International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

MAP Inference via ℓ_2-Sphere Linear Program Reformulation

Baoyuan Wu · Li Shen · Tong Zhang · Bernard Ghanem

the date of receipt and acceptance should be inserted later

Abstract Maximum a posteriori (MAP) inference is an important task for graphical models. Due to complex dependencies among variables in realistic models, finding an exact solution for MAP inference is often intractable. Thus, many approximation methods have been developed, among which the linear programming (LP) relaxation based methods show promising performance. However, one major drawback of LP relaxation is that it is possible to give fractional solutions. Instead of presenting a tighter relaxation, in this work we propose a continuous but equivalent reformulation of the original MAP inference problem, called LS-LP. We add the ℓ_2-sphere constraint onto the original LP relaxation, leading to an intersected space with the local marginal polytope that is equivalent to the space of all valid integer label configurations. Thus, LS-LP is equivalent to the original MAP inference problem. We propose a perturbed alternating direction method of multipliers (ADMM) algorithm to optimize the LS-LP problem, by adding a sufficiently small perturbation ϵ onto the objective function and constraints. We prove that the perturbed ADMM algorithm globally converges to the ϵ-Karush–Kuhn–Tucker (ϵ-KKT) point of the LS-LP problem. The convergence rate will also be analyzed. Experiments on several benchmark datasets from Probabilistic Inference Challenge (PIC 2011) and OpenGM 2 show competitive performance of our proposed method against state-of-the-art MAP inference methods.

Baoyuan Wu was partially supported by Tencent AI Lab and King Abdullah University of Science and Technology (KAUST). Li Shen was supported by Tencent AI Lab. Bernard Ghanem was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research through the Visual Computing Center (VCC) funding. Tong Zhang was supported by the Hong Kong University of Science and Technology (HKUST). Li Shen is the corresponding author.

Baoyuan Wu
Tencent AI Lab, Shenzhen 518000, China
E-mail: wubaoyuan1987@gmail.com

Li Shen
Tencent AI Lab, Shenzhen 518000, China
E-mail: mathshenli@gmail.com

Tong Zhang
Hong Kong University of Science and Technology, Hong Kong, China
E-mail: tongzhang@tongzhang-ml.org

Bernard Ghanem
King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia
E-mail: bernard.ghanem@kaust.edu.sa

1 Introduction

Given the probability distribution of a graphical model, maximum a posteriori (MAP) inference aims to infer the most probable label configuration. MAP inference can be formulated as an integer linear program (ILP) [39]. However, due to the integer constraint, the exact optimization of ILP is intractable in many realistic problems. To tackle it, a popular approach is relaxing ILP to a continuous linear program over a local marginal polytope, *i.e.*, $\mathcal{L}_G$ (defined in Section 3), called linear programming (LP) relaxation. The optimal solution to the LP relaxation will be obtained at the vertices of $\mathcal{L}_G$. It has been known [39] that all valid integer label configurations are at the vertices of $\mathcal{L}_G$, but not all vertices of $\mathcal{L}_G$ are integer, while some are fractional. Since LP relaxation is likely to give fractional solutions, the rounding method must be adopted to generate integer solutions. To alleviate this issue, intense efforts have been made to design tighter relaxations (*e.g.*, high-order relaxation [36]) based on LP relaxation, such that the proportion of fractional vertices of $\mathcal{L}_G$ can be reduced. However, the possibility of fractional solutions still exists. And, these tighter relaxations are often much more computationally expensive than the original LP relaxation. Moreover, there are also exact inference meth-

ods, such as branch-and-bound [23] and cutting-plane [17], which utilize LP relaxation as sub-routines, leading to much higher computational cost than approximate methods.

Instead of proposing a new approximation with a tighter relaxation, we propose an exact reformulation of the original MAP inference problem. Specifically, we add a new constraint, called ℓ_2-sphere [41], onto the original LP relaxation problem. It enforces that the solution $\mathbf{x} \in \mathbb{R}^n$ should be on a ℓ_2-sphere, *i.e.*, $\| \mathbf{x} - \frac{1}{2} \|_2^2 = \frac{n}{4}$. We can prove that the intersection between the ℓ_2-sphere constraint and the local polytope $\mathcal{L}_G$ is equivalent to the set of all possible label configurations of the original MAP inference problem, *i.e.*, the constraint space of the ILP problem. Thus, the proposed formulation, dubbed LS-LP, is an equivalent but continuous reformulation of the ILP formulation for MAP inference. Furthermore, inspired by [28] and [41], we adopt the ADMM algorithm [6], to not only separate the different constraints, but also decompose variables to allow parallel inference by exploiting the factor graph structure. Although the ℓ_2-sphere constraint is non-convex, we prove that the ADMM algorithm for the LS-LP problem with a sufficiently small perturbation ϵ will globally converges to the ϵ-KKT [16,22] point of the original LS-LP problem. The obvious advantages of the proposed LS-LP formulation and the corresponding ADMM algorithm include: **1)** compared to other LP relaxation based methods, our method directly gives the valid integer label configuration, without any rounding techniques as post-processing; **2)** compared to the exact methods like branch-and-bound [23] and cutting-plane [17], our method optimizes one single continuous problem once, rather than multiple times. Experiments on benchmarks from Probabilistic Inference Challenge (PIC 2011) [7] and OpenGM 2 [14] verify the competitive performance of LS-LP against state-of-the-art MAP inference methods.

The main contributions of this work are three-fold. **1)** We propose a continuous but equivalent reformulation of the MAP inference problem. **2)** We present the ADMM algorithm for optimizing the perturbed LS-LP problem, which is proved to be globally convergent to the ϵ-KKT point of the original LS-LP problem. The analysis of convergence rate is also presented. **3)** Experiments on benchmark datasets verify the competitive performance of our method compared to state-of-the-art MAP inference methods.

2 Related Work

As our method is closely related to LP relaxation based MAP inference methods, here we mainly review MAP inference methods of this category. For other categories of methods, such as message passing and move making, we refer the readers to [39] and [14] for more details. Although some off-the-shelf LP solvers can be used to optimize the LP relaxation problem, in many real-world applications the problem scale is too large to adopt these solvers. Hence, most methods focus on developing efficient algorithms to optimize the dual LP problem. Block coordinate descent methods [9,19] are fast, but they may converge to sub-optimal solutions. Sub-gradient based methods [20,15] can converge to global solutions, but their convergence is slow. Their common drawback is the non-smoothness of the dual objective function. To handle this difficulty, some smoothing methods have been developed. The Lagrangian relaxation [12] method uses the smooth log-sum-exp function to approximate the non-smooth max function in the dual objective. A proximal regularization [13] or an ℓ_2 regularization term [30] is added to the dual objective. Moreover, the steepest ϵ-descent method proposed in [34] and [35] can accelerate the convergence of the standard sub-gradient based methods. Parallel MAP inference methods based on ADMM have also been developed to handle large-scale inference problems. For example, AD3 [28,27] and Bethe-ADMM [8] optimize the primal LP problem, while ADMM-dual [29] optimizes the dual LP problem. The common drawback of these methods is that they are likely to produce fractional solutions, since the underlying problem is merely a relaxation to the MAP inference problem.

Another direction is pursuing tighter relaxations, such as high-order consistency [36] and SDP relaxation [24]. But they are often more computationally expensive than LP relaxations. In contrast, the formulation of the proposed LS-LP is an exact reformulation of the original MAP inference problem, and the adopted ADMM algorithm can explicitly produce valid integer label configurations, without any rounding operation. In comparison with other expensive exact MAP inference methods (*e.g.*, Branch-and-Bound [23] and cutting plane [17]), LS-LP is very efficient owing to the resulting parallel inference, similar to other ADMM based methods.

Another related work is ℓ_p-Box ADMM [41], which is a framework to optimize the general integer program. The proposed LS-LP is inspired by this framework, where the integer constraints are replaced by the intersection of two continuous constraints. However, **1)** LS-LP is specifically designed for MAP inference, as it replaces the valid integer configuration space (*e.g.*, $\{(0,1),(1,0)\}$ for the variable with binary states), rather than the whole binary space (*e.g.*, $\{(0,0),\ (0,1),(1,0),(1,1)\}$) as did in ℓ_p-Box ADMM. **2)** LS-LP is tightly combined with LP relaxation, and the ADMM algorithm decomposes the problem into multiple simple sub-problems by utilizing the structure of the factor graph, which allows parallel inference for any type of inference problems (*e.g.*, multiple variable states and high-order factors). In contrast, ℓ_p-Box ADMM does not assume any special property for the objective function, and it optimizes all variable nodes in one sub-problem. Especially for large-scale models, the sub-problem involved in ℓ_p-Box ADMM will be very cost. **3)** As LP relaxation is parameterized according to the factor

graph, any type of graphical models (*e.g.*, directed models, high-order potentials, asymmetric potentials) can be naturally handled by LS-LP. In contrast, ℓ_p-Box ADMM needs to transform the inference objective based on MRF models to some simple forms (*e.g.*, binary quadratic program (BQP)). However, the transformation is non-trivial in some cases. For example, if there are high-order potentials, the graphical model is difficult to input into a BQP problem.

3 Background

3.1 Factor Graph

Denote $\mathbf{G} = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_N\}$ as a set of N random variables in a discrete space $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_N$, where $\mathcal{X}_i = \{0, \dots, r_i - 1\}$ with $r_i = |\mathcal{X}_i|$ being the number of possible states of $\mathbf{g}_i$. The joint probability of $\mathbf{G}$ is formulated based on a factor graph G [18],

$$P(\mathbf{G}) \propto \exp\big(\sum_{i\in V} \boldsymbol{\theta}_i(\mathbf{g}_i) + \sum_{\alpha\in F} \boldsymbol{\theta}_\alpha(\mathbf{g}_\alpha)\big), \tag{1}$$

where $G = (V, F, E)$ with $V = \{1, \dots, N\}$ being the node set of variables, F being the node set of factors, as well as the edge set $E \subseteq V \times F$ linking the variable and factor nodes. A simple MRF model and its factor graph are shown in Fig. 1(a,b). We refer the readers to [18] for the detailed definition of the factor graph. $\mathbf{g}_\alpha$ indicates the label configuration of the factor α, and its state will be determined according to the states of connected variable nodes, *i.e.*, $\{\mathbf{g}_i | i \in \mathcal{N}_\alpha\}$, with $\mathcal{N}_\alpha$ being the set of neighborhood variable nodes of the factor α. $\boldsymbol{\theta}_i(\cdot)$ denotes the unary log potential (logPot) function, while $\boldsymbol{\theta}_\alpha(\cdot)$ indicates the factor logPot function.

3.2 MAP Inference as Linear Program

Given $P(\mathbf{G})$, an important task is to find the most probable label configuration of $\mathbf{G}$, referred to as MAP inference,

$$\text{MAP}(\boldsymbol{\theta}) = \max_{\mathbf{G}\in\mathcal{X}} \sum_{i\in V} \boldsymbol{\theta}_i(\mathbf{g}_i) + \sum_{\alpha\in F} \boldsymbol{\theta}_\alpha(\mathbf{g}_\alpha). \tag{2}$$

Eq. (2) can be reformulated as the integer linear program (ILP) [39],

$$\text{ILP}(\boldsymbol{\theta}) = \max_{\boldsymbol{\mu}} \sum_{i\in V} \boldsymbol{\theta}_i^\top \boldsymbol{\mu}_i + \sum_{\alpha\in F} \boldsymbol{\theta}_\alpha^\top \boldsymbol{\mu}_\alpha = \max_{\boldsymbol{\mu}} \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle, \tag{3}$$
$$\text{s.t. } \boldsymbol{\mu} \in \mathcal{L}_G \cap \{0,1\}^{|\boldsymbol{\mu}|},$$

where $\boldsymbol{\theta} = (\dots; \boldsymbol{\theta}_i; \dots; \boldsymbol{\theta}_\alpha; \dots), i \in V, \alpha \in F$ denotes the log potential (logPot) vector, derived from $\boldsymbol{\theta}_i(\mathbf{g}_i)$ and $\boldsymbol{\theta}_\alpha(\mathbf{g}_\alpha)$. $\boldsymbol{\mu} = [\boldsymbol{\mu}_V; \boldsymbol{\mu}_F]$, where $\boldsymbol{\mu}_V = [\boldsymbol{\mu}_1; \dots; \boldsymbol{\mu}_{|V|}]$ and $\boldsymbol{\mu}_F = [\boldsymbol{\mu}_1; \dots; \boldsymbol{\mu}_{|F|}]$. $\boldsymbol{\mu}_i \in \{0,1\}^{|\mathcal{X}_i|}$ indicates the label vector corresponding to $\mathbf{g}_i$: if the state of $\mathbf{g}_i$ is t, then $\boldsymbol{\mu}_i(t) = 1$, while all other entries are 0. Similarly, $\boldsymbol{\mu}_\alpha \in \{0,1\}^{|\mathcal{X}_\alpha|}$ indicates the label vector corresponding to $\mathbf{g}_\alpha$. The local marginal polytope is defined as follows,

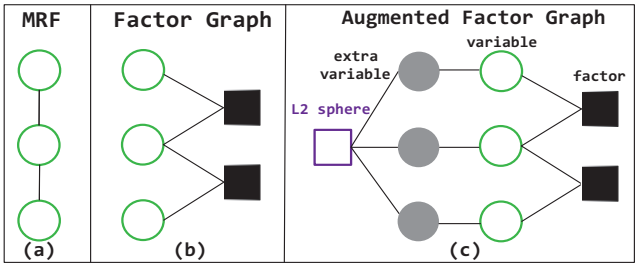

Fig. 1: An example of (a) MRF, (b) factor graph corresponding to LP and (c) augmented factor graph corresponding to LS-LP.

$$\mathcal{L}_G = \big\{\boldsymbol{\mu} \big| \boldsymbol{\mu}_\alpha \in \Delta^{|\boldsymbol{\mu}_\alpha|}, \forall \alpha \in F; \quad \boldsymbol{\mu}_i = \mathbf{M}_{i\alpha}\boldsymbol{\mu}_\alpha,\ \forall (i,\alpha) \in E\big\}. \tag{4}$$

with $\Delta^{|\mathbf{a}|} = \{\mathbf{a} | \mathbf{1}^\top \mathbf{a} = 1, \mathbf{a} \geq \mathbf{0}\}$ being the probability simplex, and the second constraint ensures the local consistency between $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_\alpha$. $\mathbf{M}_{i\alpha} \in \{0,1\}^{|\mathcal{X}_i| \times |\mathcal{X}_\alpha|}$ of the local consistency constraint included in $\mathcal{L}_G$ is defined as: the entry of $\mathbf{M}_{i\alpha}$ is 1 if $\mathbf{g}_\alpha \sim \mathbf{g}_i$, where $\mathbf{g}_\alpha \sim \mathbf{g}_i$ indicates the state of $\mathbf{g}_i$ and the state of the corresponding element in $\mathbf{g}_\alpha$ are the same; otherwise, the entry is 0. For example, we consider a binary-state variable node $\boldsymbol{\mu}_i \in \{0,1\}^2$ and a pairwise factor node $\boldsymbol{\mu}_\alpha \in \{0,1\}^4$ connected to two variable nodes (the variable node i is the first). The first entry of $\boldsymbol{\mu}_i$ indicates the score of choosing state 0, while the second entry corresponds to that of choosing state 1. The four entries of $\boldsymbol{\mu}_\alpha$ indicate the scores of four label configurations of two connected variables, *i.e.*, $(0,0), (0,1), (1,0), (1,1)$. In this case, $\mathbf{M}_{i\alpha} = [1,1,0,0;0,0,1,1]$.

Moreover, Eq. (2) can also be rewritten as

$$\text{MAP}(\boldsymbol{\theta}) = \max_{\boldsymbol{\mu}\in\mathcal{M}_G} \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle, \tag{5}$$

where the marginal polytope is defined as follows,

$$\mathcal{M}_G = \{\boldsymbol{\mu} \mid \exists P(X), \text{ such that } \boldsymbol{\mu}_i, \boldsymbol{\mu}_\alpha \in \mathcal{L}_G\}. \tag{6}$$

Solving MAP($\boldsymbol{\theta}$) is difficult (NP-hard in general), especially for large scale problems. Instead, the approximation over $\mathcal{L}_G$ is widely adopted, as follows:

$$\text{LP}(\boldsymbol{\theta}) = \max_{\boldsymbol{\mu}\in\mathcal{L}_G} \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle \geq \text{ILP}(\boldsymbol{\theta}) = \text{MAP}(\boldsymbol{\theta}), \tag{7}$$

which is called LP relaxation. Note that here $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_\alpha$ are continuous variables, and they are considered as local marginals of $\mathbf{g}_i$ and $\mathbf{g}_\alpha$, respectively.

According to [39], the characteristics of LP($\boldsymbol{\theta}$), MAP($\boldsymbol{\theta}$) and their relationships are briefly summarized in Lemma 1.

Lemma 1 *[39] The relationship between $\mathcal{M}_G$ and $\mathcal{L}_G$, and that between MAP($\boldsymbol{\theta}$) and LP($\boldsymbol{\theta}$) are as follows.*

- $\mathcal{M}_G \subseteq \mathcal{L}_G$;
- *MAP*($\boldsymbol{\theta}$) $\leq$ *LP*($\boldsymbol{\theta}$);
- *All vertices of $\mathcal{M}_G$ are integer, while $\mathcal{L}_G$ includes both integer and fractional vertices. And the set of integer vertices of $\mathcal{L}_G$ is same with the set of the vertices of $\mathcal{M}_G$. All non-vertices in $\mathcal{M}_G$ and $\mathcal{L}_G$ are fractional points.*
- *Since both $\mathcal{M}_G$ and $\mathcal{L}_G$ are convex polytopes, the global solutions of MAP($\boldsymbol{\theta}$) and LP($\boldsymbol{\theta}$) will be on the vertices of $\mathcal{M}_G$ and $\mathcal{L}_G$, respectively.*
- *The global solution $\boldsymbol{\mu}^*$ of LP($\boldsymbol{\theta}$) can be fractional or integer. If it is integer, then it is also the global solution of MAP($\boldsymbol{\theta}$).*

3.3 Kurdyka-Lojasiewicz Inequality

The Kurdyka-Lojasiewicz inequality was firstly proposed in [26], and it has been widely used in many recent works [2, 40, 25] for the convergence analysis of non-convex problems. Since it will also be used in the later convergence analysis of our algorithm, it is firstly produced here, as shown in Definition 1.

