SPARSE PROBABILISTIC BOOLEAN NETWORK PROBLEMS: A PARTIAL PROXIMAL-TYPE OPERATOR SPLITTING METHOD

KANGKANG DENG, ZHENG PENG* AND JIANLI CHEN

College of Mathematics and Computer Science, Fuzhou University
Fuzhou 350108, China

(Communicated by Kok Lay Teo)

ABSTRACT. The sparse probabilistic Boolean network (SPBN) model has been applied in various fields of industrial engineering and management. The goal of this model is to find a sparse probability distribution based on a given transition-probability matrix and a set of Boolean networks (BNs). In this paper, a partial proximal-type operator splitting method is proposed to solve a separable minimization problem arising from the study of the SPBN model. All the subproblem-solvers of the proposed method do not involve matrix multiplication, and consequently the proposed method can be used to deal with large-scale problems. The global convergence to a critical point of the proposed method is proved under some mild conditions. Numerical experiments on some real probabilistic Boolean network problems show that the proposed method is effective and efficient compared with some existing methods.

1. Introduction. Probabilistic Boolean network (PBN) is a rule-based uncertainty model which has many applications in the fields of management and industrial engineering. In [15, 16], a PBN model was proposed for analyzing and control of complex systems and those with human decision making such as real-time pricing systems. In [9, 18], the PBN model was applied to the credit default data analysis for the investigation of the relationship between correlated defaults of different industrial sectors and businesses. The PBN model was also applied in [17] to social networks for practical applications, such as product recommendation, targeted online advertising, friend recommendation, or for helping social scientists and political analysts gain insights into public opinion and user behavior. As the early applications, the PBN model was widely used to model the genetic regulatory networks, and was utilized to find the major genes that are in close relationship with the virulence genes, see [4, 5, 23, 24, 25] and their references therein. Besides, the PBN model has many other applications, e.g., optimal control problems and controllability in engineering [13, 14, 35], stock markets [3] and small-world type networks [27].

The PBN model is devoted to constructing the probabilistic Boolean network from a given transition-probability matrix $P$ and a set of Boolean networks (BNs)

2010 Mathematics Subject Classification. Primary: 90B15, 90C52; Secondary: 90C90.

Key words and phrases. Sparse probabilistic Boolean network, $\ell_1$-regularization, separable optimization problem, method of multipliers, proximal point algorithm.

The work is supported by NSFC grants 11571074, 11726505, and the Natural Science Foundation of FuJian Province grant 2015J01010.

* Corresponding author: Zheng Peng.
with matrices \( \{P_i\}_{i=1}^n \). Mathematically, the aim of the PBN model is to seek a probability distribution vector \( x \in \mathbb{R}^n \) such that
\[
\sum_{i=1}^{n} P_i x_i = P, \quad (1a)
\]
\[
e^T x = 1, \quad x \geq 0, \quad (1b)
\]
where \( e = (1, 1, ..., 1)^T \).

Problem (1) is essentially a problem of finding a non-negative solution of a linear system. In this system, the transition-probability matrix \( P \) and the Boolean matrices \( P_i \) are sparse. If a BN has \( j \) nodes, there are \( 2^j \) possible states, and correspondingly the matrices \( P \) and \( P_i \) \((i = 1, ..., n)\) are \( 2^j \) by \( 2^j \) matrices, each column of \( P \) matrix has \( l \) \((\approx j)\) non-zero entries, and each column of \( P_i \) has a unique non-zero entry. System (1) is considered under a very large scale, for instance, in the case that there are \( j \) nodes and each column of \( P \) matrix has \( l \) non-zero entries, system (1) has \( n = l 2^j \) unknowns. Hence, it is also indeterminate. On the other hand, the probability distribution of the PBN model is essentially a sparse vector in many practical applications. In these cases, the sparse PBN model, i.e., a sparse solution of system (1), is recommended.

A least squares solution of system (1) is given by the following minimization problem
\[
\begin{cases}
\min_x \frac{1}{2} \left\| P - \sum_{i=1}^{n} P_i x_i \right\|_F^2, \\
s.t. \quad e^T x = 1, \quad x \geq 0.
\end{cases}
\]

Let “vec” be a mapping from \( \mathbb{R}^{s \times s} \) to \( \mathbb{R}^{s^2} \), where
\[
\text{vec}
\begin{pmatrix}
a_{11} & \cdots & a_{1s} \\
\vdots & \ddots & \vdots \\
a_{s1} & \cdots & a_{ss}
\end{pmatrix}
= (a_{11}, \cdots, a_{s1}, a_{12}, \cdots, a_{s2}, \cdots, \cdots, a_{s1}, \cdots, a_{ss})^T,
\]
and \( A = [\text{vec}(P_1), \text{vec}(P_2), \cdots, \text{vec}(P_n)] \), \( b = \text{vec}(P) \in \mathbb{R}^m \) with \( m = 2^{2j} \). Then problem (2) can be expressed as
\[
\begin{cases}
\min_{x} \frac{1}{2} \left\| Ax - b \right\|^2, \\
\quad e^T x = 1, \quad x \geq 0,
\end{cases}
\]
which is a convex minimization problem. The coefficient matrix \( A \) is a \( 2^{2j} \times l 2^j \) matrix.

To narrow down the solution set of problem (3), the researchers considered a solution given by the largest entropy based on the fact that \( x \) is a probability distribution in [4], and they converted problem (3) to
\[
\begin{cases}
\min_{x \in \mathbb{R}^n} \frac{1}{2} \left\| Ax - b \right\|^2 + x^T \log x, \\
\quad e^T x = 1, \quad x \geq 0.
\end{cases}
\]

However, to characterize the solution’s sparsity of optimization problem, the \( \ell_0 \)-regularization is a common choice. By \( \ell_0 \)-regularization, problem (3) becomes to
\[
\begin{cases}
\min_{x \in \mathbb{R}^n} \frac{1}{2} \left\| Ax - b \right\|^2 + \tau \left\| x \right\|_0, \\
\quad e^T x = 1, \quad x \geq 0,
\end{cases}
\]
where $\tau > 0$ is the regularization parameter. Problem (5) is NP-hard [20] and intractable. To overcome this difficulty, many researchers relaxed $\|x\|_0$ to $\|x\|_1$. By $\ell_1$-regulation the problem (5) becomes

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad \frac{1}{2}\|Ax - b\|^2 + \tau \|x\|_1, \\
\text{s.t.} & \quad e^T x = 1, \quad x \geq 0.
\end{align*}
\]

Problem (6) is also a convex minimization problem. Many methods, which are developed based on augmented Lagrangian of multipliers [28, 6, 33, 11, 10, 12, 21, 2, 7, 29, 8], can be used to solve separable convex minimization problem. These methods are globally convergent to global solution. However, since $x \geq 0$, we have $\|x\|_1 = \sum_{i=1}^{n} |x_i| = e^T x = 1$. The regularization term $\tau \|x\|_1$ is inoperative in the constrained optimization problem (6).

