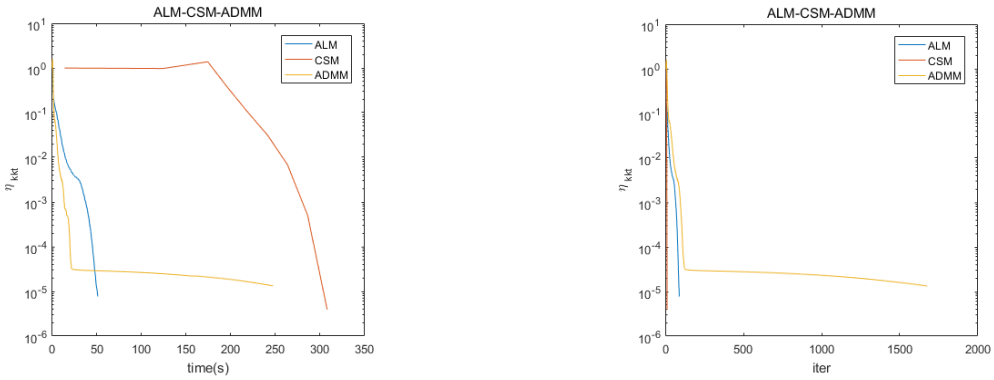

FIGURE 2. Comparison of ALM, CSM, and ADMM

6. CONCLUSION

In this paper, we applied the BCD based ALM to the entropy-regularized quadratic optimization problems. We presented the convergence results of the algorithm under mild conditions. The numerical experiments demonstrate the superiority of ALM compared with CSM and ADMM.

TABLE 2. Numerical results for problem (5.5) solved by ALM, CSM, and ADMM

(O,D)	(scale,μ)	method	iter	η_{kkt}	obj	time
$(200,20)$	$(0.01,0.05)$	ALM	163	9.15e-06	4.7740e+05	00:00:33
		CSM	8	1.09e-06	4.7740e+05	00:01:11
		ADMM	10000	2.40e-05	4.7740e+05	00:07:26
$(200,20)$	(1, 0.05)	ALM	131	9.33e-06	5.0841e+05	00:00:26
		CSM	7	3.75e-06	5.0841e+05	00:00:45
		ADMM	1612	1.34e-05	5.0841e+05	00:01:11
$(300,30)$	(0.01, 0.05)	ALM	164	8.13e-06	3.6663e+04	00:01:43
		CSM	9	3.72e-08	3.6663e+04	00:05:11
		ADMM	10000	2.36e-05	3.6662e+04	00:25:00
$(300,30)$	(0.1, 0.5)	ALM	200	1.26e-05	1.6030e+05	00:02:10
		CSM	7	8.19e-06	1.6030e+05	00:03:22
		ADMM	5278	1.34e-05	1.6030e+05	00:13:25
$(500,40)$	(0.01, 0.05)	ALM	158	7.28e-06	2.7393e+04	00:06:45
		CSM	8	2.40e-08	2.7393e+04	00:25:13
		ADMM	5323	1.22e-05	2.7394e+04	00:55:11
$(500,40)$	(0.01, 5)	ALM	65	9.82e-06	4.8971e+05	00:02:42
		CSM	7	4.19e-06	4.8971e+05	00:18:49
		ADMM	4268	1.28e-05	4.8969e+05	00:44:30
$(500,40)$	(0.1, 0.05)	ALM	145	9.41e-06	5.0022e+04	00:06:04
		CSM	8	3.31e-07	5.0022e+04	00:26:29
		ADMM	176	1.19e-05	5.0018e+04	00:01:44
$(500,40)$	(0.1, 5)	ALM	77	9.11e-06	4.9780e+05	00:03:05
		CSM	7	4.02e-06	4.9780e+05	00:18:59
		ADMM	3758	1.32e-05	4.9780e+05	00:37:32
$(500,40)$	(1, 0.05)	ALM	143	8.59e-06	1.2991e+05	00:05:44
		CSM	10	3.20e-07	1.2991e+05	00:26:08
		ADMM	3531	1.35e-05	1.2991e+05	00:35:16
$(500,40)$	(1, 0.5)	ALM	116	9.53e-06	2.0210e+05	00:04:45
		CSM	6	1.34e-06	2.0210e+05	00:15:06
		ADMM	401	1.42e-05	2.0220e+05	00:03:55
$(500,40)$	(1,5)	ALM	66	9.06e-06	6.3330e+05	00:02:41
		CSM	6	2.61e-06	6.3330e+05	00:15:40
		ADMM	2396	1.27e-05	6.3330e+05	00:24:00

Acknowledgements

Chengjing Wang's work was supported in part by the Zhejiang Provincial Natural Science Foundation of China (Grant No. LTGY23H240002) and the National Natural Science Foundation of China (No. U21A20169). We would like to thank the Editor and the anonymous referees for their helpful suggestions to improve the paper.

REFERENCES

[1] S.-C. Fang, H.-S. Tsao, Linearly-constrained entropy maximization problem with quadratic cost and its applications to transportation planning problems, Transportation Science 29 (1995), 353–365.
[2] L. de Grange, E. Fernández, J. de Cea, A consolidated model of trip distribution, Transportation Research Part E: Logistics and Transportation Review 46 (2010), 61–75.
[3] J. De Cea, J. E. Fernandezz, L. De Grange, Combined models with hierarchical demand choices: A multi-objective entropy optimization approach, Transport Reviews 28 (2008), 415–438.
[4] S. Chakrabartty, G. Cauwenberghs, Forward decoding kernel machines: A hybrid hmm/svm approach to sequence recognition, In: International Workshop on Support Vector Machines, pp. 278–292, Springer, 2002.
[5] S. Gold, A. Rangarajan, Softassign versus softmax: Benchmarks in combinatorial optimization, Advances in Neural Information Processing Systems vol. 8, pp. 626-632, 1995.
[6] R. T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, 1970.
[7] G. H. Golub, C. F. Van Loan, Matrix Computations, JHU press, 2013.
[8] R. T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Contr. Optim. 14 (1976), 877–898.
[9] R. T. Rockafellar, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, Math. Oper. Res. 1 (1976), 97-196.
[10] F. J. Luque, Asymptotic convergence analysis of the proximal point algorithm, SIAM J. Contr. Optim. 22 (1984), 277–293.
[11] A. Ben-Tal, A. Melman, J. Zowe, Curved search methods for unconstrained optimization, Optimization, 21 (1990), 669–695.