Definition 1 [2] A function $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup +\infty$ is said to have the Kurdyka-Lojasiewicz (KL) property at $x^* \in \text{dom}(\partial f)$ (dom$(\cdot)$ denotes the domain of function, ∂ indicates the sub-gradient operator), if the following two conditions hold

- there exist a constant $\eta \in (0, +\infty]$, a neighborhood $\mathcal{V}_{x^*}$ of x^*, as well as a continuous concave function $\varphi : [0, \eta) \rightarrow \mathbb{R}_+$, with $\varphi(0) = 0$ and φ is differentiable on $(0, \eta)$ with positive derivatives.
- $\forall x \in \mathcal{V}_{x^*}$ satisfying $f(x^*) < f(x) < f(x^*) + \eta$, the Kurdyka-Lojasiewicz inequality holds

$$\varphi'(f(x) - f(x^*))\text{dist}(0, \partial f(x)) \geq 1. \tag{8}$$

Remark. According to [2,4,5], if f is semi-algebraic, then it satisfies the KL property with $\varphi(s) = cs^{1-p}$, where $p \in [0, 1)$ and $c > 0$ are constants. This point will be used in later analysis of convergence.

4 MAP Inference via ℓ_2-sphere Linear Program Reformulation

4.1 Equivalent Reformulation

Firstly, we introduce the ℓ_2-sphere constraint [41],

$$\mathcal{S} = \big\{\mathbf{x} \in \mathbb{R}^n \mid \ \| \mathbf{x} - \frac{1}{2}\mathbf{1} \|_2^2 = \frac{n}{4}\big\}. \tag{9}$$

Note that $\mathcal{S}$ is defined with respect to the vector $\mathbf{x}$, rather than individual scalars $x_i, i = 1, \ldots, n$. We propose to add the ℓ_2-sphere constraint onto the variable nodes $\boldsymbol{\mu}_V$. Combining this with LP relaxation (see Eq. (7)), we propose a new formulation for MAP inference,

$$\text{LS-LP}(\boldsymbol{\theta}) = \max_{\boldsymbol{\mu}} \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle, \ \text{ s.t. } \boldsymbol{\mu} \in \mathcal{L}_G, \boldsymbol{\mu}_V \in \mathcal{S}. \tag{10}$$

Due to the non-convex constraint $\mathcal{S}$, it is no longer a linear program. However, to emphasize its relationship to LP relaxation, we still denote it as a ℓ_2-sphere constrained linear program (LS-LP) reformulation. More importantly, as shown in Proposition 1, LS-LP is equivalent to the original MAP inference problem, rather than a relaxation as in LP. Inspired by the constraint separation in ℓ_p-Box ADMM [41], we introduce the extra variable $\boldsymbol{\upsilon}$ to reformulate (10) as

$$\begin{aligned} &\text{LS-LP}(\boldsymbol{\theta}) = \max_{\boldsymbol{\mu}, \boldsymbol{\upsilon}} \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle = \min_{\boldsymbol{\mu}, \boldsymbol{\upsilon}} \langle -\boldsymbol{\theta}, \boldsymbol{\mu} \rangle, \\ &\text{s.t. } \boldsymbol{\mu} \in \mathcal{L}_G, \boldsymbol{\upsilon} \in \mathcal{S}, \boldsymbol{\mu}_i = \boldsymbol{\upsilon}_i, \forall i \in V, \end{aligned} \tag{11}$$

where $\boldsymbol{\upsilon} = [\boldsymbol{\upsilon}_1; \ldots; \boldsymbol{\upsilon}_i; \ldots; \boldsymbol{\upsilon}_{|V|}], i \in V$ is the concatenated vector of all extra variable nodes. The combination of the original factor graph and these extra variable nodes is referred to as *augmented factor graph* (AFG). An example of AFG corresponding to Problem (11) is shown in Figure 1(c). The gray circles correspond to extra variables $\boldsymbol{\upsilon}$, and connections to the purple box indicate that $\boldsymbol{\upsilon} \in \mathcal{S}$. Note that AFG does not satisfy the definition of the standard factor graph, where connections only exist between variables nodes and factor nodes. However, AFG provides a clear picture of the structure of LS-LP and the node relationships. The proposed LS-LP problem is equivalent to the original MAP inference problem, as shown in Proposition 1. It means that the global solutions of this two problems are equivalent.

Lemma 2 *The following constraint spaces are equivalent,*

$$\begin{aligned} &\mathcal{C}_1 = \{\boldsymbol{\mu} \mid \boldsymbol{\mu} \in \mathcal{L}_G \cap \{0, 1\}^{|\boldsymbol{\mu}|}\} \\ \equiv &\mathcal{C}_2 = \{\boldsymbol{\mu} \mid \boldsymbol{\mu} \in \mathcal{L}_G \textit{ and } \boldsymbol{\mu}_V \in \mathcal{S}\} \\ \equiv &\mathcal{C}_3 = \{\boldsymbol{\mu} \mid \boldsymbol{\mu} \in \mathcal{M}_G \cap \{0, 1\}^{|\boldsymbol{\mu}|}\}. \end{aligned} \tag{12}$$

Proof We start from $\mathcal{C}_2$, where we have

$$\boldsymbol{\mu}_V \in \mathcal{S} \Longleftrightarrow \sum_{i \in V} \| \boldsymbol{\mu}_i - \frac{1}{2} \|_2^2 = \frac{\sum_{i \in V} |\mathcal{X}_i|}{4}. \tag{13}$$

Besides, the following relations hold

$$\begin{aligned} &\boldsymbol{\mu} \in \mathcal{L}_G \Longleftrightarrow \boldsymbol{\mu}_\alpha \in \Delta^{|\mathcal{X}_\alpha|} \text{ and } \boldsymbol{\mu}_i = \mathbf{M}_{i\alpha} \boldsymbol{\mu}_\alpha \\ \Rightarrow &\boldsymbol{\mu}_i \in [0, 1]^{|\mathcal{X}_i|} \Rightarrow \| \boldsymbol{\mu}_i - \frac{1}{2} \|_2^2 \leq \frac{|\mathcal{X}_i|}{4}, \end{aligned} \tag{14}$$

$\forall i \in V, \forall (i,\alpha) \in E$. The equation in the last relation holds if and only if $\boldsymbol{\mu}_i \in \{0,1\}^{|\mathcal{X}_i|}$. Combining with (13), we conclude that $\boldsymbol{\mu}_i \in \{0,1\}$ holds $\forall i \in V$. Consequently, utilizing the local consistency constraint $\boldsymbol{\mu}_i = \mathbf{M}_{i\alpha}\boldsymbol{\mu}_\alpha$, we obtain that $\boldsymbol{\mu}_\alpha \in \{0,1\}$ also holds $\forall \alpha \in F$. Thus, we have $\boldsymbol{\mu} \in \{0,1\}^{|\boldsymbol{\mu}|}$. Then, the relation $\mathcal{C}_1 \equiv \mathcal{C}_2$ is proved.

Besides, as shown in Lemma 1, the set of integer vertices of $\mathcal{L}_G$ is same with the one of $\mathcal{M}_G$, and all non-vertices in $\mathcal{M}_G$ and $\mathcal{L}_G$ are fractional points. Thus, it is easy to know $\mathcal{C}_1 \equiv \mathcal{C}_3$. Hence the proof is finished.

Theorem 1 *Utilizing Lemma 2, the aforementioned MAP inference problems have the following relationships,*

$$\textit{LS-LP}(\boldsymbol{\theta}) = \textit{ILP}(\boldsymbol{\theta}) = \textit{MAP}(\boldsymbol{\theta}) \leq \textit{LP}(\boldsymbol{\theta}). \tag{15}$$

Proof According to Lemma 1.3 and 1.4, as well as $\mathcal{C}_2 \equiv \mathcal{C}_3$ in Lemma 2 (see Eq. (12)), we have

$$\max_{\boldsymbol{\mu}\in\mathcal{M}_G} \langle \boldsymbol{\theta}, \boldsymbol{\mu}\rangle = \max_{\boldsymbol{\mu}\in\mathcal{C}_3} \langle \boldsymbol{\theta}, \boldsymbol{\mu}\rangle = \max_{\boldsymbol{\mu}\in\mathcal{C}_2} \langle \boldsymbol{\theta}, \boldsymbol{\mu}\rangle \tag{16}$$

$$\Longleftrightarrow \text{MAP}(\boldsymbol{\theta}) = \text{LS-LP}(\boldsymbol{\theta}). \tag{17}$$

Combining with $\text{MAP}(\boldsymbol{\theta}) = \text{ILP}(\boldsymbol{\theta}) \leq \text{LP}(\boldsymbol{\theta})$ (see Eq. (7)), the proof is finished.

4.2 A General Form and KKT Conditions

For clarity, we firstly simplify the notations and formulations in Eq. (11) to the general shape,

$$\text{LS-LP}(\boldsymbol{\theta}) = \min_{\mathbf{x},\mathbf{y}} f(\mathbf{x}) + h(\mathbf{y}), \text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{B}\mathbf{y}. \tag{18}$$

Our illustration for (18) is separated into three parts, as follows:

1. **Variables**. $\mathbf{x} = [\boldsymbol{\mu}_1; \ldots; \boldsymbol{\mu}_{|V|}] \in \mathbb{R}^{\sum_i^V |\mathcal{X}_i|}$, and it concatenates all variable nodes $\boldsymbol{\mu}_V$. $\mathbf{y} = [\mathbf{y}_1; \ldots; \mathbf{y}_{|V|}]$ with $\mathbf{y}_i = [\boldsymbol{\upsilon}_i; \boldsymbol{\mu}_{\alpha_{i,1}}; \ldots; \boldsymbol{\mu}_{\alpha_{i,|\mathcal{N}_i|}}] \in \mathbb{R}^{|\mathcal{X}_i| + \sum_\alpha^{\mathcal{N}_i} |\mathcal{X}_\alpha|}$. $\mathbf{y}$ concatenates all factor nodes $\boldsymbol{\mu}_V$ and the extra variable nodes $\boldsymbol{\upsilon}$; $\mathbf{y}_i$ concatenates the factor nodes and the extra variable node connected to the i-th variable node $\boldsymbol{\mu}_i$. $\mathcal{N}_i$ indicates the set of neighborhood factor nodes connected to the i-th variable node; the subscript $\alpha_{i,j}$ indicates the j-th factor connected to the i-th variable, with $i \in V$ and $j \in \mathcal{N}_i$.
2. **Objective functions**. $f(\mathbf{x}) = \mathbf{w}_\mathbf{x}^\top \mathbf{x}$ with $\mathbf{w}_\mathbf{x} = -[\boldsymbol{\theta}_1; \ldots; \boldsymbol{\theta}_{|V|}]$. $h(\mathbf{y}) = g(\mathbf{y}) + \mathbf{w}_\mathbf{y}^\top \mathbf{y}$, with $\mathbf{w}_\mathbf{y} = [\mathbf{w}_1; \ldots; \mathbf{w}_{|V|}]$ with $\mathbf{w}_i = -[\mathbf{0}; \frac{1}{|\mathcal{N}_{\alpha_{i,1}}|}\boldsymbol{\theta}_{\alpha_{i,1}}; \ldots; \frac{1}{|\mathcal{N}_{\alpha_{i,|\mathcal{N}_i|}}|}\boldsymbol{\theta}_{\alpha_{i,|\mathcal{N}_i|}}]$, and $\mathcal{N}_\alpha = \{i \mid (i,\alpha) \in E\}$ being the set of neighborhood variable nodes connected to the α-th factor. $g(\mathbf{y}) = \mathbb{I}(\boldsymbol{\upsilon} \in \mathcal{S}) + \sum_{\alpha\in F} \mathbb{I}(\boldsymbol{\mu}_\alpha \in \Delta^{|\mathcal{X}_\alpha|})$, with $\mathbb{I}(a)$ being the indicator function: $\mathbb{I}(a) = 0$ if a is true, otherwise $\mathbb{I}(a) = \infty$.
3. **Constraint matrices**. The constraint matrix $\mathbf{A} = \text{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_i, \ldots, \mathbf{A}_{|V|})$ with $\mathbf{A}_i = [\mathbf{I}_{|\mathcal{X}_i|}; \ldots; \mathbf{I}_{|\mathcal{X}_i|}] \in \{0,1\}^{(|\mathcal{N}_i|+1)|\mathcal{X}_i| \times |\mathcal{X}_i|}$. $\mathbf{B} = \text{diag}(\mathbf{B}_1, \ldots, \mathbf{B}_i, \ldots, \mathbf{B}_{|V|})$, with $\mathbf{B}_i = \text{diag}(\mathbf{I}_{|\mathcal{X}_i|}, \mathbf{M}_{i,\alpha_{i,1}}, \ldots, \mathbf{M}_{i,\alpha_{i,|\mathcal{N}_i|}})$. $\mathbf{A}$ summarizes all constraints on $\boldsymbol{\mu}_V$, while $\mathbf{B}$ collects all constraints on $\boldsymbol{\mu}_F$ and $\boldsymbol{\upsilon}$.

Note that Problem (18) has a clear structure with two groups of variables, corresponding the augmented factor graph (see Fig. 1(c)).

Definition 2 The solution $(\mathbf{x}^*, \mathbf{y}^*)$ of the LS-LP problem (18) is said to be the KKT point if the following conditions are satisfied:

$$\mathbf{B}^\top \boldsymbol{\lambda}^* \in \partial h(\mathbf{y}^*),\ \nabla f(\mathbf{x}^*) = -\mathbf{A}^\top \boldsymbol{\lambda}^*,\ \mathbf{A}\mathbf{x}^* = \mathbf{B}\mathbf{y}^*, \tag{19}$$

where $\boldsymbol{\lambda}^*$ denotes the Lagrangian multiplier; ∂h indicates the sub-gradient of h, while ∇f represents the gradient of f. Moreover, $(\mathbf{x}^*, \mathbf{y}^*)$ is considered as the ϵ-KKT point if the following conditions hold:

$$\begin{aligned}&\text{dist}(\mathbf{B}^\top \boldsymbol{\lambda}^*, \partial h(\mathbf{y}^*)) \leq O(\epsilon),\ \|\nabla f(\mathbf{x}^*) + \mathbf{A}^\top \boldsymbol{\lambda}^*\| \leq O(\epsilon),\\ &\|\mathbf{A}\mathbf{x}^* - \mathbf{B}\mathbf{y}^*\| \leq O(\epsilon).\end{aligned} \tag{20}$$

5 Perturbed ADMM Algorithm for LS-LP

We propose a perturbed ADMM algorithm to optimize the following perturbed augmented Lagrangian function,

$$\begin{aligned}\mathcal{L}_{\rho,\epsilon}(\mathbf{y}, \hat{\mathbf{x}}, \boldsymbol{\lambda}) =& \hat{f}(\hat{\mathbf{x}}) + h(\mathbf{y}) + \boldsymbol{\lambda}^\top(\hat{\mathbf{A}}\hat{\mathbf{x}} - \mathbf{B}\mathbf{y})\\ &+ \frac{\rho}{2}\|\hat{\mathbf{A}}\hat{\mathbf{x}} - \mathbf{B}\mathbf{y}\|_2^2,\end{aligned} \tag{24}$$

where $\hat{\mathbf{A}} = [\mathbf{A}, \epsilon\mathbf{I}]$ with a sufficiently small constant $\epsilon > 0$, then $\hat{\mathbf{A}}$ is full row rank. $\hat{\mathbf{x}} = [\mathbf{x}; \bar{\mathbf{x}}]$, with $\bar{\mathbf{x}} = [\bar{\mathbf{x}}_1; \ldots; \bar{\mathbf{x}}_{|V|}] \in$

Algorithm 1 The perturbed ADMM algorithm

Input: The initialization $\mathbf{y}^0, \hat{\mathbf{x}}^0, \boldsymbol{\lambda}^0$, the perturbation ϵ, the hyper-parameter ρ

1: **for** $k = 0$ to K **do**:
2: Update $\mathbf{y}^{k+1}$ as follows (see Section 5.1 for details)

$$\mathbf{y}^{k+1} = \operatorname*{argmin}_{\mathbf{y}} \mathcal{L}_{\rho,\epsilon}(\mathbf{y}, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k) \tag{21}$$

3: Update $\hat{\mathbf{x}}^{k+1}$ as follows (Section 5.2 for details)

$$\hat{\mathbf{x}}^{k+1} = \operatorname*{argmin}_{\hat{\mathbf{x}}} \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1}, \hat{\mathbf{x}}, \boldsymbol{\lambda}^k) \tag{22}$$

4: Update $\boldsymbol{\lambda}^{k+1}$ (see Section 5.3 for details)

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\hat{\mathbf{A}}\hat{\mathbf{x}}^{k+1} - \mathbf{B}\mathbf{y}^{k+1}) \tag{23}$$

5: Check stopping criterion, as shown in Section 5.4
6: **end for**
7: **return** $\mathbf{y}^*, \mathbf{x}^*$

$\mathbb{R}^{\sum_i^V (|\mathcal{N}_i|+1)|\mathcal{X}_i|}$ and $\bar{\mathbf{x}}_i = [\boldsymbol{\mu}_i; \ldots; \boldsymbol{\mu}_i] \in \mathbb{R}^{(|\mathcal{N}_i|+1)|\mathcal{X}_i|}$. $\hat{f}(\hat{\mathbf{x}}) = f(\mathbf{x}) + \frac{1}{2}\epsilon\hat{\mathbf{x}}^\top\hat{\mathbf{x}}$. Note that both $\hat{\mathbf{A}}$ and $\mathbf{B}$ are full row rank, and the second-order gradient $\nabla^2\hat{f}(\hat{\mathbf{x}}) = \epsilon\mathbf{I}$ is bounded. These properties will play key roles in our later analysis of convergence.