It was suggested in [31] that, we should relax the $\ell_0$ to $\ell_1$ regularization for characterizing the sparsity. By a phase diagram study, it was shown in [30] that, the $\ell_q$ ($0 < q < 1$) regularization can assuredly generate more sparse solutions than $\ell_1$ regularization, in which the index $q = \frac{1}{2}$ somehow plays a representative role. If $q \in [\frac{1}{2}, 1)$, the smaller $q$ will result in more sparse of the solution obtained by $\ell_q$ regularization. If $q \in (0, \frac{1}{2}]$, the performance of $\ell_q$ regularization has no significant difference. From the computational point of view, there is an iterative half thresholding algorithm for the fast solution of $\ell_1$ regularization.

For $\ell_{\frac{1}{2}}$ regularization problem of the form

\[
\min_{x \in \mathbb{R}^n} \|Ax - b\|^2 + \gamma \|x\|^{\frac{1}{2}},
\]

the iterative half thresholding algorithm is given by [31]:

\[
x^{k+1} = H_{\gamma \eta, \frac{1}{2}}(x^k + \etaos A^T (b - Ax^k)),
\]

where $\eta > 0$. The operator $H_{\gamma \eta, \frac{1}{2}}$ is specified by

\[
H_{\gamma \eta, \frac{1}{2}}(z) = (h_{\gamma \eta, \frac{1}{2}}(x_1), h_{\gamma \eta, \frac{1}{2}}(x_2), \ldots, h_{\gamma \eta, \frac{1}{2}}(x_n))^T,
\]

where

\[
h_{\gamma \eta, \frac{1}{2}}(z) = \begin{cases} f_{\gamma \eta, \frac{1}{2}}(z), & |z| > \frac{\sqrt{3\pi}}{4}(\gamma \eta)^{\frac{2}{3}}, \\
0, & \text{otherwise,}
\end{cases}
\]

with

\[
f_{\gamma \eta, \frac{1}{2}}(z) = \frac{2}{3} z \left( 1 - \cos \left( \frac{2\pi}{3} \frac{2}{3} - \frac{2}{3} \varphi_{\gamma \eta}(z) \right) \right)
\]

and

\[
\varphi_{\gamma \eta}(z) = \arccos \left( \frac{\gamma \eta \frac{|z|}{3} - \frac{2}{3}}{8} \right).
\]

Under some conditions, it was shown in [34] that the iterative half thresholding algorithm converges to a local solution of the regularization problem (7) with an eventually linear convergence rate. More recently, an interior-point $\ell_{\frac{1}{2}}$-penalty method is proposed for inequality constrained nonlinear optimization [26], and the global convergence results of the proposed method are established.
Based on the above observation, we relax the $\ell_0$ regularization in this paper. Then we get the $\ell_1$ regularization problem of the form
\[
\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|^2 + \frac{1}{2\mu} (e^T x - 1)^2 + \tau \|x\|^{1/2}_{1/2},
\]
\[
\text{s.t. } e^T x = 1, \quad x \geq 0.
\]
By using a penalty method to problem (13), we have
\[
\min_{x \geq 0} \frac{1}{2} \|Ax - b\|^2 + \frac{1}{2\mu} (e^T x - 1)^2 + \tau \|y\|^{1/2}_{1/2},
\]
(14)
where $\mu > 0$ is a penalty parameter. A separable reformulation of (14) is given by
\[
\begin{align*}
\min & \quad \frac{1}{2} \|Ax - b\|^2 + \frac{1}{2\mu} (e^T x - 1)^2 + \tau \|y\|^{1/2}_{1/2} \\
\text{s.t.} & \quad x - y = 0, \quad x \geq 0.
\end{align*}
\]
We will propose a partial proximal-type operator splitting method to solve problem (15), and prove the convergence to a critical point of the proposed method.

The rest of this paper is organized as follows. In Section 2, we propose a partial proximal-type operator splitting method to solve problem (15) arising in the SPBN, and all the subproblem-solvers of the proposed method are developed. In Section 3, the convergence to a critical point of the proposed method is proven under some mild conditions. Some numerical results are presented in Section 4 to demonstrate the effectiveness and efficiency of the proposed method. Section 5 concludes this paper with some final remarks.

2. The proposed method. In this section, a partial proximal-type operator splitting method for solving problem (15) will be proposed, and the solvers of all iteration subproblem will be developed.

The augmented Lagrangian function associated with problem (15) is
\[
L_{\rho}(x, y, \lambda) = \frac{1}{2} \|Ax - b\|^2 + \frac{1}{2\mu} (e^T x - 1)^2 + \tau \|y\|^{1/2}_{1/2} - \lambda^T (x - y) + \frac{\rho}{2} \|x - y\|^2,
\]
(16)
where $\lambda \in \mathbb{R}^n$ is a Lagrange multiplier and $\rho > 0$ is a penalty parameter.

Based on the augmented Lagrangian function, we propose the partial proximal-type split augmented Lagrangian method as follows.