Following the conventional ADMM algorithm, the solution to the LS-LP problem (18) can be obtained through optimizing the following sub-problems based on (24) iteratively. The general structure of the algorithm is summarized in Algorithm 1.

5.1 Sub-Problem w.r.t. $\mathbf{y}$ in LS-LP Problem

Given $\hat{\mathbf{x}}^k$ and $\boldsymbol{\lambda}^k$, $\mathbf{y}^{k+1}$ can be updated by solving the sub-problem (21) (see Algorithm 1). According to the definitions of $\hat{\mathbf{A}}, \hat{\mathbf{x}}, \mathbf{B}, \mathbf{y}$, this problem can be further separated to the following two independent sub-problems, which can be solved in parallel.

Update $\boldsymbol{\upsilon}^{k+1}$:

$$\min_{\boldsymbol{\upsilon}\in\mathcal{S}} \sum_{i\in V} \big[-(\boldsymbol{\lambda}_i^k)^\top\boldsymbol{\upsilon}_i + \frac{\rho_i}{2} \parallel (1+\epsilon)\boldsymbol{\mu}_i^k - \boldsymbol{\upsilon}_i \parallel_2^2 \big]. \tag{25}$$

It has a closed form solution as follows

$$\boldsymbol{\upsilon}^{k+1} = \mathcal{P}_{\mathcal{S}}(\overline{\boldsymbol{\upsilon}}^{k+1}), \tag{26}$$

where $\overline{\boldsymbol{\upsilon}}^{k+1} = [\boldsymbol{\upsilon}_1^{k+1}; \ldots; \boldsymbol{\upsilon}_{|V|}^{k+1}]$ with $\boldsymbol{\upsilon}_i^{k+1} = (1+\epsilon)\boldsymbol{\mu}_i^k + \frac{1}{\rho_i}\boldsymbol{\lambda}_i^k$. $\mathcal{P}_{\mathcal{S}}(\cdot)$ is the projection onto $\mathcal{S}$: $\mathcal{P}_{\mathcal{S}}(\mathbf{a}) = \frac{n^{1/2}}{2} \times \frac{\bar{\mathbf{a}}}{\|\bar{\mathbf{a}}\|_2} + \frac{1}{2}\mathbf{1}_n$, with $\bar{\mathbf{a}} = \mathbf{a} - \frac{1}{2}\mathbf{1}_n$ and n being the dimension of $\mathbf{a}$. As demonstrated in [41], this projected solution is the optimal solution to (25).

Update $\boldsymbol{\mu}_\alpha^{k+1}$: The sub-problems w.r.t. $\{\boldsymbol{\mu}_\alpha\}_{\alpha\in F}$ can be run in parallel $\forall\alpha \in F$,

$$\begin{aligned}\min_{\boldsymbol{\mu}_\alpha\in\Delta^{|\mathcal{X}_\alpha|}} \; & -\boldsymbol{\theta}_\alpha^\top\boldsymbol{\mu}_\alpha + \sum_{i\in\mathcal{N}_\alpha} \big[\frac{\rho_{i\alpha}}{2} \parallel (1+\epsilon)\boldsymbol{\mu}_i^k - \mathbf{M}_{i\alpha}\boldsymbol{\mu}_\alpha \parallel_2^2 \\ & - (\boldsymbol{\lambda}_{i\alpha}^k)^\top\mathbf{M}_{i\alpha}\boldsymbol{\mu}_\alpha\big]. \end{aligned} \tag{27}$$

It is easy to know that Problem (27) is convex, as $\mathbf{M}_{i\alpha}^\top\mathbf{M}_{i\alpha}$ is positive semi-definite and $\Delta^{|\mathcal{X}_\alpha|}$ is a convex set. Any off-the-shelf QP solver can be adopted to solve (27). In experiments, we adopt the active-set algorithm implemented by a publicly-available toolbox called Quadratic Programming in C (QPC)[1], which is written in C and can be called from MATLAB.

[1] http://sigpromu.org/quadprog/download.php?sid=3wtwk5tb

5.2 Sub-Problem w.r.t. $\hat{\mathbf{x}}$ in LS-LP Problem

Given $\mathbf{y}^{k+1}$ and $\boldsymbol{\lambda}^k$, $\hat{\mathbf{x}}^{k+1}$ can be updated by solving the sub-problem (22) (see Algorithm 1). According to the definition of $\hat{\mathbf{x}}$, this problem can be separated to $|V|$ independent sub-problems w.r.t. $\{\boldsymbol{\mu}_i\}_{i\in V}$, as follows:

$$\begin{aligned}&\min_{\boldsymbol{\mu}_i} \; (\boldsymbol{\lambda}_i^k - \boldsymbol{\theta}_i)^\top\boldsymbol{\mu}_i + \frac{\epsilon(|\mathcal{N}_i|+2)}{2}\boldsymbol{\mu}_i^\top\boldsymbol{\mu}_i + \sum_{\alpha\in\mathcal{N}_i}\Big[\\ &(1+\epsilon)(\boldsymbol{\lambda}_{i\alpha}^k)^\top\boldsymbol{\mu}_i + \frac{\rho_{i\alpha}}{2} \parallel (1+\epsilon)\boldsymbol{\mu}_i - \mathbf{M}_{i\alpha}\boldsymbol{\mu}_\alpha^{k+1} \parallel_2^2 \Big] \\ &+ \frac{\rho_i}{2} \parallel (1+\epsilon)\boldsymbol{\mu}_i - \boldsymbol{\upsilon}_i \parallel_2^2 \end{aligned} \tag{28}$$

$$\begin{aligned}&= (1+\epsilon)\Big[\sum_{\alpha\in\mathcal{N}_i}\big(\boldsymbol{\lambda}_{i\alpha}^k - \rho_{i\alpha}\mathbf{M}_{i\alpha}\boldsymbol{\mu}_\alpha^{k+1}\big) - \rho_i\boldsymbol{\upsilon}_i^{k+1} - \boldsymbol{\theta}_i \\ &+ \boldsymbol{\lambda}_i^k\Big]^\top\boldsymbol{\mu}_i + \boldsymbol{\mu}_i^\top\mathbf{Q}\boldsymbol{\mu}_i + \text{const}, \end{aligned} \tag{29}$$

where $\mathbf{Q} = \frac{1}{2}[\epsilon(|\mathcal{N}_i|+2) + \rho_i(1+\epsilon)^2 + \sum_{\alpha\in\mathcal{N}_i}\rho_{i\alpha}(1+\epsilon)^2] \cdot \mathbf{I}$. The above sub-problem can be further simplified to $\min_{\boldsymbol{\mu}_i} a\cdot\boldsymbol{\mu}_i^\top\boldsymbol{\mu}_i + \mathbf{b}^\top\boldsymbol{\mu}_i$, where a and $\mathbf{b}$ can be easily derived from the above equation. Its close-form solution is obtained by setting its gradient to $\mathbf{0}$, *i.e.*, $\boldsymbol{\mu}_i^{k+1} = \frac{\mathbf{b}}{2a}$.

5.3 Update $\boldsymbol{\lambda}$ in LS-LP Problem

Given $\mathbf{y}^{k+1}$ and $\hat{\mathbf{x}}^{k+1}$, $\boldsymbol{\lambda}^{k+1}$ is updated using (23) (see Algorithm 1). Similarly, it can be separately to $|V|+|E|$ independent sub-problems, as follows

$$\boldsymbol{\lambda}_i^{k+1} = \boldsymbol{\lambda}_i^k + \rho_i[(1+\epsilon)\boldsymbol{\mu}_i^{k+1} - \boldsymbol{\upsilon}_i^{k+1}], \tag{30}$$

$$\boldsymbol{\lambda}_{i\alpha}^{k+1} = \boldsymbol{\lambda}_{i\alpha}^k + \rho_{i\alpha}[(1+\epsilon)\boldsymbol{\mu}_i^{k+1} - \mathbf{M}_{i\alpha}\boldsymbol{\mu}_\alpha^{k+1}], \tag{31}$$

where $i \in V, (i,\alpha) \in E$.

5.4 Complexity and Implementation Details

Complexity. In terms of computational complexity, as all other update steps have simple closed-form solutions, the main computational cost lies in updating $\boldsymbol{\mu}_\alpha$, which is convex quadratic programming with the probability simplex constraint. Its computational complexity is $O(|\mathcal{X}_\alpha|^3)$. As the matrix with the largest size is $\mathbf{M}_{i\alpha}^\top\mathbf{M}_{i\alpha} \in \mathbb{R}^{|\mathcal{X}_\alpha|\times|\mathcal{X}_\alpha|}$ in LS-LP, the space complexity is $\mathcal{O}(\sum_{\alpha\in F}(|\mathcal{X}_\alpha|)^2)$. Both the computational and space complexity of AD3 are similar with LS-LP. More detailed analysis about the computational complexity will be presented in Section 7.5.

Implementation details. In each iteration, we use the same value of ρ for all ρ_i and $\rho_{i\alpha}$. After each iteration, we update ρ using an incremental rate η, *i.e.*, $\rho \leftarrow \rho \times \eta$. A upper limit ρ_{upper} of ρ is also set: if ρ is larger than ρ_{upper}, it is not updated anymore. The perturbation ϵ is set to 10^{-5}, and ρ_{upper}

can be set as any constant than $\frac{1}{\epsilon}$, such as 2×10^5. We utilize two stopping criterion jointly, including: 1) the violation of the local consistency constraint, *i.e.*, $(\sum_{(i,\alpha)\in E}\frac{\rho_{i\alpha}}{2}\parallel(1+\epsilon)\boldsymbol{\mu}_i-\mathbf{M}_{i\alpha}\boldsymbol{\mu}_\alpha\parallel_2^2)^{\frac{1}{2}}$; 2) the violation of the equivalence constraint $(1+\epsilon)\boldsymbol{\mu}_i=\boldsymbol{\upsilon}_i$, *i.e.*, $(\sum_{i\in V}\frac{\rho_i}{2}\parallel(1+\epsilon)\boldsymbol{\mu}_i-\boldsymbol{\upsilon}_i\parallel_2^2)^{\frac{1}{2}}$. We set the same threshold 10^{-5} for both criterion. If this two violations are lower than 10^{-5} simultaneously, then the algorithm stops.

6 Convergence Analysis

The convergence property of the above ADMM algorithm is demonstrated in Theorem 2. Due to the space limit, the detailed proof will be presented in **Appendix** A.

Theorem 2 *We suppose that ρ is set to be larger than a constant, then the variable sequence $\{\mathbf{y}^k,\hat{\mathbf{x}}^k,\boldsymbol{\lambda}^k\}$ generated by the perturbed ADMM algorithm globally converges to $(\mathbf{y}^*,\hat{\mathbf{x}}^*,\boldsymbol{\lambda}^*)$, where $(\mathbf{y}^*,\mathbf{x}^*)$ is the ϵ-KKT point to the LS-LP problem (18), as defined in Definition 2.*

Furthermore, according to Definition 1, we assume that $\mathcal{L}_{\rho,\epsilon}$ has the KL property at $(\mathbf{y}^*,\mathbf{x}^*,\boldsymbol{\lambda}^*)$ with the concave function $\varphi(s)=cs^{1-p}$, where $p\in[0,1),c>0$. Consequently, we can obtain the following inequalities:

(i) If $p=0$, then the perturbed ADMM algorithm will converge in finite steps.
(ii) If $p\in(0,\frac{1}{2}]$, then we will obtain the ϵ-KKT solution to the LS-LP problem in at least $O\big(\log_{\frac{1}{\tau}}(\frac{1}{\epsilon})^2\big)$ steps, with $\tau\in(0,1)$ being a small constant, which will be later defined in Appendix A.5.
(iii) If $p\in(\frac{1}{2},1)$, then we will obtain the ϵ-KKT solution to the LS-LP problem in at least $O\big((\frac{1}{\epsilon})^{\frac{4p-2}{1-p}}\big)$ steps.

Proof The general structure of the proof consists of the following steps, as follows:

1. The perturbed augmented Lagrangian function $\mathcal{L}_{\rho,\epsilon}$ (see (24)) is monotonically decreasing along the optimization.
2. The variable sequence $\{\mathbf{y}^k,\hat{\mathbf{x}}^k,\boldsymbol{\lambda}^k\}$ is bounded.
3. The sequence of variable residuals is converged, *i.e.*, $\{\|\mathbf{y}^{k+1}-\mathbf{y}^k\|,\|\hat{\mathbf{x}}^{k+1}-\hat{\mathbf{x}}^k\|,\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|\}\to0$, as $k\to\infty$.
4. The variable sequence $\{\mathbf{y}^k,\hat{\mathbf{x}}^k,\boldsymbol{\lambda}^k\}$ globally converges to the cluster point $(\mathbf{y}^*,\hat{\mathbf{x}}^*,\boldsymbol{\lambda}^*)$.
5. $(\mathbf{y}^*,\mathbf{x}^*)$ is the ϵ-KKT point of the LS-LP problem (18).
6. We finally analyze the convergence rate that how many steps are required to achieve the ϵ-KKT point.

Table 1: Benchmark datasets used in the Probabilistic Inference Challenge (PIC 2011) [7] and OpenGM 2 [14]. C_1 to C_7 represent: number of models, average variables, average factors, average edges, average factor sizes (*i.e.*, the average number of adjacent variables for each factor), average variable states, average factor states.

dataset	C_1	C_2	C_3	C_4	C_5	C_6	C_7
Seg-2	50	229.14	622.28	1244.56	2	2	4
Seg-21	50	229.14	622.28	1244.56	2	21	441
Scene	715	182.56	488.99	977.98	2	8	64
Grids	21	3142.86	6236.19	12472.4	2	2	4
Protein	7	14324.7	21854.7	57680.4	2.64	2	6.56

7 Experiments

7.1 Experimental Settings

7.1.1 Datasets

We evaluate on four benchmark datasets from the Probabilistic Inference Challenge (PIC 2011) [7] and OpenGM 2 [14], including Segmentation [7] , Scene [10], Grids [7], and Protein [11], as shown in Table 1. Segmentation consists of Seg-2 and Seg-21, with different variable states. Protein includes higher-order potentials, while others include pairwise potentials.

7.1.2 Compared Methods

We compare with different categories of MAP inference methods, including: **1)** moving making methods, *i.e.*, ICM [3]; **2)** message-passing methods, including belief propagation (BP) [21] and TRBP [38]; **3)** polyhedral methods (including LP relaxation based methods), including dual decomposition using sub-gradient (DD-SG) [15], TRWS [19], AD-Sal [33], PSDD [20] and AD3 [27][28]. **4)** We also compare with LP-LP, which calls the the active-set method (implemented by *linprog* in MATLAB) to optimize LP($\boldsymbol{\theta}$). It serves as a baseline to measure the performance of above methods. **5)** The most related work ℓ_2-Box ADMM (*i.e.*., the special case of ℓ_2-Box ADMM with $p=2$) algorithm [41] is also compared. However, the presented algorithm in [41] can only handle MRF models with pairwise potentials, which is formulated as a binary quadratic programming (BQP) problem. Thus, ℓ_2-Box ADMM (hereafter we call it ℓ_2-Box for clarity) is not compared on Protein, of which models include high-order potentials. **6)** We also compare with two hybrid methods, including method DAOOPT (adopting branch-and-bound method [23] as a sub-routine) [31][32] and MPLP-C [37] (adopting MPLP [9] as a sub-routine). The 'hybrid' indicates that the method is a combination of an off-the-shelf single method and some heuristic

steps. And we call above 5 types as non-hybrid methods. Both the proposed LS-LP and ℓ_2-Box are implemented by MATLAB. The following methods are implemented by the author provided C++ package, including: PSDD and AD3[2], MPLP-C[3], and DAOOPT[4]. All other methods are implemented through the OpenGM 2 software [14], and we add a prefix "ogm" before the method name, such as ogm-TRWS.

In experiments, we set some upper limits: the maximal iteration as 2000 for PSDD and AD3, 500 for ℓ_2-Box and LS-LP, and 1000 for other methods; for DAOOPT, the memory limit of mini buckets is set as 4000 MB and the upper time limit as 2 hours. The parameter tuning of all compared methods (except ℓ_2-Box) is self-included in their implementations. Both LS-LP and ℓ_2-Box are ADMM algorithms, and their hyper-parameters are tuned as follows: the hyper-parameters ρ_0, η and ρ_{upper} (see implementation details of Section 5) are adjusted in the ranges $\{0.05, 0.1, 1, 5, 10, 10^2, 10^3, 10^4\}$, $\{1.01, 1.03, 1.05, 1.1, 1.2\}$ and $\{10^6, 10^8\}$, respectively, and those leading to the higher logPot value are used.

7.1.3 Evaluation Metrics

We evaluate the performance of all compared methods using three types of metrics, including the log potential (logPot) values, the solution type, as well as the computational complexity and runtime.

Evaluation using logPot values. The logPot value indicates the objective value $\langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle$ of MAP($\boldsymbol{\theta}$) (see Eq. (5)). Given that the constraint $\mathcal{M}_G$ of MAP($\boldsymbol{\theta}$) is satisfied, the larger logPot value indicates the better inference performance. Since LP-LP gives the optimal solution to LP($\boldsymbol{\theta}$) (see Eq. (7)) constrained to $\mathcal{L}_G$, and we know that $\mathcal{M}_G \subseteq \mathcal{L}_G$, then the logPot value of any valid label configurations cannot be larger than that of LP-LP. Note that in the implementation of OpenGM 2 [14], a rounding method is adopted as the post-processing step to produce the integer solution for the continuous MAP inference methods. However, the performance of different MAP inference methods may be significantly changed by rounding. Thus, for other methods not implemented by OpenGM 2, we report the logPot values of original continuous solutions, without any rounding.