**Algorithm 1.** The partial proximal-type operator splitting method

For a given $(x^k, y^k, \lambda^k)$, the new iterate $(x^{k+1}, y^{k+1}, \lambda^{k+1})$ is generated via:

\[
y^{k+1} = \arg \min_{y \geq 0} \left\{ L_{\rho}(x^k, y, \lambda^k) \right\}, \quad (17)
\]
\[
x^{k+1} = \arg \min_x \left\{ L_{\rho}(x, y^{k+1}, \lambda^k) + \frac{r}{2} \|x - x^k\|^2, \quad r > 0 \right\}, \quad (18)
\]
\[
\lambda^{k+1} = \lambda^k - \rho (x^{k+1} - y^{k+1}). \quad (19)
\]

In what follows, we will specify all the subproblem-solvers of the proposed method. Firstly, for subproblem (17), we have
\[
y^{k+1} = \arg \min_{y \geq 0} \left\{ \tau \|y\|^{1/2}_{1/2} + \frac{\rho}{2} \|y - (x^k + \frac{1}{\rho} \lambda^k)\|^2 \right\}. \quad (20)
\]
Let $\epsilon \in (0, 1)$ be a small number, and $\eta = 1 - \epsilon$. Then, the iterative half thresholding algorithm for problem (20) can be described as follows.

In what follows, we will specify all the subproblem-solvers of the proposed method. Firstly, for subproblem (17), we have
\[
y^{k+1} = \arg \min_{y \geq 0} \left\{ \tau \|y\|^{1/2}_{1/2} + \frac{\rho}{2} \|y - (x^k + \frac{1}{\rho} \lambda^k)\|^2 \right\}. \quad (20)
\]
Algorithm 2. Iterative half thresholding method for problem (20)

s0. initial: \( \tilde{y}^0 = y^k, t = 0, \varepsilon_t = 1.0; \)
s1. while \( \varepsilon_t > \varepsilon \), do

\[
\theta(\tilde{y}^t) = y^t - \eta(y^t - (x^k - \frac{1}{\rho} \lambda^k));
\]

\[
\kappa = \frac{\sqrt{96}}{9\eta} \left| \theta(\tilde{y}^t) \right|_{s+1}^2
\]

where \([v]_r\) denotes the \(r\)-th largest component of \(v\).

compute a new predictor via

\[
\tilde{y}^{t+1} = H_{\kappa \eta} \theta(\tilde{y}^t)
\]

update \( \varepsilon_t = \|\tilde{y}^{t+1} - \tilde{y}^t\|\infty; \)

end (while)

s2. compute a new iterate via

\[
y^{k+1} = \max\{\tilde{y}^{t+1}, 0\}.
\]

The iterative half thresholding algorithm is computationally tractable. Hence, the subproblem (17) is referred to as an easy problem.

Secondly, we focus on subproblem (18). Note that

\[
L_\rho(x, y^{k+1}, \lambda^k) = \frac{1}{2} \|Ax - b\|^2 + \frac{1}{2\mu} (e^T x - 1)^2
\]

\[
-(\lambda^k)^T (x - y^{k+1}) + \frac{\rho}{2} \|x - y^{k+1}\|^2 + \tau \|y^{k+1}\|_1^2.
\]

(21)

Since matrix \(A\) is very large size, the storage-cost of the full matrix \(A\) may be very expensive. By the sparsity of \(A = [f_1, \cdots, f_n]\) where \(f_i = \text{vec}(P_i)\), a coordinate descent method is recommended to solve subproblem (18).

Let

\[
F_k(x) = L_\rho(x, y^{k+1}, \lambda^k) + \frac{\rho}{2} \|x - x^k\|^2.
\]

The coordinate descent method for subproblem (18) can be stated as follows:

Algorithm 3. Coordinate descent method for problem (18)

s0. initial: let \( \tilde{x}^0 = x^k, l = 0, \varepsilon_l = 1.0; \)
s1. while \( \varepsilon_l > \varepsilon \), do

for \( i = 1, 2, \cdots, n \), let

\[
\tilde{x}_i^{l+1} = \arg \min F_k(\tilde{x}_i^{l+1}, x_i, \tilde{x}_{i+1}^{l})
\]

end (for)

update \( \varepsilon_l = \|\nabla F_k(\tilde{x}^{l+1})\|; \)

end (while)

s2. \( x^{k+1} = \tilde{x}^{l+1} \)

By the definition, \( f_i = \text{vec}(P_i) \) \((i = 1, 2, \cdots, n)\) and \( f = \text{vec}(P)\) are sparse vectors which can be stored in the sparse form. To do so, the memory-cost can be reduced immensely. When the \(i\)-th component, \(x_i\), is updated, the coordinate descent method only needs to compute the inner product of some sparse vectors. The coordinate descent method has a low computational cost because it does not involve the matrix-computation.
3. Convergence analysis. The Lagrange function associated with problem (15) is
\[ L(x, y, \lambda) = \frac{1}{2} \|Ax - b\|^2 + \frac{1}{2\mu}(e^T x - 1)^2 + \tau \|y\|_{\frac{3}{2}}^2 - \lambda^T(x - y). \] (22)
By the first-order optimality conditions, \((x^*, y^*, \lambda^*) \in R^n \times R^n_+ \times R^n\) is a critical point of (15) if and only if
\[
\begin{aligned}
& A^T(Ax^* - b) + \frac{1}{\mu}(e^T x^* - 1)e - \lambda^* = 0, \\
& \tau \|y\|_{\frac{3}{2}}^2 - \tau \|y^*\|_{\frac{3}{2}}^2 + (\lambda^*)^T(y - y^*) \geq 0, \quad \forall y \geq 0 \\
& x^* - y^* = 0.
\end{aligned}
\] (23)
By the optimality conditions of the iteration subproblems (17) and (18), which are combined with (19), we obtain
\[
\tau(\|y\|_{\frac{3}{2}}^2) - \tau(\|y^{k+1}\|_{\frac{3}{2}}^2) + (y - y^{k+1})^T \lambda^{k+1} \geq \rho(y - y^{k+1})^T(x^k - x^{k+1}), \forall y \geq 0,
\] (24)
\[
A^T(Ax^{k+1} - b) + \frac{1}{\mu}(e^T x^{k+1} - 1)e - \lambda^{k+1} = r(x^k - x^{k+1}),
\] (25)
\[
x^{k+1} - y^{k+1} = \frac{1}{\rho}(\lambda^k - \lambda^{k+1}).
\] (26)
Let \(v = (x, \lambda)\). It is easy to show that, to prove the convergence to a critical point of the proposed partial proximal-type operator splitting method, one needs only to show that
\[
\lim_{k \to \infty} \|v^{k+1} - v^k\|^2 = 0.
\] (27)