Evaluation using solution types. Since LP-LP, PSDD and AD3 are possible to give continuous solutions, the larger logPot value doesn't always mean the better MAP inference result. Thus, we also define four qualitative measures, including *valid, uniform, fractional* and *approximate*, to intuitively measure the inference quality. *Valid* (V) means that the solution is integer and satisfies the constraints in $\mathcal{L}_G$; *Uniform* (U) denotes that the solution belongs to $\mathcal{L}_G$, but the value is uniform, such as $(0.5, 0.5)$ for the variable node with binary states. *Fractional* (F) indicates that the solution belongs to $\mathcal{L}_G$, but its value is fractional, while not uniform; *Approximate* (A) means that some constraints in $\mathcal{L}_G$ are violated, and its solutions is integer or fractional. Note that it makes sense to compare the logPot values for different solutions, only when the solutions are valid. In contrast, it makes no sense to compare the logPot values, if the solutions belong to the other three types of measures. To better illustrate above measures, we present a brief example on a toy graphical model, as shown in Table 2.

Table 2: A brief illustration of four types of measures for inference quality, on a toy graphical model with two connected variable nodes.

Possible states	Variable 1 $\{0,1\}$	Variable 2 $\{0,1\}$	Factor $\{00,01,10,11\}$	Measure
Inferred Probability	$(1,0)$	$(0,1)$	$(0,1,0,0)$	Valid
	$(0.5,0.5)$	$(0.5,0.5)$	$(0.25,0.25,0.25,0.25)$	Uniform
	$(0.2,0.8)$	$(0.4,0.6)$	$(0.08,0.12,0.32,0.48)$	Fractional
	$(0.2,0.8)$	$(0.4,0.6)$	$(0.16,0.3,0.4,0.14)$	Approximate

Evaluation using the computational complexity and practical runtime. The computational complexity and the practical runtime are also important performance measures for MAP inference methods, as shown in Section 7.5.

7.2 Results on Segmentation and Scene

The average results on Seg-2, Seg-21 and Scene are shown in Table 3. LP-LP gives valid solutions on all models, *i.e.*, the best solutions. Except for PSDD, all other methods give valid solutions, and their logPot values can not be higher than those of LP-LP. The logPot values of ICM are the lowest, and those of ogm-BP, ogm-TRBP are slightly lower than the best logPot values, while other methods achieve the best logPot values on most models. Only PSDD gives approximate solutions (*i.e.*, the constraints in $\mathcal{L}_G$ are not fully satisfied) on some models, specifically, 5 models in Seg-2, 8 models in Seg-21 and 166 models in Scene. ogm-DD-SG fails to give solutions on some models of these datasets, thus we ignore it. Evaluations on these easy models only show that the performance ranking is ogm-ICM < ogm-BP, ogm-TRBP, ℓ_2-Box < others.

7.3 Results on Grids

The results on Grids are shown in Table 4. For clarity, we use the model indexes M1 to M21 to indicate the model name to save space in this section. The corresponding model names from M1 to M21 are grid20x20.f10.uai, grid20x20.f10.wrap.uai, grid20x20.f15.uai, grid20x20.f15.wrap.uai, grid20x20.f5.wrap.uai, grid40x40.f10.uai, grid40x40.f10.wrap.uai, grid

[2] http://www.cs.cmu.edu/ ark/AD3/

[3] https://github.com/opengm/MPLP

[4] https://github.com/lotten/daoopt

Table 3: LogPot values of MAP inference solutions on Seg-2, Seg-21 and Scene. Except of PSDD, all other methods give valid solutions. The best result among valid solutions in each row is highlighted in bold. Please refer to Section 7.2 for details.

Method type →		Baseline	Hybrid methods		Non-hybrid methods								Proposed
Dataset ↓		LP-LP	DAOOPT	MPLP-C	ogm-ICM	ogm-BP	ogm-TRBP	ogm-TRWS	ogm-ADSal	PSDD	AD3	ℓ_2-Box	LS-LP
Seg-2	mean	**-75.5**	**-75.5**	**-75.5**	-137.1	-79	-76.8	**-75.5**	**-75.5**	-75.4	**-75.5**	-76.5	-75.6
	std	19.63	19.63	19.63	70.1	20.24	19.36	19.24	19.24	19.77	19.63	20.3	19.69
Seg-21	mean	**-324.89**	-325.34	**-324.89**	-393.37	-330.37	-328.92	**-324.89**	**-324.89**	-325.1	**-324.89**	-344.51	**-324.89**
	std	58.12	58.14	58.12	74.47	58.54	58.57	56.97	56.97	58.16	58.12	59.24	58.12
Scene	mean	**866.66**	**866.66**	**866.66**	864.27	866.49	866.51	**866.66**	**866.66**	866.65	**866.66**	864.11	**866.66**
	std	109.34	109.34	109.36	109.64	109.22	109.2	109.19	109.19	109.34	109.34	108.66	109.34

Table 4: MAP inference results on Grids dataset. LP-LP, PSDD and AD3 produce uniform solutions on all models in Grids, while all other methods give valid solutions. Here we only show the logPot of LP-LP as the upper bound of other methods. The best logPot among integer solutions in each row is highlighted in bold. The number with in circle indicates the performance ranking of each method. Please refer to Section 7.3 for details.

Method type →	Baseline	Hybrid methods		Non-hybrid methods							Proposed
Model ↓	LP-LP	DAOOPT	MPLP-C	ogm-ICM	ogm-BP	ogm-TRBP	ogm-DD-SG	ogm-TRWS	ogm-ADSal	ℓ_2-Box	LS-LP
M1	3736.7	**3015.7** ①	**3015.7** ①	2708.9 ⑤	121.3 ⑨	-235.2 ⑩	1286.3 ⑧	2524.9 ⑦	2605.2 ⑥	2794.8 ④	2931.8 ③
M2	3830.3	**3051** ①	3033.6 ②	2567.9 ⑦	276.4 ⑨	19.2 ⑩	1484.7 ⑧	2674.4 ⑤	2670.2 ⑥	2812.4 ④	2936.7 ③
M3	5605.1	**4517.3** ①	**4517.3** ①	4067.3 ⑤	332.1 ⑨	14.02 ⑩	1889.7 ⑧	3829.3 ⑦	3884 ⑥	4301.1 ④	4408.9 ③
M4	5745.5	**4563.2** ①	**4563.2** ①	3837.12 ⑦	924.5 ⑨	-36.7 ⑩	2023.4 ⑧	3894.6 ⑥	4015 ⑤	4202.4 ④	4446.6 ③
M5	1915.2	**1542.7** ①	**1542.7** ①	1318.41 ⑦	481.5 ⑨	-47.8 ⑩	807.6 ⑧	1325.5 ⑤	1323.9 ⑥	1427.4 ④	1503.2 ③
M6	15601.2	12662.9 ②	**12665.7** ①	10753.7 ⑥	2793.5 ⑨	2214.3 ⑩	5051.9 ⑧	10500.8 ⑦	11029 ⑤	11486.2 ④	12336.1 ③
M7	16291.5	13050.7 ②	**13054.8** ①	10903.8 ⑤	1217.1 ⑨	132.4 ⑩	4634.8 ⑧	10665 ⑦	10870.4 ⑥	11867.6 ④	12537.2 ③
M8	23401.8	**18952.45** ①	18896.8 ①	16154.2 ⑥	4314.9 ⑩	5371.1 ⑨	7160 ⑧	16014 ⑦	16276.9 ⑤	17367.5 ④	18358.7 ③
M9	24437.3	**19538** ①	19427.5 ②	16334.2 ⑥	3560.8 ⑨	-1111 ⑩	7187.3 ⑧	16004.3 ⑦	16508.1 ⑤	17990 ④	18785.8 ③
M10	3121.2	**2689** ①	2688.8 ②	2255.38 ⑥	1665.3 ⑧	1582.9 ⑨	1330.7 ⑩	2215.7 ⑦	2369.1 ⑤	2552.6 ④	2659.8 ③
M11	3231.6	**2714.67** ①	2714.52 ②	2258.54 ⑦	1399.6 ⑧	42.8 ⑩	1285.9 ⑨	2271.5 ⑥	2370.1 ⑤	2556.8 ④	2654.9 ③
M12	7800.6	**6401.15** ①	6396 ②	5356.28 ⑥	2033.5 ⑨	1953.1 ⑩	2832.5 ⑧	5282.5 ⑦	5558.8 ⑤	5903 ④	6201.2 ③
M13	8078.5	**6472.9** ①	6469.7 ②	5425.16 ⑦	1711.5 ⑨	381 ⑩	2814.3 ⑧	5452.8 ⑥	5646.1 ⑤	5923.5 ④	6275.4 ③
M14	62943	–	45813.6 ②	43538.9 ④	5690.9 ⑨	6426.7 ⑧	18700.4 ⑦	42274.2 ⑥	43292.5 ⑤	44397.5 ③	**48766.1** ①
M15	63993.1	–	47444.4 ②	42855 ⑤	4287.1 ⑧	956.4 ⑨	18811.9 ⑦	42535 ⑥	42918.7 ④	44759.5 ③	**48657.3** ①
M16	94414.5	–	69408.6 ②	65081.2 ④	4374.2 ⑨	4656.5 ⑧	27320.6 ⑦	63148.1 ⑥	64401.1 ⑤	66784.2 ③	**72993.8** ①
M17	96243.6	–	71730.8 ②	63768.1 ⑥	13662.7 ⑧	-529.3 ⑨	27287.7 ⑦	63885.1 ⑤	64487.9 ④	67589.4 ③	**73486** ①
M18	12721.3	–	10445.8 ②	9062.03 ⑤	5198.7 ⑦	4975.4 ⑧	4785.5 ⑨	8793.5 ⑥	9408.4 ④	10015.1 ③	**10580.8** ①
M19	12875.6	–	10674.1 ②	9214.57 ⑤	5944.6 ⑦	1213.1 ⑨	5328.5 ⑧	8952.4 ⑥	9385.1 ④	10163.6 ③	**10698.4** ①
M20	31809.7	–	22292.5 ③	21527.9 ⑥	5410.9 ⑧	4762 ⑨	9837.3 ⑦	21546.8 ⑤	22109.5 ④	22913.3 ②	**24834.5** ①
M21	31996.9	–	24032.4 ②	21529.6 ⑤	4242.3 ⑧	47.6 ⑨	10423.8 ⑦	21195.9 ⑥	21730.3 ④	22668.7 ③	**24532.8** ①

40x40.f15.uai, grid40x40.f15.wrap.uai, grid40x40.f2.uai, grid 40x40.f2.wrap.uai, grid40x40.f5.uai, grid40x40.f5.wrap.uai, grid80 x80.f10.uai, grid80x80.f10.wrap.uai, grid80x80.f15.uai, grid 80x80.f15.wrap.uai, grid80x80.f2.uai, grid80x80.f2.wrap. uai, grid80x80.f5.uai, grid80x80.f5.wrap.uai, respectively.

The models in Grids are much challenging for LP relaxation based methods, as all models have symmetric pairwise log potentials and very dense cycles in the graph. In this case, many vertices of $\mathcal{L}_G$ are uniform solutions $(0.5, 0.5)$. Consequently, the LP relaxation based methods are likely to produce uniform solutions. This is verified by that LP-LP, AD3, PSDD give uniform solutions on all models in Grids, *i.e.*, most solutions are 0.5. Thus, we only show the logPot values of LP-LP in Table 4, to provide the theoretical upper-bound of logPot of valid solutions from other methods. In contrast, the additional ℓ_2-sphere constraint in LS-LP excludes the uniform solutions. On small scale models M1 to M13, DAOOPT and MPLP-C show the highest logPot values, while LS-LP gives slightly lower values. On large scale models M14 to M21, DAOOPT fails to give any result within 2 hours. LS-LP gives the best results, while MPLP-C shows slightly lower results. ℓ_2-Box performs worse than LS-LP, MPLP-C and DAOOPT on most models, while better than all other methods, among which ogm-BP, ogm-TRBP and ogm-DD-SG perform worst. These results demonstrate that **1)** LS-LP is comparable to hybrid methods DAOOPT and MPLP-C, but with much lower computational cost (shown in Section 7.5); **2)** LS-LP performs much better than other non-hybrid methods.

Table 5: LogPot values of MAP inference solutions on Protein dataset. Except for PSDD and AD3, all other methods give integer solutions. Both PSDD and AD3 produce mixed types of solutions on all models. The best result among valid solutions in each row is highlighted in bold. The number with in circle indicates the performance ranking of each method. Please refer to Section 7.4 for details.

Method type →	Hybrid methods	Non-hybrid methods						Proposed
Model ↓	MPLP-C	ogm-ICM	ogm-BP	ogm-TRBP	ogm-DD-SG	PSDD	AD3	LS-LP
M1	-30181.3 ②	-32409.9 ⑤	-32019.1 ④	-31671.6 ③	-33381.2 ⑥	-30128.8	-30143.6	**-30165.5** ①
M2	-29305.4 ②	-32561.3 ⑤	-30966.1 ③	-31253.3 ④	-33583.6 ⑥	-29307.3	-29302.6	**-29295.4** ①
M4	-28952.1 ②	-32570 ⑤	-31031.4 ③	-31176.6 ④	-33747.7 ⑥	-28952.5	-28952	**-28952** ①
M5	-269567 ③	**-256489** ①	-382766 ⑤	-357330 ④	-553376 ⑥	-66132.3	-115562	-267814 ②
M6	-30070.6 ②	-31699.1 ⑤	-30765.2 ③	-30772.2 ④	-32952.9 ⑥	-30063.6	-30062.2	**-30063.4** ①
M7	-30288.3 ②	-32562.2 ⑤	-31659.6 ③	-31791.1 ④	-33620.4 ⑥	-30248.5	-30239.8	**-30266** ①
M8	-29336.5 ②	-32617.2 ⑤	-31064.7 ③	-31219.9 ④	-34549.9 ⑥	-29331	-29336.1	**-29334.7** ①

7.4 Results on Protein

The results on Protein are shown in Table 5. Different with above three datasets, Protein includes 8 large scale models, and with high-order factors. Similarly, we use the model indexes M1 to M8 to indicate the model name to save space in this section. The corresponding model names of M1 to M8 are didNotconverge1.uai, didNotconverge2.uai, didNotconverge4.uai, didNotconverge5.uai, didNotconverge6.uai, didNotconverge7.uai, didNotconverge8.uai, respectively. As M1 and M3 are the same model, we remove M3 in experiments. DAOOPT fails to give solutions within 2 hours on all models, and LP-LP cannot produce solutions due to the memory limit. ogm-TRWS and ℓ_2-Box are not evaluated as it cannot handle high-order factors. LS-LP produces valid integer solutions on all models, and gives the highest logPot values on all models except of M5. MPLP-C gives slightly lower logPot values than LS-LP. AD3 only produces a fractional solution on M4, while produces approximate and fractional solutions on other models, while PSDD gives approximate and fractional solutions on all models. Specifically, the solution types of AD3 on M1 to M8 are: $A+2.01\%U+21.52F$; $A+0.55\%U+0.66F$; $0.17\%U$; $A+9.51\%U+32.44\%F$; $A+0.74\%U+13.7\%F$; $A+3.41\%U+17.16\%F$; $A+0.38\%U+0.13\%F$. Those of PSDD are: $A+6.96\%U+11.75\%F$; $A+1.47\%U+3.19\%F$; $A+0.73\%U+2.06\%F$; $A+13.72\%U+19.67\%F$; $A+4.35\%U+7.9\%F$; $A+6.28\%U+10.4\%F$; $A+0.68\%U+1.89\%F$. Other methods also show much worse performance than LS-LP and MPLP-C. One exception is that ogm-ICM gives the best results on M5, and we find that M5 is the most challenging model for approximated methods.

Table 6: Computational complexities of all compared methods. Excluding $\mathcal{E}$, the definitions of all other notations can be found in Section 3. $\mathcal{E}$ denotes the edge set of the original MRF graph, while E indicates the edge set of the corresponding factor graph. T represents the number of iterations.

Methods	Complexities
MPLP	$O\big(\sum_i^V \lvert\mathcal{N}_i\rvert^2 \cdot \lvert\mathcal{X}_i\rvert + 2\sum_{(i,j)}^{\mathcal{E}}(\lvert\mathcal{X}_i\rvert + \lvert\mathcal{X}_j\rvert)\big)$
MPLP-C	$O\big(T_{\text{outer}}\big[T_{\text{inner}}O(\text{MPLP}) + \lvert\mathcal{E}\rvert\big]\big)$
ogm-ICM	$O\big(T[\sum_i^V \lvert\mathcal{X}_i\rvert]\big)$
ogm-BP ogm-TRBP	$O\big(T\big[\sum_i^V(\lvert\mathcal{N}_i\rvert - 1)\sum_\alpha^{\mathcal{N}_i}\lvert\mathcal{X}_\alpha\rvert + \sum_\alpha^F(\lvert\mathcal{N}_\alpha\rvert - 1)\sum_i^{\mathcal{N}_\alpha}\lvert\mathcal{X}_i\rvert\big]\big)$
ogm-TRWS ogm-ADSal	$O\big(T\big[\lvert\mathcal{E}\rvert \cdot \max_{i\in V}\lvert\mathcal{X}_i\rvert\big]\big)$
PSDD AD3	$O\big(T\big[\sum_i^V[\lvert\mathcal{N}_i\rvert\cdot\lvert\mathcal{X}_i\rvert] + \sum_\alpha^F\lvert\mathcal{X}_\alpha\rvert^3 + \sum_{(i,\alpha)}^E\lvert\mathcal{X}_i\rvert\cdot\lvert\mathcal{X}_\alpha\rvert\big]\big)$
ℓ_2-Box	$O\big(T[\sum_i^V\lvert\mathcal{X}_i\rvert]^3\big)$
LS-LP	$O\big(T\big[\sum_i^V\lvert\mathcal{X}_i\rvert + \sum_\alpha^F\lvert\mathcal{X}_\alpha\rvert^3 + \sum_{(i,\alpha)}^E\lvert\mathcal{X}_i\rvert\cdot\lvert\mathcal{X}_\alpha\rvert\big]\big)$

7.5 Comparisons on Computational Complexities and Practical Runtime

Computational complexities of all compared methods (except of LP-LP and DAOOPT) are summarized in Table 6. As there is no clear conclusion of the complexity of the active-set algorithm for linear programming, the complexity of LP-LP is not presented. DAOOPT [32] is combination of 6 sequential sub-algorithms and heuristic steps, thus its computational complexity cannot be computed. As these complexities depend on the graph structure (*i.e.*, $V, E, \mathcal{E}, F, \mathcal{N}_i$, $\mathcal{N}_\alpha, \mathcal{X}_i, \mathcal{X}_\alpha$), it is impossible to give a fixed ranking of them. However, it is notable that the complexity of LS-LP is linear w.r.t. the number of variables, factors and edges (*i.e.*, $|V|, |F|, |E|$). Moreover, as the sub-problems w.r.t. variables/factors/edges in LS-LP are independent, they can be solved in parallel. However, the complexity of LS-LP is super-linear w.r.t. the state space $|\mathbf{X}_\alpha|$ and $|\mathbf{X}_i|$. Thus, the proposed LS-LP method is suitable for graphical models with very large-

Table 7: Iterations and practical runtime on Seg-2, Seg-21 and Scene.

Datasets			LP-LP	DAOOPT	MPLP-C	ogm-ICM	ogm-BP	ogm-TRBP	ogm-TRWS	ogm-ADSal	PSDD	AD3	ℓ_2-Box	LS-LP
Seg-2	iters	mean	9.4	–	1.3	2.12	1000	1000	18.74	16.96	632.5	70.12	78.84	84.14
		std	1.03	–	0.463	0.32	0	0	15.67	8.55	618.2	76.79	100.6	49.58
	runtime	mean	0.096	0.54	0.033	0.007	36.51	40.73	0.191	0.7828	0.032	0.001	0.849	12.53
		std	0.01	0.504	0.054	0.001	0.61	0.67	0.167	0.6373	0.027	0.006	1.044	7.39
Seg-21	iters	mean	16.13	–	1.46	76.9	1000	1000	23.4	17.58	783	73.26	39.4	78.63
		std	1.55	–	0.646	82.2	0	0	20.53	11.95	672	45.09	68.1	34.65
	runtime	mean	45.71	1311.8	0.745	0.185	1612	1891.6	16.33	57.01	1.14	0.228	8.643	342
		std	9.34	1593.2	1.1	0.062	25.05	25.7	12.89	44.51	1.1	0.114	14.71	147
Scene	iters	mean	12.56	–	1.32	319.5	1000	1000	15.03	10.94	791	71.59	43.03	75.36
		std	1.04	–	0.48	58.89	0	0	12.21	11.15	766	82.83	37.47	52.47
	runtime	mean	1.907	82.89	0.072	0.081	229.67	265.45	1.7	5.03	0.066	0.029	0.777	29.23
		std	0.266	18.46	0.111	0.011	15.46	17.6	1.3	5.88	0.071	0.024	0.681	20.36

Table 8: Iterations and practical runtime on Grids.

Models		DAOOPT	MPLP-C	ogm-ICM	ogm-BP	ogm-TRBP	ogm-DD-SG	ogm-TRWS	ogm-ADSal	ℓ_2-Box	LS-LP
M1	iters	–	420	195	1000	1000	1000	14	39	22	234
	runtime	8	303.9	0.012	40.1	40.89	86.9	0.13	2.79	0.317	41.4
M2	iters	–	1000	192	1000	1000	1000	18	37	35	227
	runtime	10	613.3	0.013	42.71	42.13	91.1	0.18	2.72	0.478	39.4
M3	iters	–	686	195	1000	1000	1000	13	36	36	155
	runtime	8	331.5	0.022	40.2	39.28	86.4	0.12	2.52	0.412	27
M4	iters	–	1000	193	1000	1000	1000	13	42	27	434
	runtime	10	523.5	0.013	42.72	45.53	91.5	0.17	3.16	0.318	75.3
M5	iters	–	1000	193	1000	1000	1000	21	37	26	265
	runtime	11	468.7	0.013	42.77	48.07	91.2	0.02	2.73	0.291	45.7
M6	iters	–	1000	741	1000	1000	1000	19	37	18	271
	runtime	7200	840.3	0.052	166.6	182.3	362.3	0.78	11.1	2.19	46.5
M7	iters	–	1000	750	1000	1000	1000	18	37	16	204
	runtime	7200	1040.5	0.053	171.6	187.2	371.7	0.7	11.4	1.96	35.3
M8	iters	–	1000	730	1000	1000	1000	17	37	29	407
	runtime	7200	917.1	0.072	166.6	182.1	362.4	0.68	10.9	3.47	66.1
M9	iters	–	1000	739	1000	1000	1000	16	36	31	327
	runtime	7200	1096.8	0.053	171.7	189.5	372.3	0.85	10.6	3.80	56.3
M10	iters	–	521	738	1000	1000	1000	45	38	45	500
	runtime	7200	273.9	0.072	166.6	182.1	362.4	2.14	10.6	5.68	88.3
M11	iters	–	1000	765	1000	1000	1000	36	44	39	390
	runtime	7200	843.1	0.053	171.7	187.7	372.2	1.57	12.6	5.13	68.4
M12	iters	–	1000	727	1000	1000	1000	26	38	19	314
	runtime	7200	824.8	0.073	166.7	187.6	361.5	1.05	11	2.25	52.5
M13	iters	–	1000	755	1000	1000	1000	34	41	17	493
	runtime	7200	999.7	0.052	171.8	188.4	372.3	1.44	11.9	2.14	85.9
M14	iters	–	1000	3029	1000	1000	1000	21	37	12	290
	runtime	–	1956.4	0.218	676.5	756	1474	3.52	43.5	21.72	50.5
M15	iters	–	1000	2991	1000	1000	1000	19	36	12	202
	runtime	–	2028.9	0.218	687.2	770.2	1495	3.37	42	21.65	33.7
M16	iters	–	1000	3038	1000	1000	1000	15	35	20	435
	runtime	–	2110.4	0.217	676.8	762.2	1471	2.58	41.5	36.53	76
M17	iters	–	1000	3048	1000	1000	1000	20	37	15	354
	runtime	–	2238.8	0.221	686.8	767.6	1492.2	3.37	43	26.93	64
M18	iters	–	1000	3173	1000	1000	1000	54	53	35	250
	runtime	–	1956.3	0.221	677	756.1	1475.2	8.87	63.7	60.28	43.5
M19	iters	–	1000	3095	1000	1000	1000	109	45	37	250
	runtime	–	1953.5	0.221	687.4	773	1493.8	19.13	50.7	63.93	39.9
M20	iters	–	1000	3038	1000	1000	1000	30	38	13	316
	runtime	–	1974.2	0.206	676.4	756	1473.2	4.99	42.5	23.06	55.4
M21	iters	–	1000	2985	1000	1000	1000	22	37	12	280
	runtime	–	2167.3	0.218	687.2	767.5	1494.4	3.75	40.6	21.54	49.6

Table 9: Iterations and practical runtime on Protein.

Models		MPLP-C	ogm-ICM	ogm-BP	ogm-TRBP	ogm-DD-SG	PSDD	AD3	LS-LP
M1	iters	499	797	1000	1000	1000	2000	2000	161
	runtime	2320	0.39	32019	31671	33381	14.14	20.28	1746
M2	iters	478	634	1000	1000	1000	2000	2000	98
	runtime	2399	0.44	30966	31253	33583	7.74	3.87	1076
M4	iters	24	859	1000	1000	1000	2000	1524	140
	runtime	80	0.51	31031	31177	33748	5.73	3.32	1535
M5	iters	500	5275	1000	1000	1000	2000	2000	1000
	runtime	2248	0.8	382766	357330	553376	22.62	28.36	10787
M6	iters	433	597	1000	1000	1000	2000	2000	482
	runtime	2427	0.36	30765	30772	32953	12.96	14.12	5314
M7	iters	475	836	1000	1000	1000	2000	2000	145
	runtime	2399	0.43	31660	31791	33620	13.55	18.63	1989.7
M8	iters	16	778	1000	1000	1000	2000	2000	186
	runtime	80	0.51	31065	31220	34550	5.13	3.31	1566

scale variables and dense connections, but with modest state spaces for variables and factors.

Practical runtime. Due to the dependency of the computational complexity on the graph structure, the practical runtime of these methods will vary significantly on different graphs. In the following, we present the practical runtime of all compared methods on above evaluated datasets. To test the runtime fairly, we run all methods at the same machine, and only run one experiment at the same time. The iterations and practical runtime on different datasets are shown respectively in Table 7 for Segmentation and Scene, Table 8 for Grids and Table 9 for Protein.

As shown in Table 7, on the small and easy models, the runtime of LP-LP, MPLP-C, ogm-ICM, PSDD and AD3 are very small, and the runtime of ogm-BP, ogm-TRBP, ogm-TRWS, ogm-ADSal, LS-LP and ℓ_2-Box are larger, while the runtime of DAOOPT are the largest. Besides, the iterations of AD3 and LS-LP are much smaller than the one of PSDD, given the fact that their similar computational complexities per iteration. The iterations of LP-LP and MPLP-C are also provided, but they are incomparable with PSDD, AD3 and LS-LP. The complexity of each iteration in LP-LP depends on the problem size $|\boldsymbol{\mu}|$ (see Eq. (7)). In MPLP-C, each outer iteration includes 100 iterations of MPLP and adding violated constraints. The complexity of each MPLP iteration is stable, but the complexity of adding violated constraints varies significantly in different outer iterations.

In terms of the comparison on Grids (see Table 8), the iterations and runtime of LP-LP are smaller than those of other methods, but it gives uniform solutions on all models. As demonstrated Section 7.2, LP-LP, PSDD and AD3 produce uniform solutions on all models, thus we also don't present their iterations and runtime. The runtime of DAOOPT on small models (*i.e.*, M1 to M5) are small, and it achieves 7200 seconds (the upper limit) on M6 to M13, while it cannot give any solution in 7200 seconds on M14 to M21. For MPLP-C, both iterations and runtime are large on all models. Note that the runtime per iteration of MPLP-C becomes larger along with the model scale. For ICM, the iteration is large on multiple models, but with very small runtime, as its complexity per iteration is low. Both message passing methods, including ogm-BP and ogm-TRBP, achieve the upper limit of iterations. It demonstrate that their convergence is very slow. The convergence of ogm-DD-SG is also slow, and its runtime per iteration is even higher than above two message passing methods. Two LP relaxation based methods, including ogm-TRWS and ogm-ADSal, converge in a few iterations, and are of small runtime. In contrast, the iterations of LS-LP are only larger than those of ogm-TRWS, ogm-ADSal and ℓ_2-Box, while smaller than other methods. And, the runtime per iteration of LS-LP is similar with that of ogm-BP, while smaller than those of other methods except of ogm-ICM and ogm-TRWS. Considering that C++ is much more efficient than MATLAB, if LS-LP is also implemented by C++, its runtime should be much lower than those of most compared methods. In other words, the computational complexity of LS-LP is much smaller than most compared methods. Note that the runtime per iteration of ℓ_2-Box increases along with the model size. For example, its runtime per iteration on M1 is 0.014 seconds, while that on M21 is 1.795 seconds. In contrast, the runtime per iteration of LS-LP are rather stable. As shown in Table 6, the complexity per iteration of ℓ_2-Box is $O\big([\sum_i^V |\mathcal{X}_i|]^3\big)$. Obviously, ℓ_2-Box is difficult to apply to large-scale models. In contrast, LS-LP is conducted based on the decomposition of the factor graph to independent factors and variables, due to which the parallel computation is allowed. Thus, the scalability of LS-LP is much better than ℓ_2-Box for MAP inference.

In terms of the comparison on Protein (see Table 9), the ascending ranking of runtime is ogm-ICM, PSDD, AD3, MPLP-C, LS-LP, ogm-BP, ogm-TRBP, ogm-DD-SG. Although

the runtime of PSDD and AD3 are very small, but they only give *approximate* solutions in 2000 iterations on all models, except for AD3 on M4. For MPLP-C, the iterations and runtime are small on M4 and M8, but very large on all other 5 models. In contrast, although the runtime of LS-LP is much larger than those of PSDD and AD3, its iterations are much smaller on all models. If LS-LP is also implemented by C++, its practical runtime will be much smaller than those of PSDD and AD3.

In summary, the above comparisons on iterations and practical runtime demonstrate:

1. LS-LP converges much faster than most compared methods, except of ogm-TRWS, ogm-ADSal and ℓ_2-Box. On large-scale models (see Table 5), the complexities per iteration of LS-LP, PSDD and AD3 are similar, and are much smaller than those of other methods (except of ICM).
2. The complexity of MPLP-C is larger than PSDD, AD3 and LS-LP, while smaller than DAOOPT. But its practical iterations and runtime vary significantly on different models. Both the complexity and practical runtime of DAOOPT are much larger than other methods.
3. Considering the performance of the MAP inference results presented in Section 7.2, 7.3 and 7.4, we conclude that LS-LP shows very competitive performance compared to state-of-the-art MAP inference methods.

7.6 Discussions

We obtain three conclusions from above experiments evaluated on different types of models. **1)** Compared with the hybrid methods including DAOOPT and MPLP-C, the performance of LS-LP is comparable. However, the computational cost of LS-LP is much lower, as the hybrid methods adopt LP relaxation based methods as sub-routines. **2)** Compared with LP relaxation based methods, especially PSDD and AD3, LS-LP always give valid solutions, without rounding; and, the logPot values of LS-LP are much higher, with similar computational cost. **3)** Compared to ℓ_2-Box, which can be only applied to models with pairwise potentials, LS-LP is applied to any type of models. Besides, the decomposition of the factor graph allows for the parallel computations with respect to each factor and each variable, while ℓ_2-Box solves a QP problem over the whole MRF model. Thus, LS-LP is a much better choice than ℓ_2-Box for MAP inference. **4)** Compared to other approximated methods, LS-LP always shows much better performance in difficult models (*e.g.*, Grids and Protein).

8 Conclusions

In this work, we proposed an novel formulation of MAP inference, called ℓ_2-sphere linear program (LS-LP). Starting from the standard linear programming (LP) relaxation, we added the ℓ_2-sphere constraint onto variable nodes. The intersection between the ℓ_2-sphere constraint and the local marginal polytope $\mathcal{L}_G$ in LP relaxation is proved to be the exact set of all valid integer label configurations. Thus, the proposed LS-LP problem is equivalent to the original MAP inference problem. By adding a sufficiently small perturbation ϵ onto the objective function and constraints, we proposed a perturbed ADMM algorithm for optimizing the LS-LP problem. Although the ℓ_2-sphere constraint is non-convex, we proved that the ADMM algorithm will globally converge to the ϵ-KKT point of the LS-LP problem. The analysis of convergence rate is also presented. Experiments on three benchmark datasets show the competitive performance of LS-LP compared to state-of-the-art MAP inference methods.

A Convergence Analysis

To facilitate the convergence analysis, here we rewrite some equations and notations firstly defined in Sections 5. Problem (11) can be simplified to the following general shape, as follows

$$\text{LS-LP}(\boldsymbol{\theta}) = \min_{\mathbf{x},\mathbf{y}} f(\mathbf{x}) + h(\mathbf{y}),\ \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{B}\mathbf{y}. \tag{32}$$

Our illustration for (18) is separated into three parts, as follows:

1. **Variables**. $\mathbf{x} = [\boldsymbol{\mu}_1; \ldots; \boldsymbol{\mu}_{|V|}] \in \mathbb{R}^{\sum_i^V |\mathcal{X}_i|}$, and it concatenates all variable nodes $\boldsymbol{\mu}_V$. $\mathbf{y} = [\mathbf{y}_1; \ldots; \mathbf{y}_{|V|}]$ with $\mathbf{y}_i = [\boldsymbol{\upsilon}_i; \boldsymbol{\mu}_{\alpha_{i,1}}; \ldots; \boldsymbol{\mu}_{\alpha_{i,|\mathcal{N}_i|}}] \in \mathbb{R}^{|\mathcal{X}_i|+\sum_\alpha^{\mathcal{N}_i}|\mathcal{X}_\alpha|}$. $\mathbf{y}$ concatenates all factor nodes $\boldsymbol{\mu}_V$ and the extra variable nodes $\boldsymbol{\upsilon}$; $\mathbf{y}_i$ concatenates the factor nodes and the extra variable node connected to the i-th variable node $\boldsymbol{\mu}_i$. $\mathcal{N}_i$ indicates the set of neighborhood factor nodes connected to the i-th variable node; the subscript $\alpha_{i,j}$ indicates the j-th factor connected to the i-th variable, with $i \in V$ and $j \in \mathcal{N}_i$.
2. **Objective functions**. $f(\mathbf{x}) = \mathbf{w}_\mathbf{x}^\top \mathbf{x}$ with $\mathbf{w}_\mathbf{x} = -[\boldsymbol{\theta}_1; \ldots; \boldsymbol{\theta}_{|V|}]$. $h(\mathbf{y}) = g(\mathbf{y}) + \mathbf{w}_\mathbf{y}^\top \mathbf{y}$, with $\mathbf{w}_\mathbf{y} = [\mathbf{w}_1; \ldots; \mathbf{w}_{|V|}]$ with $\mathbf{w}_i = -[\mathbf{0}; \frac{1}{|\mathcal{N}_{\alpha_{i,1}}|}\boldsymbol{\theta}_{\alpha_{i,1}}; \ldots; \frac{1}{|\mathcal{N}_{\alpha_{i,|\mathcal{N}_i|}}|}\boldsymbol{\theta}_{\alpha_{i,|\mathcal{N}_i|}}]$, and $\mathcal{N}_\alpha = \{i \mid (i,\alpha) \in E\}$ being the set of neighborhood variable nodes connected to the α-th factor. $g(\mathbf{y}) = \mathbb{I}(\boldsymbol{\upsilon} \in \mathcal{S}) + \sum_{\alpha \in F} \mathbb{I}(\boldsymbol{\mu}_\alpha \in \Delta^{|\mathcal{X}_\alpha|})$, with $\mathbb{I}(a)$ being the indicator function: $\mathbb{I}(a) = 0$ if a is true, otherwise $\mathbb{I}(a) = \infty$.
3. **Constraint matrices**. The constraint matrix $\mathbf{A} = \text{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_i, \ldots, \mathbf{A}_{|V|})$ with $\mathbf{A}_i = [\mathbf{I}_{|\mathcal{X}_i|}; \ldots; \mathbf{I}_{|\mathcal{X}_i|}] \in \{0,1\}^{(|\mathcal{N}_i|+1)|\mathcal{X}_i| \times |\mathcal{X}_i|}$. $\mathbf{B} = \text{diag}(\mathbf{B}_1, \ldots, \mathbf{B}_i, \ldots, \mathbf{B}_{|V|})$, with $\mathbf{B}_i = \text{diag}(\mathbf{I}_{|\mathcal{X}_i|}, \mathbf{M}_{i,\alpha_{i,1}}, \ldots, \mathbf{M}_{i,\alpha_{i,|\mathcal{N}_i|}})$. $\mathbf{A}$ summarizes all constraints on $\boldsymbol{\mu}_V$, while $\mathbf{B}$ collects all constraints on $\boldsymbol{\mu}_F$ and $\boldsymbol{\upsilon}$.