Lemma 3.1. For a given \(w^k := (x^k, y^k, \lambda^k)\), let \(w^{k+1} := (x^{k+1}, y^{k+1}, \lambda^{k+1})\) be generated by the Algorithm 1. Then,
\[
L_\rho(x^{k+1}, y^{k+1}, \lambda^{k+1}) \leq L_\rho(x^k, y^k, \lambda^k) + \frac{1}{\rho}\|\lambda^{k+1} - \lambda^k\|^2 - \frac{\tau}{2}\|x^{k+1} - x^k\|^2.
\] (28)
Proof. Since \(w^{k+1}\) is generated by Algorithm 1, it follows from (17) and (18) that
\[
L_\rho(x^k, y^{k+1}, \lambda^k) \leq L_\rho(x^k, y^k, \lambda^k),
\] (29)
\[
L_\rho(x^{k+1}, y^{k+1}, \lambda^k) \leq L_\rho(x^k, y^{k+1}, \lambda^k) - \frac{\tau}{2}\|x^{k+1} - x^k\|^2.
\] (30)
By (19) of Algorithm 1, we get
\[
L_\rho(x^{k+1}, y^{k+1}, \lambda^{k+1}) - L_\rho(x^k, y^{k+1}, \lambda^k) = \frac{1}{\rho}\|\lambda^{k+1} - \lambda^k\|^2.
\] (31)
Adding (29), (30) and (31) yields (28). □

Letting \(\sigma = \|A^T A + \frac{1}{\mu}ee^T + rI\|\), we have the following lemma.

Lemma 3.2. For the sequence \(\{v^k\}\) generated by the Algorithm 1, it holds that
\[
\|\lambda^{k+1} - \lambda^k\|^2 \leq 2\sigma^2\|x^{k+1} - x^k\|^2 + 2\sigma^2\|x^k - x^{k-1}\|^2.
\] (32)
Proof. By (25), we get
\[
\lambda^{k+1} = A^T(Ax^{k+1} - b) + \frac{1}{\mu}(e^T x^{k+1} - 1)e + r(x^{k+1} - x^k),
\]
and at the previous iteration, we have
\[ \lambda^k = A^T (Ax^k - b) + \frac{1}{\mu} (e^T x^k - 1)e + r(x^k - x^{k-1}). \]

Hence
\[ \lambda^{k+1} - \lambda^k = (A^T A + \frac{1}{\mu} ee^T + rI)(x^{k+1} - x^k) - r(x^k - x^{k-1}). \] (33)

Taking norm on both sides of (33) and using triangle inequality, we get
\[ \| \lambda^{k+1} - \lambda^k \| \leq \| A^T A + \frac{1}{\mu} ee^T + rI \| \| x^{k+1} - x^k \| + r\| x^k - x^{k-1} \|. \]

Since \( a^2 + b^2 \geq 2ab \), we have
\[ \| \lambda^{k+1} - \lambda^k \|^2 \leq 2\sigma^2 \| x^{k+1} - x^k \|^2 + 2r^2 \| x^k - x^{k-1} \|^2, \] (34)

which establishes the desired result.

Combining Lemma 3.1 with Lemma 3.2, we obtain
\[ L_\rho(x^{k+1}, y^{k+1}, \lambda^{k+1}) \leq L_\rho(x^k, y^k, \lambda^k) - \left( r^2 - 2\sigma^2 - 2r^2 \right) \frac{\| x^{k+1} - x^k \|^2}{\rho} \]
which deduces to
\[ L_\rho(x^{k+1}, y^{k+1}, \lambda^{k+1}) + \frac{2r^2}{\rho} \| x^{k+1} - x^k \|^2 \leq L_\rho(x^k, y^k, \lambda^k) + \frac{2r^2}{\rho} \| x^k - x^{k-1} \|^2 \]
\[ - \left( r^2 - 2\sigma^2 - 2r^2 \right) \frac{\| x^{k+1} - x^k \|^2}{\rho}. \] (35)

If the penalty parameter \( \rho > 0 \) is large enough, the augmented Lagrange function \( L_\rho(x, y, \lambda) \) is bounded below. Assume that
\[ \rho > \frac{4\sigma^2}{r} + 4r, \]
which implies
\[ \kappa = \frac{r}{2} - \frac{2(\sigma^2 + r^2)}{\rho} > 0. \]

We have the following theorem.

**Theorem 3.3.** The sequence \( \{v^k = (x^k, \lambda^k)\} \) generated by Algorithm 1 is asymptotically regular \(^1\). This is
\[ \lim_{k \to \infty} \| v^{k+1} - v^k \|^2 = 0. \] (36)

**Proof.** By (35), we get
\[ \kappa \| x^{k+1} - x^k \|^2 \leq \left\{ L_\rho(x^k, y^k, \lambda^k) + \frac{2r^2}{\rho} \| x^k - x^{k-1} \|^2 \right\} \]
\[ - \left\{ L_\rho(x^{k+1}, y^{k+1}, \lambda^{k+1}) + \frac{2r^2}{\rho} \| x^{k+1} - x^k \|^2 \right\}. \] (37)

\(^1\)A sequence \( \{\xi_k\} \) is said to be asymptotically regular if \( \| \xi_k - \xi_{k+1} \| \to 0 \) as \( k \to \infty \). The general definition of “asymptotically regularity” can be found in [22].
Summing (37) with respect to $k$ yields
\[
\sum_{j=1}^{k} \|x^{j+1} - x^j\|^2 \leq \left\{ L_{\rho}(x^1, y^1, \lambda^1) + \frac{2\tau^2}{\rho} \|x^1 - x^0\|^2 \right\} \\
- \left\{ L_{\rho}(x^{k+1}, y^{k+1}, \lambda^{k+1}) + \frac{2\tau^2}{\rho} \|x^{k+1} - x^k\|^2 \right\}.
\]
(38)

Since the augmented Lagrange function $L_{\rho}(x, y, \lambda)$ is lower-bounded, $L_{\rho}(x^{k+1}, y^{k+1}, \lambda^{k+1}) + \frac{2\tau^2}{\rho} \|x^{k+1} - x^k\|^2$ is also lower-bounded for any $k$. We have
\[
\sum_{j=1}^{k} \|x^{j+1} - x^j\|^2 \leq C < \infty,
\]
which implies
\[
\lim_{k \to \infty} \|x^{k+1} - x^k\|^2 = 0.
\]
(39)

Combining (39) with Lemma 3.2, we have
\[
\lim_{k \to \infty} \|\lambda^{k+1} - \lambda^k\|^2 = 0.
\]
(40)

The assertion (36) follows directly from (39) and (40).

It is obvious that, for a given $(x^0, y^0, \lambda^0) \in R^n \times R^n \times R^n$ the level set $H := \{(x, y, \lambda) \in R^n \times R^n \times R^n \mid L_{\rho}(x, y, \lambda) \leq L_{\rho}(x^0, y^0, \lambda^0)\}$ is bounded and closed, where $L_{\rho}(x, y, \lambda)$ is defined by (16). By the settings of the parameters $\rho$ and $\tau$, it follows from (35) that, the sequence $\{L_{\rho}(x^k, y^k, \lambda^k) + \frac{2\tau^2}{\rho} \|x^k - x^{k-1}\|^2\}$ is decreasing for $\forall \ k \geq 1$. Hence, the sequence $\{x^k := (x^k, y^k, \lambda^k, x^{k-1})\}$ is contained in a bounded and closed set, and so is the partial vector-sequence $\{w^k := (x^k, y^k, \lambda^k)\}$. This implies that the sequence $\{w^k\}$ has convergent subsequence. Assume that $\{w^{k_j}\} \subseteq \{w^k\}, k_j \in \mathcal{K} := \{1, 2, \cdots \}$ is a convergent subsequence, and let $w^{\infty} := \lim_{k_j \to \infty} w^{k_j}$.

**Theorem 3.4.** The cluster point of the sequence $\{w^k\}$ generated by Algorithm 1 is a critical point of problem (15).

**Proof.** Note that (24)-(25) and (26) hold for $\forall \ k_j \in \mathcal{K}$. Taking limits as $k_j \to \infty$ on the both sides of these equations, and using Theorem 3.3, we have $w^{\infty} \in R^n \times R^n \times R^n$ and
\[
\begin{align*}
A^T (Ax^{\infty} - b) + \frac{1}{\tau} (e^T x^{\infty} - 1) e - \lambda^{\infty} &= 0, \\
\tau \|y\| \frac{1}{2} - \tau \|y^{\infty}\| \frac{1}{2} + (\lambda^{\infty})^T (y - y^{\infty}) &\geq 0, \quad \forall y \geq 0, \\
x^{\infty} - y^{\infty} &= 0.
\end{align*}
\]
(41)

Which implies that $w^{\infty}$ is a critical point of problem (15). As generality of the subsequence $\{w^{k_j}\}$, any cluster point of the sequence $\{w^k\}$ (generated by Algorithm 1) is a critical point of problem (15).

Assume that $w^*$ is a cluster point of the sequence $\{w^k\}$ generated by Algorithm 1. Then, by Theorem 3.4, it is a critical point defined by (23), and we have
\[
\lim_{k \to \infty} \inf \|w^k - w^*\| = 0.
\]
4. **Numerical experiments.** In this section, some numerical results on the SPBN problem will be presented to demonstrate the performance of Algorithm 1. All the testing examples are chosen from the BN/PBN MATLAB-based toolbox proposed in [19]. All methods used in these numerical experiments are coded in MATLAB 2014a and run on a personal computer Macbook Pro with 2.6 GHz Intel Core i5 and 8GB RAM.

The stopping criterion of Algorithm 1 is set to \( \|v^{k+1} - v^k\|_{\infty} < \varepsilon \), where \( \varepsilon > 0 \) is a small real number. The parameters are set to \( \rho = 0.8, \mu = 0.3, r = 1.6, \eta = 0.95 \).

For subproblem (17), the sparsity regularization parameter \( s \) is given by a cross-validation, and the \( \ell_1 \) regularization parameter \( \tau \) is set to \( \tau_t = \frac{\sqrt{96}}{9\eta} \rho \|(\theta(y^t))_{s+1}\|_2^2 \), (42)

where \( t \) is the inner iteration counter of Algorithm 2. The initial point is set to \( x_0 = 0, y_0 = 1, \lambda_0 = 0 \).

**Example 1.** There are \( j = 2 \) nodes, and each column of the given transition-probability matrix \( P \) has \( l = 2 \) non-zero entries. The observed transition-probability matrix of the PBN is

\[
P = \begin{pmatrix}
0.1 & 0.3 & 0.5 & 0.6 \\
0.0 & 0.7 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.5 & 0.0 \\
0.9 & 0.0 & 0.0 & 0.4
\end{pmatrix}
\]

In this case, there are 16 Boolean network matrices. The SPBN problem seeks a sparse solution of the following system

\[
\sum_{i=1}^{16} x_i P_i = P
\]

with \( x \geq 0 \) and \( e^T x = 1 \).

For this example, three methods are tested: Algorithm 1, the maximum entropy rate approach (MEM) proposed in [4] and the modified maximum entropy rate approach (MMEM) proposed in [5]. The computational results are displayed in Figure 1 for easy comparison. The solution generated by Algorithm 1 is as sparse as that of the MMEM, and both are sparser than that of generated by the MEM. We notice that, the MMEM also emphasizes the sparsity by using \( \ell_1 \)-regularization.