Note that Problem (18) has a clear structure with two groups of variables, corresponding the augmented factor graph (see Fig. 1(c)).

According to the analysis presented in [40], a sufficient condition to ensure the global convergence of the ADMM algorithm for the problem LS-LP($\boldsymbol{\theta}$) is that $\text{Im}(\mathbf{B}) \subseteq \text{Im}(\mathbf{A})$, with $\text{Im}(\mathbf{A})$ being the image of $\mathbf{A}$, *i.e.*, the column space of $\mathbf{A}$. However, $\mathbf{A}$ in (32) is full column rank, rather than full row rank, while $\mathbf{B}$ is full row rank. To satisfy this sufficient condition, we introduce a sufficiently small perturbation to both the objective function and the constraint in (32), as follows

$$\text{LS-LP}(\boldsymbol{\theta}; \epsilon) = \min_{\hat{\mathbf{x}},\mathbf{y}} \hat{f}(\hat{\mathbf{x}}) + h(\mathbf{y}),\ \text{s.t. } \hat{\mathbf{A}}\hat{\mathbf{x}} = \mathbf{B}\mathbf{y}, \tag{33}$$

where $\hat{\mathbf{A}} = [\mathbf{A}, \epsilon\mathbf{I}]$ with a sufficiently small constant $\epsilon > 0$, then $\hat{\mathbf{A}}$ is full row rank. $\hat{\mathbf{x}} = [\mathbf{x}; \bar{\mathbf{x}}]$, with $\bar{\mathbf{x}} = [\bar{\mathbf{x}}_1; \ldots; \bar{\mathbf{x}}_{|V|}] \in \mathbb{R}^{\sum_i^V (|\mathcal{N}_i|+1)|\mathcal{X}_i|}$ and $\bar{\mathbf{x}}_i = [\boldsymbol{\mu}_i; \ldots; \boldsymbol{\mu}_i] \in \mathbb{R}^{(|\mathcal{N}_i|+1)|\mathcal{X}_i|}$. $\hat{f}(\hat{\mathbf{x}}) = f(\mathbf{x}) + \frac{1}{2}\epsilon\hat{\mathbf{x}}^\top\hat{\mathbf{x}}$. Consequently, $\text{Im}(\hat{\mathbf{A}}) \equiv \text{Im}(\mathbf{B}) \subseteq \mathbb{R}^{\text{rank of } \hat{\mathbf{A}}}$, as both $\hat{\mathbf{A}}$ and $\mathbf{B}$ are full row rank. Then, the sufficient condition $\text{Im}(\mathbf{B}) \subseteq \text{Im}(\hat{\mathbf{A}})$ holds.

The augmented Lagrangian function of (33) is formulated as

$$\mathcal{L}_{\rho,\epsilon}(\hat{\mathbf{x}}, \mathbf{y}, \boldsymbol{\lambda}) = \hat{f}(\hat{\mathbf{x}}) + h(\mathbf{y}) + \boldsymbol{\lambda}^\top(\hat{\mathbf{A}}\hat{\mathbf{x}} - \mathbf{B}\mathbf{y}) + \frac{\rho}{2}\|\hat{\mathbf{A}}\hat{\mathbf{x}} - \mathbf{B}\mathbf{y}\|_2^2 \tag{34}$$

The updates of the ADMM algorithm to optimize (33) are as follows

$$\begin{cases} \mathbf{y}^{k+1} & = \text{argmin}_\mathbf{y}\, \mathcal{L}_{\rho,\epsilon}(\hat{\mathbf{x}}^k, \mathbf{y}, \boldsymbol{\lambda}^k), \\ \hat{\mathbf{x}}^{k+1} & = \text{argmin}_{\hat{\mathbf{x}}}\, \mathcal{L}_{\rho,\epsilon}(\hat{\mathbf{x}}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^k), \\ \boldsymbol{\lambda}^{k+1} & = \boldsymbol{\lambda}^k + \rho(\hat{\mathbf{A}}\hat{\mathbf{x}}^{k+1} - \mathbf{B}\mathbf{y}^{k+1}). \end{cases} \tag{35}$$

The optimality conditions of the variable sequence $(\mathbf{y}^{k+1}, \hat{\mathbf{x}}^{k+1}, \boldsymbol{\lambda}^{k+1})$ generated above are

$$\mathbf{B}^\top\boldsymbol{\lambda}^k + \rho\mathbf{B}^\top(\hat{\mathbf{A}}\hat{\mathbf{x}}^k - \mathbf{B}\mathbf{y}^{k+1}) = \mathbf{B}^\top\boldsymbol{\lambda}^{k+1} - \rho\mathbf{B}^\top\hat{\mathbf{A}}(\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k) \in \partial h(\mathbf{y}^{k+1}), \tag{36}$$

$$\nabla\hat{f}(\hat{\mathbf{x}}^{k+1}) + \hat{\mathbf{A}}^\top\boldsymbol{\lambda}^k + \rho\hat{\mathbf{A}}^\top(\hat{\mathbf{A}}\hat{\mathbf{x}}^{k+1} - \mathbf{B}\mathbf{y}^{k+1}) = \nabla\hat{f}(\hat{\mathbf{x}}^{k+1}) + \hat{\mathbf{A}}^\top\boldsymbol{\lambda}^{k+1} = \mathbf{0}, \tag{37}$$

$$\frac{1}{\rho}(\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k) = \hat{\mathbf{A}}\hat{\mathbf{x}}^{k+1} - \mathbf{B}\mathbf{y}^{k+1}. \tag{38}$$

The convergence of this perturbed ADMM algorithm for the LS-LP problem is summarized in Theorem 2. The detailed proof is presented in the following sub-sections sequentially. Note that hereafter $\|\cdot\|$ indicates the ℓ_2 norm for a vector, or the Frobenius norm for a matrix; $\mathcal{A}_1 \succeq \mathcal{A}_2$ represents that $\mathcal{A}_1 - \mathcal{A}_2$ is positive semi-definite, with $\mathcal{A}_1, \mathcal{A}_2$ being square matrices; ∇ denotes the gradient operator, ∇^2 means the Hessian operator, and ∂ is the sub-gradient operator; $\mathbf{I}$ represents the identity matrix with compatible shape.

A.1 Properties

In this section, we present some important properties of the objective function and constraints in (33), which will be used in the followed convergence analysis.

Properties on objective functions (P1)

- (P1.1) f, h and $\mathcal{L}_{\rho,\epsilon}$ are semi-algebraic, lower semi-continuous functions and satisfy Kurdyka-Lojasiewicz (KL) property, and h is closed and proper
- (P1.2) There exist $\mathcal{Q}_1, \mathcal{Q}_2$ such that $\mathcal{Q}_1 \succeq \nabla^2\hat{f}(\hat{\mathbf{x}}) \succeq \mathcal{Q}_2, \forall\hat{\mathbf{x}}$
- (P1.3) $\liminf_{\|\hat{\mathbf{x}}\|\to\infty} \|\nabla\hat{f}(\hat{\mathbf{x}})\| = \infty$

Properties on constraint matrices (P2)

- (P2.1) There exists $\sigma > 0$ such that $\hat{\mathbf{A}}\hat{\mathbf{A}}^\top \succeq \sigma\mathbf{I}$

- (P2.2) $\mathcal{Q}_2 + \rho\hat{\mathbf{A}}^\top\hat{\mathbf{A}} \succeq \delta\mathbf{I}$ for some $\rho, \delta > 0$, and $\rho > \frac{1}{\epsilon}$
- (P2.3) There exists $\mathcal{Q}_3 \succeq [\nabla^2\hat{f}(\hat{\mathbf{x}})]^2, \forall\hat{\mathbf{x}}$, and $\delta\mathbf{I} \succ \frac{2}{\sigma\rho}\mathcal{Q}_3$
- (P2.4) Both $\hat{\mathbf{A}}$ and $\mathbf{B}$ are full row rank, and $\text{Im}(\hat{\mathbf{A}}) \equiv \text{Im}(\mathbf{B}) \subseteq \mathbb{R}^{\text{rank of } \hat{\mathbf{A}}}$

Remark. **(1)** Although the definition of KL property (see Definition 1) is somewhat complex, but it holds for many widely used functions, according to [42]. Typical functions satisfying KL property includes: *a)* real analytic functions, and any polynomial function such as $\|\mathbf{Hx} - \mathbf{b}\|$ belongs to this type; *b)* locally strongly convex functions, such as the logistic loss function $\log(1 + \exp(-\mathbf{x}))$; *c)* semi-algebraic functions, such as $\|\mathbf{x}\|_1, \|\mathbf{x}\|_2, \|\mathbf{x}\|_\infty, \|\mathbf{Hx} - \mathbf{b}\|_1, \|\mathbf{Hx} - \mathbf{b}\|_2, \|\mathbf{Hx} - \mathbf{b}\|_\infty$ and the indicator function $\mathbb{I}(\cdot)$. It is easy to verify that P1.1 holds in our problem. **(2)** Here we provide an instantiation of above hyper-parameters satisfying above properties. Firstly, it is easy to obtain that $\nabla^2\hat{f}(\hat{\mathbf{x}}) = \epsilon\mathbf{I}$, and $\hat{\mathbf{A}}\hat{\mathbf{A}}^\top = [\mathbf{A}, \epsilon\mathbf{I}][\mathbf{A}, \epsilon\mathbf{I}]^\top = \mathbf{A}\mathbf{A}^\top + \epsilon^2\mathbf{I} \succ \epsilon^2\mathbf{I}$, as well as $\rho\hat{\mathbf{A}}^\top\hat{\mathbf{A}} \succeq \epsilon\mathbf{I}$, when ϵ is small enough and $\rho > \frac{1}{\epsilon}$ (*e.g.*, $\rho = \frac{2}{\epsilon}$). Then, the values $\mathcal{Q}_1 = \mathcal{Q}_2 = \epsilon\mathbf{I}$, $\mathcal{Q}_3 = \epsilon^2\mathbf{I}, \delta = 2\epsilon, \sigma = \epsilon^2$ satisfy P1.2, P2.1, P2.2 and P2.3. Without loss of generality, we will adopt these specific values for these hyper-parameters to simplify the following analysis, while only keeping ρ and ϵ.

A.2 Decreasing of $\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k)$

In this section, we firstly prove the decreasing property of the augmented Lagrangian function, *i.e.*,

$$\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k) > \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1}, \hat{\mathbf{x}}^{k+1}, \boldsymbol{\lambda}^{k+1}), \forall k. \tag{39}$$

Firstly, utilizing P2.1, P2.3 and (37), we obtain that

$$\epsilon^2\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|_2^2 \leq \|\hat{\mathbf{A}}(\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k)\|_2^2 = \|\nabla\hat{f}(\hat{\mathbf{x}}^{k+1}) - \nabla\hat{f}(\hat{\mathbf{x}}^k)\|_2^2 = \epsilon^2\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2^2. \tag{40}$$

Then, we have

$$\begin{aligned}&\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1}, \hat{\mathbf{x}}^{k+1}, \boldsymbol{\lambda}^{k+1}) - \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1}, \hat{\mathbf{x}}^{k+1}, \boldsymbol{\lambda}^k) = (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k)^\top(\hat{\mathbf{A}}\hat{\mathbf{x}}^{k+1} - \mathbf{B}\mathbf{y}^{k+1}) \\ =&\frac{1}{\rho}\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|_2^2 \leq \frac{1}{\rho}\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2^2\end{aligned} \tag{41}$$

According to P1.2 and P2.2, $\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1}, \hat{\mathbf{x}}, \boldsymbol{\lambda}^k)$ is strongly convex with respect to $\hat{\mathbf{x}}$, with the parameter of at least 2ϵ. Then, we have

$$\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1}, \hat{\mathbf{x}}^{k+1}, \boldsymbol{\lambda}^k) - \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1}, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k) \leq -\epsilon\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2^2. \tag{42}$$

As $\mathbf{y}^{k+1}$ is the minimal solution of $\mathcal{L}_{\rho,\epsilon}(\mathbf{y}, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k)$, it is easy to know

$$\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1}, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k) - \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k) \leq 0. \tag{43}$$

Combining (41), (42) and (43), we have

$$\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1}, \hat{\mathbf{x}}^{k+1}, \boldsymbol{\lambda}^{k+1}) - \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k) \leq (\frac{1}{\rho} - \epsilon)\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2^2 < 0, \tag{44}$$

where the last inequality utilizes P2.3 and $\rho > \frac{1}{\epsilon}$.

A.3 Boundedness of $\{\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k\}$

Next, we prove the boundedness of $\{\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k\}$. We suppose that ρ is large enough such that there is $0 < \gamma < \rho$ with

$$\inf_{\hat{\mathbf{x}}}\big(\hat{f}(\hat{\mathbf{x}}) - \frac{1}{2\epsilon^2\gamma}\|\nabla\hat{f}(\hat{\mathbf{x}})\|_2^2\big) = f^* > -\infty. \tag{45}$$

According to (44), for any $k \geq 1$, we have

$$\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k) = \hat{f}(\hat{\mathbf{x}}^k) + h(\mathbf{y}^k) + \frac{\rho}{2}\|\hat{\mathbf{A}}\hat{\mathbf{x}}^k - \mathbf{B}\mathbf{y}^k + \frac{\boldsymbol{\lambda}^k}{\rho}\|_2^2 - \frac{1}{2\rho}\|\boldsymbol{\lambda}^k\|_2^2 \leq \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^1, \hat{\mathbf{x}}^1, \boldsymbol{\lambda}^1) < \infty. \tag{46}$$

Besides, according to P2.1, we have

$$\epsilon^2\|\boldsymbol{\lambda}^k\|_2^2 \leq \|\hat{\mathbf{A}}^\top\boldsymbol{\lambda}^k\|_2^2 = \|\nabla\hat{f}(\hat{\mathbf{x}}^k)\|_2^2. \tag{47}$$

Plugging (47) into (46), we obtain that

$$\infty > \hat{f}(\hat{\mathbf{x}}^k) + h(\mathbf{y}^k) + \frac{\rho}{2}\|\hat{\mathbf{A}}\hat{\mathbf{x}}^k - \mathbf{B}\mathbf{y}^k + \frac{\boldsymbol{\lambda}^k}{\rho}\|_2^2 - \frac{1}{2\epsilon^2\rho}\|\nabla\hat{f}(\hat{\mathbf{x}}^k)\|_2^2 \geq f^* + \frac{\frac{1}{\gamma} - \frac{1}{\rho}}{2\epsilon^2}\|\nabla\hat{f}(\hat{\mathbf{x}}^k)\|_2^2 + h(\mathbf{y}^k) + \frac{\rho}{2}\|\hat{\mathbf{A}}\hat{\mathbf{x}}^k - \mathbf{B}\mathbf{y}^k + \frac{\boldsymbol{\lambda}^k}{\rho}\|_2^2. \tag{48}$$

According to the coerciveness of $\nabla\hat{f}(\hat{\mathbf{x}}^k)$ (*i.e.*, P1.3), we obtain that $\hat{\mathbf{x}}^k < \infty, \forall k$, *i.e.*, the boundedness of $\{\hat{\mathbf{x}}^k\}$. From (47), we know the boundedness of $\{\boldsymbol{\lambda}^k\}$. Besides, according to P2.4, $\{\hat{\mathbf{A}}\hat{\mathbf{x}}^k\}$ is also bounded. From (38), we obtain the boundedness of $\{\mathbf{B}\mathbf{y}^k\}$. Considering the full row rank of $\mathbf{B}$ (*i.e.*, P2.4), the boundedness of $\{\mathbf{y}^k\}$ is proved.