To do so, the objective function of the MMEM is

\[
\max_{x} \left\{ -\sum_{i=1}^{n} x_i \log x_i - \beta \sum_{i=1}^{n} x_i^\alpha \right\}.
\]

With the best choice of the parameter pair \((\alpha, \beta)\), the MMEM obtains a sparse solution of the PBN problem. However, the computational results of the MMEM are very sensitive to the parameter pair \((\alpha, \beta)\). In the case when there are 2 nodes and each column of the given transition-probability matrix \( P \) has 2 non-zero entries, the best choice of the pair \((\alpha, \beta) = (0.63, 1.40)\) is determined by numerical experiments without theoretical justification. Algorithm 1 is independent of experimental parameters, and also gives a solution as sparse as that of the MMEM.
model of the MEM is a convex optimization, the solution generated by MEM is more logical since it is a global optimal solution. The MEM identifies six major BNs: $P_9, P_{11}, P_{13}, P_{14}, P_{15}$ and $P_{16}$, and Algorithm 1 gives a more sparse solution with four major BNs included in those generated by the MEM. The MMEM gives a different solution for which it identifies the major BNs: $P_6, P_8, P_{10}, P_{12}, P_{13}$ and $P_{15}$.

Example 2. There are $j=2$ nodes, and each column of the observed transition-probability matrix $P$ has $l=3$ non-zero entries, where

$$
P = \begin{pmatrix}
0.1 & 0.3 & 0.2 & 0.1 \\
0.2 & 0.3 & 0.2 & 0.0 \\
0.0 & 0.0 & 0.6 & 0.4 \\
0.7 & 0.4 & 0.0 & 0.5
\end{pmatrix}.
$$

In this case, there are 81 BNs in the SPBN problem.

The MMEM obtains the best choice of the parameter pair $(0.61, 0.6)$ from 1980 pairs of $(\alpha, \beta)$, then it gives a sparse solution with six major BNs being identified. Algorithm 1 identifies eight major Boolean networks, and the re-constructed PBN is dominated by the $5^{th}$, $11^{th}$, $47^{th}$, $50^{th}$, $63^{rd}$, $72^{nd}$, $73^{rd}$ and $80^{th}$ BNs. The numerical solutions generated by the three methods are displayed in Figure 2 for comparison. Compared with the results obtained by the MEM and MMEM, the solution generated by Algorithm 1 is more sparse, and Algorithm 1 can identify the major BNs in the PBN model more readily.

Example 3. There are 1024 BNs, and each column of the observed transition-probability matrix $P$ has different non-zero entries, where
Figure 2. The probability distribution $x$ for the case $j = 2$ and $l = 3$

$$P = \begin{pmatrix} 0.12 & 0 & 0.60 & 0.42 & 0 & 0 & 0 & 0 \\ 0.28 & 0 & 0 & 0.18 & 0 & 0 & 0 & 0 \\ 0 & 0.40 & 0 & 0 & 0.40 & 0.18 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.42 & 0 & 0.60 \\ 0.18 & 0 & 0.40 & 0.28 & 0 & 0 & 0 & 0 \\ 0.42 & 0 & 0 & 0.12 & 0 & 0 & 0 & 0 \\ 0 & 0.60 & 0 & 0 & 0.60 & 0.12 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.28 & 1.00 & 0.40 \end{pmatrix}$$

With the other parameters being fixed, we observe the effect of stopping error $\varepsilon$ in relation to the computational results. The computational results generated by Algorithm 1 are put in Table 1. From Table 1, we find that the identified major BNs

| Stopping error $\varepsilon$ | $10^{-2}$ | $10^{-3}$ | $5 \times 10^{-4}$ | $10^{-4}$ | $5 \times 10^{-5}$ | $10^{-5}$ |
|-----------------------------|----------|----------|---------------------|----------|---------------------|----------|
| Total iteration number $k$  | 97       | 260      | 267                 | 520      | 562                 | 722      |
| Identified major BNs        | 104      | 118      | 189                 | 360      | 360                 | 395      |
|                             | 360      | 395      | 395                 | 395      | 395                 | 395      |
|                             | 376      | 594      | 594                 | 594      | 594                 | 594      |
|                             | 395      | 836      | 836                 | 836      | 836                 | 836      |
|                             | 594      | 911      | 911                 | 911      | 911                 | 911      |
|                             | 836      | 939      | 939                 | 939      | 939                 | 939      |
|                             | 911      | 939      | 939                 | 939      | 939                 | 939      |
keep unchanged while the computational cost increases as the stopping error $\varepsilon$ is continuously deceasing to $1.0 \times 10^{-5}$. This observation implies that, the identified major BNs generated by Algorithm 1 is spot-on, reliable, and robust under the different error settings.

If the stopping error is set to $\varepsilon = 1.0 \times 10^{-3}$, Algorithm 1 identifies 7 major BNs from 1024 BNs with 260 iterations. The reconstructed transition probability matrix is

$$
\hat{P} = \begin{pmatrix}
0.1199 & 0 & 0.5999 & 0.42 & 0 & 0 & 0 & 0 \\
0.2801 & 0 & 0 & 0.18 & 0 & 0 & 0 & 0 \\
0 & 0.4001 & 0 & 0 & 0.4001 & 0.18 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.42 & 0 & 0.5999 \\
0.18 & 0 & 0.4001 & 0.2801 & 0 & 0 & 0 & 0 \\
0.42 & 0 & 0 & 0.1199 & 0 & 0 & 0 & 0 \\
0 & 0.5999 & 0 & 0 & 0.5999 & 0.1199 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.28 & 0.00 & 0.28 & 1.00 & 0.4001 \\
\end{pmatrix}.
$$

The reconstructed residual is $\|\hat{P} - P\|_F = 3.4067 \times 10^{-4}$, and the cpu-time cost is 66.86 seconds. The major BNs identified by Algorithm 1 is displayed in Figure 3.

![Figure 3. The probability distribution $x$ for the case $j = 3$ and 1024 BNs](image)

**Example 4.** There are $j = 3$ nodes, and each column of the observed transition-probability matrix $P$ has 2 or 4 non-zero entries, where

$$
P = \begin{pmatrix}
0.5672 & 0.4328 & 0.2881 & 0 & 0.1447 & 0 & 0.4328 & 0 \\
0 & 0 & 0.1447 & 0 & 0.2881 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3776 \\
0 & 0 & 0 & 0.4328 & 0 & 0 & 0 & 0.1896 \\
0.4328 & 0.5672 & 0.3376 & 0 & 0.1896 & 0 & 0.5672 & 0 \\
0 & 0 & 0.1896 & 0 & 0.3776 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.6657 & 0 & 0.2881 & 0 \\
0 & 0 & 0 & 0.5672 & 0 & 0.3343 & 0 & 0.1447 \\
\end{pmatrix}.
$$

In this case, there are 2048 BNs in the SPBN problem.
By setting the stopping error $\varepsilon = 1.0 \times 10^{-3}$ and the sparsity regularization parameter $s = 20$, Algorithm 1 identifies 14 major BNs from 2048 BNs within 307 iterations. The reconstructed residual $\| \hat{P} - P \|_F = 5.1609 \times 10^{-4}$ and the cpu-time cost is 524.21 seconds. The identified major BNs are displayed in Figure 4.

![Figure 4. The probability distribution $x$ for the case $j = 3$ and 2048 BNs](image)

**Example 5.** There are $j = 4$ nodes, and each column of the observed transition-probability matrix $P$ has 1 or 2 non-zero entries. The matrix $P$ to be reconstructed with the sparse form as follows:

$$
\begin{align*}
P(1, 1) &= 1.0 & P(1, 4) &= 1.0 & P(1, 8) &= 0.5078 & P(1, 12) &= 0.5078 \\
P(2, 3) &= 0.5078 & P(2, 7) &= 0.5078 & P(2, 11) &= 0.5078 & P(2, 15) &= 0.5078 \\
P(5, 2) &= 1.0 & P(5, 6) &= 1.0 & P(5, 10) &= 0.5078 & P(5, 14) &= 0.5078 \\
P(6, 4) &= 0.5078 & P(6, 8) &= 0.5078 & P(6, 12) &= 0.5078 & P(6, 16) &= 0.5078 \\
P(9, 9) &= 0.4922 & P(9, 13) &= 0.4922 & P(10, 3) &= 0.4922 & P(10, 7) &= 0.4922 \\
P(10, 11) &= 0.4922 & P(10, 15) &= 0.4922 & P(13, 10) &= 0.4922 & P(13, 14) &= 0.4922 \\
P(14, 4) &= 0.4922 & P(14, 8) &= 0.4922 & P(14, 12) &= 0.4922 & P(14, 16) &= 0.4922
\end{align*}
$$

In this case, there are 4096 BNs in the SPBN problem.

By setting the stopping error $\varepsilon = 1.0 \times 10^{-3}$ and the regularization parameter $s = 40$, Algorithm 1 identifies 79 major BNs from 4096 BNs within 44 iterations. The reconstructed residual is $\| \hat{P} - P \|_F = 1.2 \times 10^{-4}$, and the cpu-time cost is 426.85 seconds. The identified major BNs are displayed in Figure 5.

In summary, the numerical results show that Algorithm 1 is valid for solving problem (15) resulted from the sparse probabilistic Boolean network model (13). Compared with the MEM and MMEM, the $\ell_1$ regularization model (13) and Algorithm 1 of this paper are more stable and efficient for the sparse probabilistic Boolean network reconstruction problem.

5. **Conclusions.** The sparse probabilistic Boolean networks model has many applications in management and industrial engineering fields. These applications includes real-time pricing systems, credit default data analysis, social networks for
Figure 5. The probability distribution $x$ for the case $j = 4$ and 4096 BNs

practical applications, genetic regulatory networks engineering, controllability in practical engineering, and so on.

This paper focuses on the sparse solution of probabilistic Boolean networks problem. We first reformulated the SPBN problem as a separable $\ell_\frac{1}{2}$ regularization optimization problem, and then proposed a partial proximal-type split augmented Lagrangian method (Algorithm 1) to solve the resulting problem. Under some mild conditions, the global convergence to a critical point of Algorithm 1 was established. By the proposed subproblem-solvers, all the iteration subproblems of Algorithm 1 are computationally tractable, and consequently the proposed method is efficient, even for large scale problem. Numerical results are presented to indicate that Algorithm 1 is numerically stable and efficient for the SPBN problem.

The $\ell_0$ regularization optimization model (5), and the $\ell_0$ constrained optimization model of the form

$$
\begin{align*}
\begin{cases}
\min_{x \in \mathbb{R}^n} & \frac{1}{2} \|Ax - b\|^2, \\
\text{s.t.} & \|x\|_0 \leq \kappa, \quad e^T x = 1, \quad x \geq 0,
\end{cases}
\end{align*}
$$

(45)

also have some important applications in sparse probabilistic Boolean network. The recent development of optimization technique, such as projected gradient methods proposed in [32], provides some useful approaches for problems (5) and (45) with high efficiency. However, from the view point of information representation, the regularization parameter $\tau$ in (5) and the sparsity parameter $\kappa$ in (45) are very sensitive to practical applications of the SPBN problem. Hence, models (5) and (45), and the corresponding computational algorithms, are interesting topics in the future works.

Acknowledgments. The first version of this work was completed during the second author visiting Prof. Y-H Dai of Academy of Mathematics and Systems Science, Chinese Academy of Sciences. The authors are grateful to Prof. D-R Han of Beihang University and Prof. X-M Yuan of Hong Kong University for any useful discussions. They also thank all the anonymous referees, whose suggestions improve the quality of this paper.
REFERENCES

[1] Y.-Q. Bai and K.-J. Shen, Alternating direction method of multipliers for $\ell_1-\ell_2$ regularized Logistic regression model, *Journal of the Operations Research Society of China*, 4 (2016), 245–253.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends in Machine Learning*, 3 (2010), 1–122.

[3] M. Caetano and T. Yoneyama, An autocatalytic network model for stock markets, *Physica A*, 419 (2015), 122–127.

[4] X. Chen, W.-K. Ching and X.-S. Chen, Construction of probabilistic boolean networks from a prescribed transition probability matrix: A maximum entropy rate approach, *East Asian Journal on Applied Mathematics*, 1 (2011), 132–154.

[5] X. Chen, H. Jiang and W.-K. Ching, Construction of sparse probabilistic boolean networks, *East Asian Journal of Applied Mathematics*, 2 (2012), 1–18.

[6] Y.-H. Dai, D.-R. Han, X.-M. Yuan and W.-X. Zhang, A sequential updating scheme of Lagrange multiplier for separable convex programming, *Mathematics of Computation*, 86 (2017), 315–343.