A.4 Convergence of Residual

According to the boundedness of $\{\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k\}$, there is a sub-sequence $\{\mathbf{y}^{k_i}, \hat{\mathbf{x}}^{k_i}, \boldsymbol{\lambda}^{k_i}\}$ that converges to a cluster point $\{\mathbf{y}^*, \hat{\mathbf{x}}^*, \boldsymbol{\lambda}^*\}$. Considering the lower semi-continuity of $\mathcal{L}_{\rho,\epsilon}$ (*i.e.*, P1.1), we have

$$\liminf_{i\to\infty} \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k_i}, \hat{\mathbf{x}}^{k_i}, \boldsymbol{\lambda}^{k_i}) \geq \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^*, \hat{\mathbf{x}}^*, \boldsymbol{\lambda}^*) > -\infty. \tag{49}$$

Summing (44) from $k = M, \ldots, N-1$ with $M \geq 1$, we have

$$\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^N, \hat{\mathbf{x}}^N, \boldsymbol{\lambda}^N) - \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^M, \hat{\mathbf{x}}^M, \boldsymbol{\lambda}^M) \leq (\frac{1}{\rho} - \epsilon) \sum_{k=M}^{N-1} \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2^2 < 0. \tag{50}$$

Then, by setting $N = k_i$ and $M = 1$, we have

$$\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k_i}, \hat{\mathbf{x}}^{k_i}, \boldsymbol{\lambda}^{k_i}) - \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^1, \hat{\mathbf{x}}^1, \boldsymbol{\lambda}^1) \leq (\frac{1}{\rho} - \epsilon) \sum_{k=1}^{k_i-1} \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2^2. \tag{51}$$

Taking limit on both sides of the above inequality, we obtain

$$-\infty < \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^*, \hat{\mathbf{x}}^*, \boldsymbol{\lambda}^*) - \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^1, \hat{\mathbf{x}}^1, \boldsymbol{\lambda}^1) \leq (\frac{1}{\rho} - \epsilon) \sum_{k=1}^{\infty} \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2^2 < 0. \tag{52}$$

It implies that

$$\lim_{k\to\infty} \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\| = 0. \tag{53}$$

Besides, according to (40), it is easy to obtain that

$$\lim_{k\to\infty} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\| = 0. \tag{54}$$

Moreover, utilizing $\mathbf{B}\mathbf{y}^{k+1} = \hat{\mathbf{A}}\hat{\mathbf{x}}^{k+1} - \frac{1}{\rho}(\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k)$ from (38), we have

$$\|\mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^k)\| \leq \|\hat{\mathbf{A}}(\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k)\| + \frac{1}{\rho}\|(\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k)\| + \frac{1}{\rho}\|(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1})\|. \tag{55}$$

Besides, as shown in Lemma 1 in [40], the full row rank of $\mathbf{B}$ (*i.e.*, P1.4) implies that

$$\|\mathbf{y}^{k+1} - \mathbf{y}^k\| \leq \bar{M}\|\mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^k)\|, \tag{56}$$

where $\bar{M} > 0$ is a constant. Taking limit on both sides of (55) and utilizing (56), we obtain

$$\lim_{k\to\infty} \|\mathbf{y}^{k+1} - \mathbf{y}^k\| = 0. \tag{57}$$

Combining (53), (54) and (57), we obtain that

$$\lim_{k\to\infty} \|\mathbf{y}^{k+1} - \mathbf{y}^k\|_2^2 + \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2^2 + \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|_2^2 = 0. \tag{58}$$

By setting $k+1 = k_i$, plugging (53) into (36) and (54) into (37), and taking limit $k_i \to \infty$, we obtain the KKT conditions. It tells that the cluster point $(\mathbf{y}^*, \hat{\mathbf{x}}^*)$ is the KKT point of LS-LP($\boldsymbol{\theta}; \boldsymbol{\epsilon}$) (*i.e.*, (33)).

A.5 Global Convergence

Inspired by the analysis presented in [25], in this section we will prove the following conclusions:

- $\sum_{k=1}^{\infty} \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\| < \infty$;
- $\{\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k\}$ converges to $(\mathbf{y}^*, \hat{\mathbf{x}}^*, \boldsymbol{\lambda}^*)$;
- $(\mathbf{y}^*, \hat{\mathbf{x}}^*)$ is the KKT point of (33).

Firstly, utilizing the optimality conditions (36, 37, 38), we have that

$$\partial_{\mathbf{y}} \mathcal{L}_{\rho,\epsilon}(\hat{\mathbf{x}}^{k+1}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^{k+1}) = \partial h(\mathbf{y}^{k+1}) - \mathbf{B}^\top \boldsymbol{\lambda}^{k+1} - \rho \mathbf{B}^\top(\hat{\mathbf{A}}\hat{\mathbf{x}}^{k+1} - \mathbf{B}\mathbf{y}^{k+1}) \ni -\mathbf{B}^\top(\boldsymbol{\lambda}^{k+1} - \lambda^k) - \rho \mathbf{B}^\top \hat{\mathbf{A}}(\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k), \tag{59}$$

$$\nabla_{\hat{\mathbf{x}}} \mathcal{L}_{\rho,\epsilon}(\hat{\mathbf{x}}^{k+1}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^{k+1}) = \nabla_{\hat{\mathbf{x}}} \hat{f}(\hat{\mathbf{x}}^{k+1}) + \hat{\mathbf{A}}^\top \boldsymbol{\lambda}^{k+1} + \rho \hat{\mathbf{A}}^\top(\hat{\mathbf{A}}\hat{\mathbf{x}}^{k+1} - \mathbf{B}\mathbf{y}^{k+1}) = \hat{\mathbf{A}}^\top(\boldsymbol{\lambda}^{k+1} - \lambda^k), \tag{60}$$

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\rho,\epsilon}(\hat{\mathbf{x}}^{k+1}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^{k+1}) = \hat{\mathbf{A}}\hat{\mathbf{x}}^{k+1} - \mathbf{B}\mathbf{y}^{k+1} = \frac{1}{\rho}(\boldsymbol{\lambda}^{k+1} - \lambda^k). \tag{61}$$

Further, combining with (40), there exists a constant $C > 0$ such that

$$\mathrm{dist}\big(0, \partial_{(\mathbf{y},\hat{\mathbf{x}},\boldsymbol{\lambda})} \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1}, \hat{\mathbf{x}}^{k+1}, \boldsymbol{\lambda}^{k+1})\big) \leq C\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|, \tag{62}$$

where dist$(\cdot,\cdot)$ denotes the distance between a vector and a set of vectors. Hereafter we denote $\partial_{(\mathbf{y},\hat{\mathbf{x}},\boldsymbol{\lambda})}\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1},\hat{\mathbf{x}}^{k+1},\boldsymbol{\lambda}^{k+1})$ as $\partial\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1},\hat{\mathbf{x}}^{k+1},\boldsymbol{\lambda}^{k+1})$ for clarity. Besides, the relation (44) implies that there is a constant $D\in(0,\epsilon-\frac{1}{\rho})$ such that

$$\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k},\hat{\mathbf{x}}^{k},\boldsymbol{\lambda}^{k})-\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1},\hat{\mathbf{x}}^{k+1},\boldsymbol{\lambda}^{k+1})\geq D\|\hat{\mathbf{x}}^{k+1}-\hat{\mathbf{x}}^{k}\|_2^2. \tag{63}$$

Moreover, the relation (49) implies that $\{\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k},\hat{\mathbf{x}}^{k},\boldsymbol{\lambda}^{k})\}$ is lower bounded along the convergent sub-sequence $\{(\mathbf{y}^{k_i},\hat{\mathbf{x}}^{k_i},\boldsymbol{\lambda}^{k_i})\}$. Combining with the its decreasing property, the limit of $\{\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k},\hat{\mathbf{x}}^{k},\boldsymbol{\lambda}^{k})\}$ exists. Thus, we will show that

$$\lim_{k\to\infty}\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k},\hat{\mathbf{x}}^{k},\boldsymbol{\lambda}^{k})=l^*:=\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{*},\hat{\mathbf{x}}^{*},\boldsymbol{\lambda}^{*}). \tag{64}$$

To prove it, we utilize the fact that $\mathbf{y}^{k+1}$ is the minimizer of $\mathcal{L}_{\rho,\epsilon}(\mathbf{y},\hat{\mathbf{x}}^{k},\boldsymbol{\lambda}^{k})$, such that

$$\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1},\hat{\mathbf{x}}^{k},\boldsymbol{\lambda}^{k})\leq\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{*},\hat{\mathbf{x}}^{k},\boldsymbol{\lambda}^{k}). \tag{65}$$

Combining the above relation, (58) and the continuity of $\mathcal{L}_{\rho,\epsilon}$ w.r.t. $\hat{\mathbf{x}}$ and $\boldsymbol{\lambda}$, the following relation holds along the sub-sequence $\{(\mathbf{y}^{k_i},\hat{\mathbf{x}}^{k_i},\boldsymbol{\lambda}^{k_i})\}$ that converges to $(\mathbf{y}^{*},\hat{\mathbf{x}}^{*},\boldsymbol{\lambda}^{*})$,

$$\limsup_{i\to\infty}\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k_i+1},\hat{\mathbf{x}}^{k_i+1},\boldsymbol{\lambda}^{k_i+1})\leq\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{*},\hat{\mathbf{x}}^{*},\boldsymbol{\lambda}^{*}). \tag{66}$$

According to (58), the sub-sequence $\{(\mathbf{y}^{k_i+1},\hat{\mathbf{x}}^{k_i+1},\boldsymbol{\lambda}^{k_i+1})\}$ also converges to $(\mathbf{y}^{*},\hat{\mathbf{x}}^{*},\boldsymbol{\lambda}^{*})$. Then, utilizing the lower semi-continuity of $\mathcal{L}_{\rho,\epsilon}$, we have

$$\liminf_{i\to\infty}\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k_i+1},\hat{\mathbf{x}}^{k_i+1},\boldsymbol{\lambda}^{k_i+1})\geq\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{*},\hat{\mathbf{x}}^{*},\boldsymbol{\lambda}^{*}). \tag{67}$$

Combining (66) with (67), we know the existence of the limit of the sequence $\{\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k},\hat{\mathbf{x}}^{k},\boldsymbol{\lambda}^{k})\}$, which proves the relation (64).

As $\mathcal{L}_{\rho,\epsilon}$ is KL function, according to Definition 1, it has the following properties:

- There exist a constant $\eta\in(0,\infty]$, a continuous concave function $\varphi:[0,\eta)\to\mathbb{R}_+$, as well as a neighbourhood $\mathcal{V}$ of $(\mathbf{y}^{*},\hat{\mathbf{x}}^{*},\boldsymbol{\lambda}^{*})$. φ is differentiable on $(0,\eta)$ with positive derivatives.
- For all $(\mathbf{y},\hat{\mathbf{x}},\boldsymbol{\lambda})\in\mathcal{V}$ satisfying $l^*<\mathcal{L}_{\rho,\epsilon}(\mathbf{y},\hat{\mathbf{x}},\boldsymbol{\lambda})<l^*+\eta$, we have

$$\varphi'(\mathcal{L}_{\rho,\epsilon}(\mathbf{y},\hat{\mathbf{x}},\boldsymbol{\lambda})-l^*)\text{dist}(0,\partial\mathcal{L}_{\rho,\epsilon}(\mathbf{y},\hat{\mathbf{x}},\boldsymbol{\lambda}))\geq 1. \tag{68}$$

Then, we define the following neighborhood sets:

$$\mathcal{V}_{\zeta}:=\left\{(\mathbf{y},\hat{\mathbf{x}},\boldsymbol{\lambda})\;\middle|\;\|\hat{\mathbf{x}}-\hat{\mathbf{x}}^{*}\|<\zeta,\|\mathbf{y}-\mathbf{y}^{*}\|<\bar{M}(\|\hat{\mathbf{A}}\|+1)\zeta,\|\boldsymbol{\lambda}-\boldsymbol{\lambda}^{*}\|<\zeta\right\}\subseteq\mathcal{V} \tag{69}$$

$$\mathcal{V}_{\zeta,\hat{\mathbf{x}}}:=\left\{\hat{\mathbf{x}}\;\middle|\;\|\hat{\mathbf{x}}-\hat{\mathbf{x}}^{*}\|<\zeta\right\}, \tag{70}$$

where $\zeta>0$ is a small constant.

Utilizing the relations (37) and (38), as well as P2.1, we obtain that for any $k\geq 1$, the following relation holds:

$$\epsilon^2\|\boldsymbol{\lambda}^{k}-\boldsymbol{\lambda}^{*}\|_2^2\leq\|\hat{\mathbf{A}}^{\top}(\boldsymbol{\lambda}^{k}-\boldsymbol{\lambda}^{*})\|_2^2=\|\triangledown\hat{f}(\hat{\mathbf{x}}^{k})-\triangledown\hat{f}(\hat{\mathbf{x}}^{*})\|_2^2=\epsilon^2\|\hat{\mathbf{x}}^{k}-\hat{\mathbf{x}}^{*}\|_2^2. \tag{71}$$

Also, the relations (37) and (38) imply that for any $k\geq 1$, we have

$$\|\mathbf{B}(\mathbf{y}^{k}-\mathbf{y}^{*})\|=\|\hat{\mathbf{A}}(\hat{\mathbf{x}}^{k}-\hat{\mathbf{x}}^{*})-\frac{1}{\rho}(\boldsymbol{\lambda}^{k}-\boldsymbol{\lambda}^{k-1})\|\leq\|\hat{\mathbf{A}}\|\|(\hat{\mathbf{x}}^{k}-\hat{\mathbf{x}}^{*})\|+\frac{1}{\rho}\|\boldsymbol{\lambda}^{k}-\boldsymbol{\lambda}^{k-1}\|. \tag{72}$$

Moreover, the relation (58) implies that $\exists N_0\geq 1$ such that $\forall k\geq N_0$, we have

$$\|\boldsymbol{\lambda}^{k}-\boldsymbol{\lambda}^{k-1}\|\leq\rho\zeta. \tag{73}$$

Similar to (56), the full row rank of $\mathbf{B}$ implies $\|\mathbf{y}^{k}-\mathbf{y}^{*}\|\leq\bar{M}\|\mathbf{B}(\mathbf{y}^{k}-\mathbf{y}^{*})\|$. Then, plugging (73) into (72), we obtain that

$$\|\mathbf{y}^{k}-\mathbf{y}^{*}\|\leq\bar{M}(\|\hat{\mathbf{A}}\|+1)\zeta, \tag{74}$$

for any $\hat{\mathbf{x}}^{k}\in\mathcal{V}_{\zeta,\hat{\mathbf{x}}}$ and $k\geq N_0$. Combining (71) and (74), we know that if $\hat{\mathbf{x}}^{k}\in\mathcal{V}_{\zeta,\hat{\mathbf{x}}}$ and $k\geq N_0$, then $(\mathbf{y}^{k},\hat{\mathbf{x}}^{k},\boldsymbol{\lambda}^{k})\in\mathcal{V}_{\zeta}\subseteq\mathcal{V}$.

Moreover, (44) and (64) implies that $\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k},\hat{\mathbf{x}}^{k},\boldsymbol{\lambda}^{k})\geq l^*,\forall k\geq 1$. Besides, as $(\mathbf{y}^{*},\hat{\mathbf{x}}^{*},\boldsymbol{\lambda}^{*})$ is a cluster point, we will obtain that $\exists N\geq N_0$, the following relations hold:

$$\begin{cases}\hat{\mathbf{x}}^{N}\in\mathcal{V}_{\zeta,\hat{\mathbf{x}}}\\ l^*<\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{N},\hat{\mathbf{x}}^{N},\boldsymbol{\lambda}^{N})<l^*+\eta\\ \|\hat{\mathbf{x}}^{N}-\hat{\mathbf{x}}^{*}\|+2\sqrt{(\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{N},\hat{\mathbf{x}}^{N},\boldsymbol{\lambda}^{N})-l^*)/D}+\frac{C}{D}(\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{N},\hat{\mathbf{x}}^{N},\boldsymbol{\lambda}^{N})-l^*)<\zeta\end{cases} \tag{75}$$

Next, We will show that if $\hat{\mathbf{x}}^{N}\in\mathcal{V}_{\zeta,\hat{\mathbf{x}}}$ and $l^*<\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{N},\hat{\mathbf{x}}^{N},\boldsymbol{\lambda}^{N})<l^*+\eta$ hold for some fixed $k\geq N_0$, then the following relation holds

$$\|\hat{\mathbf{x}}^{k+1}-\hat{\mathbf{x}}^{k}\|+\big(\|\hat{\mathbf{x}}^{k+1}-\hat{\mathbf{x}}^{k}\|-\|\hat{\mathbf{x}}^{k}-\hat{\mathbf{x}}^{k-1}\|\big)\leq\frac{C}{D}\Big[\varphi\big(\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k},\hat{\mathbf{x}}^{k},\boldsymbol{\lambda}^{k})-l^*\big)-\varphi\big(\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1},\hat{\mathbf{x}}^{k+1},\boldsymbol{\lambda}^{k+1})-l^*\big)\Big]. \tag{76}$$

To prove (76), we utilize the fact that $\hat{\mathbf{x}}^k \in \mathcal{V}_{\zeta,\hat{\mathbf{x}}}, k \geq N_0$ implies that $(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k) \in \mathcal{V}_{\zeta,\hat{\mathbf{x}}} \subseteq \mathcal{V}$. And, combining with $l^* < \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k) < l^* + \eta$, we obtain that

$$\varphi'(\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k) - l^*)\text{dist}(0, \partial\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k)) \geq 1. \tag{77}$$

Combining the relations (62), (63) and (77), as well as the concavity of φ, we obtain that

$$\begin{aligned}
&C\|\hat{\mathbf{x}}^k - \hat{\mathbf{x}}^{k-1}\| \cdot \left[\varphi\big(\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k) - l^*\big) - \varphi\big(\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1}, \hat{\mathbf{x}}^{k+1}, \boldsymbol{\lambda}^{k+1}) - l^*\big)\right] \\
\geq &\text{dist}(0, \partial\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k)) \cdot \left[\varphi\big(\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k) - l^*\big) - \varphi\big(\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1}, \hat{\mathbf{x}}^{k+1}, \boldsymbol{\lambda}^{k+1}) - l^*\big)\right] \\
\geq &\text{dist}(0, \partial\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k)) \cdot \varphi'\big(\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k) - l^*\big) \cdot \left[\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k) - \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{k+1}, \hat{\mathbf{x}}^{k+1}, \boldsymbol{\lambda}^{k+1})\right] \\
\geq &D\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|^2,
\end{aligned} \tag{78}$$

for all such k. Taking square root on both sides of (78), and utilizing the fact that $a + b \geq 2\sqrt{ab}$, then (76) is proved.