[7] J. Eckstein and M. Fukushima, Some reformulations and applications of the alternating direction method of multipliers, in *Large Scale Optimization: State of the Art* Springer US, (1994), 115–134.

[8] M. Fukushima, Application of the alternating direction method of multipliers to separable convex programming problems, *Computational Optimization and Applications*, 1 (1992), 93–111.

[9] J.-W. Gu, W.-K. Ching, T.-K. Siu and H. Zheng, On modeling credit defaults: A probabilistic Boolean network approach, *Risk and Decision Analysis*, 4 (2013), 119–129.

[10] D.-R. Han, X.-M. Yuan and W.-X. Zhang, An augmented Lagrangian based parallel splitting method for separable convex minimization with applications to image processing, *Mathematics of Computation*, 83 (2014), 2263–2291.

[11] B.-S. He and X.-M. Yuan, Alternating direction method of multipliers for linear programming, *Journal of the Operations Research Society of China*, 4 (2016), 425–436.

[12] B.-S. He, M. Tao and X.-M. Yuan, Alternating direction method with Gaussian back substitution for separable convex programming, *SIAM Journal on Optimization*, 22 (2012), 313–340.

[13] I. Ivanov, R. Pal and E.-R. Dougherty, Dynamics preserving size reduction mappings for probabilistic Boolean networks, *IEEE Transactions on Signal Processing*, 55 (2007), 2310–2322.

[14] K. Kobayashi and K. Hiraishi, An integer programming approach to optimal control problems in context-sensitive probabilistic Boolean networks, *Automatica*, 47 (2011), 1260–1264.

[15] K. Kobayashi and K. Hiraishi, A probabilistic approach to control of complex systems and its application to real-time pricing, *Mathematical Problems in Engineering*, 2014 (2014), Art. ID 906717, 8 pp.

[16] K. Kobayashi and K. Hiraishi, Verification of real-time pricing systems based on probabilistic Boolean networks, *Applied Mathematics*, 7 (2016), Article ID:70627,14 pages.

[17] J. Li, A. Ritter and D. Jurafsky, Inferring user preferences by probabilistic logical reasoning over social networks, preprint, *arXiv:1411.2679*.

[18] R. Liang, Y. Qiu and W.-K. Ching, Construction of probabilistic Boolean network for credit default data, Computational Sciences and Optimization (CSO), 2014 Seventh International Joint Conference on. IEEE, (2014), 11–15.

[19] 2003. Available from: [http://code.google.com/p/pbn-matlab-toolbox](http://code.google.com/p/pbn-matlab-toolbox).

[20] B.-K. Natrajan, Sparse approximation to linear systems, *SIAM Journal on Computing*, 24 (1995), 227–234.

[21] Z. Peng and D.-H. Wu, A partial parallel splitting augmented Lagrangian method for solving constrained matrix optimization problems, *Computers and Mathematics with Applications*, 60 (2010), 1515–1524.

[22] B.-E. Rhoades, S. Sessa, M.-S. Khan and M. Swaleh, On fixed points of asymptotically regular mappings, *Journal of the Australian Mathematical Society*, 43 (1987), 328–346.

[23] I. Shmulevich, E.-R. Dougherty, S. Kim and W. Zhang, Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks, *Bioinformatics*, 18 (2002), 261–274.
[24] I. Shmulevich, E.-R. Dougherty and W. Zhang, From Boolean to probabilistic Boolean networks as models of genetic regulatory networks, *Proceedings of the IEEE*, 90 (2002), 1778–1792.

[25] I. Shmulevich, E.-R. Dougherty and W. Zhang, Gene perturbation and intervention in probabilistic Boolean networks, *Bioinformatics*, 18 (2002), 1319–1331.

[26] B. Tian, X.-Q. Yang and K.-W. Meng, An interior-point $\ell_1$-penalty method for inequality constrained nonlinear optimization, *Journal of Industrial & Management Optimization*, 12 (2016), 949–973.

[27] X.-F. Wang and G. Chen, Complex networks: Small-world, scale-free and beyond, *IEEE Circuits and Systems Magazine*, 3 (2003), 6–20.

[28] Z.-M. Wu, X.-J. Cai and D.-R. Han, Linearized block-wise alternating direction method of multipliers for multiple-block convex programming, *Journal of Industrial & Management Optimization*, 14 (2018), 833–855.

[29] M.-H. Xu, Proximal alternating directions method for structured variational inequalities, *Journal of Optimization Theory and Applications*, 134 (2007), 107–117.

[30] Z.-B. Xu, H. Guo, Y. Wang and H. Zhang, The representation of $\ell_2$ regularizer among $\ell_q(0 < q < 1)$ regularizer: an experimental study based on phase diagram, *Acta Automatica Sinica*, 38 (2012), 1225–1228.

[31] Z.-B. Xu, X.-Y. Chang, F.-M. Xu and H. Zhang, $\ell_2$ regularization: a thresholding representation theory and a fast slover, *IEEE Transactions on Neural Networks and Learning Systems*, 23 (2012), 1013–1027.

[32] F.-M. Xu, Y.-H. Dai, Z.-H. Zhao and Z.-B. Xu, Efficient projected gradient methods for a class of $\ell_0$ constrained optimization problems, *Mathematical Programming*, to appear.

[33] J. Yang, Y.-Q. Dai, Z. Peng, J.-P. Zhuang and W.-X. Zhu, A homotopy alternating direction method of multipliers for linearly constrained separable convex optimization, *Journal of the Operations Research Society of China*, 5 (2017), 271–290.

[34] J. Zeng, S. Lin, Y. Wang and Z.-B. Xu, $\ell_1$ Regularization: convergence of iterative half thresholding algorithm, *IEEE Transactions on Signal Processing*, 62 (2014), 2317–2329.

[35] K.-Z. Zhang and L.-Z. Zhang, Controllability of probabilistic Boolean control networks with time-variant delays in states, *Science China: Information Sciences*, 59 (2016), 092204, 10pp.

Received January 2018; revised April 2018.

*E-mail address*: m175410002@fzu.edu.cn

*E-mail address*: pzheng@fzu.edu.cn

*E-mail address*: jlchen@fzu.edu.cn