We then prove $\forall k \geq N, \hat{\mathbf{x}}^k \in \mathcal{V}_{\zeta,\hat{\mathbf{x}}}$ holds. This claim can be proved through induction. Obviously it is true for $k = N$ by construction, as shown in (75). For $k = N + 1$, we have

$$\begin{aligned}
&\|\hat{\mathbf{x}}^{N+1} - \hat{\mathbf{x}}^*\| \leq \|\hat{\mathbf{x}}^{N+1} - \hat{\mathbf{x}}^N\| + \|\hat{\mathbf{x}}^N - \hat{\mathbf{x}}^*\| \\
\leq &\sqrt{\big(\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^N, \hat{\mathbf{x}}^N, \boldsymbol{\lambda}^N) - \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{N+1}, \hat{\mathbf{x}}^{N+1}, \boldsymbol{\lambda}^{N+1})\big)/D} + \|\hat{\mathbf{x}}^N - \hat{\mathbf{x}}^*\| \\
\leq &\sqrt{\big(\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^N, \hat{\mathbf{x}}^N, \boldsymbol{\lambda}^N) - l^*\big)/D} + \|\hat{\mathbf{x}}^N - \hat{\mathbf{x}}^*\| < \zeta,
\end{aligned} \tag{79}$$

where the first inequality utilizes (63), and the last inequality follows the last relation in (75). Thus, $\forall k \geq N, \hat{\mathbf{x}}^k \in \mathcal{V}_{\zeta,\hat{\mathbf{x}}}$ holds.

Next, we suppose that $\hat{\mathbf{x}}^N, \ldots, \hat{\mathbf{x}}^{N+t-1} \in \mathcal{V}_{\zeta,\hat{\mathbf{x}}}$ for some $t > 1$, and we need to prove that $\hat{\mathbf{x}}^{N+t} \in \mathcal{V}_{\zeta,\hat{\mathbf{x}}}$ also holds, *i.e.*,

$$\begin{aligned}
&\|\hat{\mathbf{x}}^{N+t} - \hat{\mathbf{x}}^*\| \leq \|\hat{\mathbf{x}}^N - \hat{\mathbf{x}}^*\| + \|\hat{\mathbf{x}}^{N+1} - \hat{\mathbf{x}}^N\| + \sum_{i=1}^{t-1}\|\hat{\mathbf{x}}^{N+i+1} - \hat{\mathbf{x}}^{N+i}\| \\
= &\|\hat{\mathbf{x}}^N - \hat{\mathbf{x}}^*\| + 2\|\hat{\mathbf{x}}^{N+1} - \hat{\mathbf{x}}^N\| - \|\hat{\mathbf{x}}^{N+t} - \hat{\mathbf{x}}^{N+t-1}\| + \sum_{i=1}^{t-1}\left[\|\hat{\mathbf{x}}^{N+i+1} - \hat{\mathbf{x}}^{N+i}\| + \big(\|\hat{\mathbf{x}}^{N+i+1} - \hat{\mathbf{x}}^{N+i}\| - \|\hat{\mathbf{x}}^{N+i} - \hat{\mathbf{x}}^{N+i-1}\|\big)\right] \\
\leq &\|\hat{\mathbf{x}}^N - \hat{\mathbf{x}}^*\| + 2\|\hat{\mathbf{x}}^{N+1} - \hat{\mathbf{x}}^N\| + \frac{C}{D}\sum_{i=1}^{t-1}\left[\varphi^{N+i} - \varphi^{N+i+1}\right] \\
\leq &\|\hat{\mathbf{x}}^N - \hat{\mathbf{x}}^*\| + 2\|\hat{\mathbf{x}}^{N+1} - \hat{\mathbf{x}}^N\| + \frac{C}{D}\sum_{i=1}^{t-1}\varphi^{N+1} \\
\leq &\|\hat{\mathbf{x}}^N - \hat{\mathbf{x}}^*\| + 2\sqrt{\frac{\mathcal{L}_{\rho,\epsilon}^N - \mathcal{L}_{\rho,\epsilon}^{N+1}}{D}} + \frac{C}{D}\sum_{i=1}^{t-1}\varphi^{N+1} \\
\leq &\|\hat{\mathbf{x}}^N - \hat{\mathbf{x}}^*\| + 2\sqrt{\frac{\mathcal{L}_{\rho,\epsilon}^N - l^*}{D}} + \frac{C}{D}\sum_{i=1}^{t-1}\varphi^{N+1} < \zeta
\end{aligned} \tag{80}$$

where $\varphi^{N+i} = \varphi(\mathcal{L}_{\rho,\epsilon}(\mathbf{y}^{N+i}, \hat{\mathbf{x}}^{N+i}, \boldsymbol{\lambda}^{N+i}) - l^*)$ and $\mathcal{L}_{\rho,\epsilon}^N = \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^N, \hat{\mathbf{x}}^N, \boldsymbol{\lambda}^N)$. The second inequality follows from (76). The fourth inequality follows from (63). The fifth inequality utilizes the fact that $\mathcal{L}_{\rho,\epsilon}^{N+1} > l^*$, and the last inequality follows from the last relation in (75). Thus, $\hat{\mathbf{x}}^{N+k} \in \mathcal{V}_{\zeta,\hat{\mathbf{x}}}$ holds. We have proved that $\forall k \geq N, \hat{\mathbf{x}}^k \in \mathcal{V}_{\zeta,\hat{\mathbf{x}}}$ holds by induction.

Then, according to $\forall k \geq N, \hat{\mathbf{x}}^k \in \mathcal{V}_{\zeta,\hat{\mathbf{x}}}$, we can sum both sides of (76) from $k = N$ to ∞, to obtain that

$$\sum_{k=N}^{\infty}\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\| \leq \frac{C}{D}\varphi^N + \|\hat{\mathbf{x}}^N - \hat{\mathbf{x}}^{N-1}\| < \infty, \tag{81}$$

which implies that $\sum_{k=1}^{\infty}\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\| < \infty$ holds. Thus $\{\hat{\mathbf{x}}^k\}$ converges. The convergence of $\{\mathbf{y}^k\}$ follows from $\mathbf{B}\mathbf{y}^{k+1} = \hat{\mathbf{A}}\hat{\mathbf{x}}^{k+1} + \frac{1}{\rho}(\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k)$ in (38) and (58), as well as the surjectivity of $\mathbf{B}$ (*i.e.*, full row rank). The convergence of $\{\boldsymbol{\lambda}^k\}$ follows from $\nabla\hat{f}(\hat{\mathbf{x}}^{k+1}) = -\hat{\mathbf{A}}^\top\boldsymbol{\lambda}^{k+1}$ in (37) and the surjectivity of $\hat{\mathbf{A}}$ (*i.e.*, full row rank). Consequently, $\{\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k\}$ converges to the cluster point $(\mathbf{y}^*, \hat{\mathbf{x}}^*, \boldsymbol{\lambda}^*)$. The conclusion that $(\mathbf{y}^*, \hat{\mathbf{x}}^*)$ is the KKT point of Problem (33) has been proved in Section A.4.

A.6 ϵ-KKT Point of the Original LS-LP Problem

Proposition 1 *The globally converged solution $(\mathbf{y}^*, \mathbf{x}^*, \boldsymbol{\lambda}^*)$ produced by the ADMM algorithm for the perturbed LS-LP problem (33) is the ϵ-KKT solution to the original LS-LP problem (32).*

Proof The globally converged solution $(\mathbf{y}^*, \hat{\mathbf{x}}^*, \boldsymbol{\lambda}^*)$ to the perturbed LS-LP problem (33) satisfies the following relations:

$$\mathbf{B}^\top\boldsymbol{\lambda}^* \in \partial h(\mathbf{y}^*),\ \nabla\hat{f}(\hat{\mathbf{x}}^*) = -\hat{\mathbf{A}}^\top\boldsymbol{\lambda}^*,\ \hat{\mathbf{A}}\hat{\mathbf{x}}^* = \mathbf{B}\mathbf{y}^*. \tag{82}$$

Recalling the definitions $\hat{\mathbf{A}} = [\mathbf{A}, \epsilon\mathbf{I}]$, $\hat{\mathbf{x}} = [\mathbf{x}; \bar{\mathbf{x}}]$ and $\hat{f}(\hat{\mathbf{x}}) = f(\mathbf{x}) + \frac{\epsilon}{2}\hat{\mathbf{x}}^\top\hat{\mathbf{x}}$, the above relations imply that

$$\nabla\hat{f}(\hat{\mathbf{x}}^*) + \hat{\mathbf{A}}^\top\boldsymbol{\lambda}^* = \nabla f(\mathbf{x}^*) + \mathbf{A}^\top\boldsymbol{\lambda}^* + \epsilon\mathbf{x}^* = \mathbf{0} \;\Rightarrow\; \|\nabla f(\mathbf{x}^*) + \mathbf{A}^\top\boldsymbol{\lambda}^*\| = \epsilon\|\mathbf{x}^*\| = O(\epsilon), \tag{83}$$

$$\hat{\mathbf{A}}\hat{\mathbf{x}}^* + \mathbf{B}\mathbf{y}^* = \mathbf{A}\mathbf{x}^* + \epsilon\bar{\mathbf{x}}^* + \mathbf{B}\mathbf{y}^* = 0 \;\Rightarrow\; \|\mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{y}^*\| = \|\epsilon\bar{\mathbf{x}}^*\| = O(\epsilon), \tag{84}$$

where we utilize the boundedness of $\{\hat{\mathbf{x}}^*\}$. Thus, according to Definition 2, the globally converged point $(\mathbf{y}^*, \mathbf{x}^*)$ is the ϵ-KKT solution to the original LS-LP problem (32).

A.7 Convergence Rate

Lemma 3 *Firstly, without loss of generality, we can assume that $l^* = \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^*, \hat{\mathbf{x}}^*, \boldsymbol{\lambda}^*) = 0$ (e.g., one can replace $l_k = \mathcal{L}_{\rho,\epsilon}(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k)$ by $l_k - l^*$). We further assume that $\mathcal{L}_{\rho,\epsilon}$ has the KL property at $(\mathbf{y}^*, \hat{\mathbf{x}}^*, \boldsymbol{\lambda}^*)$ with the concave function $\varphi(s) = cs^{1-p}$, where $p \in [0, 1), c > 0$. Consequently, we can obtain the following inequalities:*

(i) if $p = 0$, then $\{(\mathbf{y}^k, \hat{\mathbf{x}}^k, \boldsymbol{\lambda}^k)\}_{k=1,\ldots,\infty}$ can converge in finite steps;
(ii) If $p \in (0, \frac{1}{2}]$, then there exist $c > 0$ and $\tau \in (0, 1)$ such that $\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\| \le c\tau^k$;
(iii) $p \in (\frac{1}{2}, 1)$, then there exist $c > 0$ such that $\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\| \le ck^{-\frac{1-p}{2p-1}}$.

Proof (*i*) If $p = 0$, we define a subset $H = \{k \in \mathbb{N} : \hat{\mathbf{x}}_k \neq \hat{\mathbf{x}}_{k+1}\}$. If $k \in H$ is sufficiently large, then these exists $C_3 > 0$ such that

$$\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|^2 \ge C_3 > 0. \tag{85}$$

Combining with (63), we have

$$l_k - l_{k+1} \ge D\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|^2 \ge C_3 D > 0. \tag{86}$$

If the subset H is infinite, then it will contradict to the fact that $l_k - l_{k+1} \to 0$ as $k \to \infty$. Thus, H is a finite subset, leading to the conclusion that $\{\hat{\mathbf{x}}^k\}_{k\in\mathbb{N}}$ will converge in finite steps. Recalling the relationships between $\hat{\mathbf{x}}_k$ and $\mathbf{y}_k, \boldsymbol{\lambda}_k$ (see the descriptions under (81)), we also obtain that $\{\mathbf{y}_k, \boldsymbol{\lambda}_k\}_{k\in\mathbb{N}}$ converges in finite steps.

By defining $\triangle_k = \sum_k^\infty \|x^k + 1 - x^k\|$, the inequality (81) can be rewritten as follows

$$\triangle_k \le \frac{C}{D}\varphi(l_k) + (\triangle_{k-1} - \triangle_k) < \infty. \tag{87}$$

Besides, the KL property and $l^* = 0$ give that

$$\varphi'(l_k)\text{dist}(0, \partial(l_k)) = c(1-p)l_k^{1-p}\text{dist}(0, \partial(l_k)) \ge 1 \;\Rightarrow\; l_k^p \le c(1-p)\text{dist}(0, \partial(l_k)). \tag{88}$$

Combining with (62), we obtain

$$l_k^p \le c(1-p)C(\triangle_{k-1} - \triangle_k) \;\Rightarrow\; \varphi(l_k) = cl_k^{1-p} \le c(c(1-p)C)^{\frac{1-p}{p}}(\triangle_{k-1} - \triangle_k)^{\frac{1-p}{p}} = C_1(\triangle_{k-1} - \triangle_k)^{\frac{1-p}{p}}. \tag{89}$$

Then, inserting (89) into (87), we obtain

$$\triangle_k \le C_2(\triangle_{k-1} - \triangle_k)^{\frac{1-p}{p}} + (\triangle_{k-1} - \triangle_k) < \infty. \tag{90}$$

(*ii*) If $p \in (0, \frac{1}{2}]$, then $\frac{1-p}{p} \ge 1$. Besides, since $(\triangle_{k-1} - \triangle_k) \to 0$ when $k \to \infty$, there exists an integer K_0 such that $(\triangle_{k-1} - \triangle_k) < 1$, leading to that $(\triangle_{k-1} - \triangle_k)^{\frac{1-p}{p}} \le (\triangle_{k-1} - \triangle_k)$. Inserting it into (90), we obtain that

$$\triangle_k \le (C_2 + 1)(\triangle_{k-1} - \triangle_k) \;\Rightarrow\; \triangle_k \le C_3(\triangle_{k-1} - \triangle_k) \;\Rightarrow\; \triangle_k \le \frac{C_3}{1 + C_3}\triangle_{k-1} = \tau\triangle_{k-1}, \text{ with } \tau \in (0, 1), \forall k > K_0. \tag{91}$$

It is easy to deduce that $\triangle_k \le (\triangle_{K_0}\tau^{-K_0})\tau^k = \frac{c}{2}\tau^k$, with c being a positive constant. Note that k in τ^k indicates k power of τ, rather than the iteration index. Combining with $\|\hat{\mathbf{x}}^k - \hat{\mathbf{x}}^*\| \le \triangle_k$, it is easy to obtain that $\|\hat{\mathbf{x}}^k - \hat{\mathbf{x}}^*\| \le \frac{c}{2}\tau^k$ with $\tau \in (0, 1)$ and c being a positive constant. Then, we have

$$\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\| \le \|\hat{\mathbf{x}}^k - \hat{\mathbf{x}}^*\| + \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^*\| \le \frac{c}{2}(\tau^{k+1} + \tau^k) \le c\tau^k. \tag{92}$$

(*iii*) If $p \in (\frac{1}{2}, 1)$, then $\frac{1-p}{p} < 1$. Then, it is easy to obtain that $(\triangle_{k-1} - \triangle_k)^{\frac{1-p}{p}} > (\triangle_{k-1} - \triangle_k)$. Inserting it into (90), we obtain that

$$\triangle_k \le (C_2 + 1)(\triangle_{k-1} - \triangle_k)^{\frac{1-p}{p}} \;\Rightarrow\; \triangle_k \le C_3(\triangle_{k-1} - \triangle_k)^{\frac{1-p}{p}} \;\Rightarrow\; \triangle_k^{\frac{p}{1-p}} \le C_4(\triangle_{k-1} - \triangle_k),\ \forall k > K_0. \tag{93}$$

It has been studied in Theorem 2 of [1] that the above inequality can deduce $\triangle_k \le \frac{c}{2}k^{-\frac{1-p}{2p-1}}$, with c being a positive constant. Since $\|\hat{\mathbf{x}}^k - \hat{\mathbf{x}}^*\| \le \triangle_k$, we have that $\|\hat{\mathbf{x}}^k - \hat{\mathbf{x}}^*\| \le \frac{c}{2}k^{-\frac{1-p}{2p-1}}$. Then, we have

$$\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\| \le \|\hat{\mathbf{x}}^k - \hat{\mathbf{x}}^*\| + \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^*\| \le \frac{c}{2}\big(k^{-\frac{1-p}{2p-1}} + (k+1)^{-\frac{1-p}{2p-1}}\big) \le ck^{-\frac{1-p}{2p-1}}. \tag{94}$$

Proposition 2 *We adopt the same assumptions in Lemma 3. Then,*

(i) If $p = 0$, then we will obtain the ϵ-KKT solution to the LS-LP problem in finite steps.
(ii) If $p \in (0, \frac{1}{2}]$, then we will obtain the ϵ-KKT solution to the LS-LP problem in at least $O\big(\log_{\frac{1}{\tau}}(\frac{1}{\epsilon})^2\big)$ steps.
(iii) If $p \in (\frac{1}{2}, 1)$, then we will obtain the ϵ-KKT solution to the LS-LP problem in at least $O\big((\frac{1}{\epsilon})^{\frac{4p-2}{1-p}}\big)$ steps.

Proof The conclusion (i) directly holds from Lemma 3(i).

According to the optimality condition (36), we have

$$\begin{aligned}
&\mathrm{dist}\big(\mathbf{B}^\top \boldsymbol{\lambda}^{k+1}, \partial h(\mathbf{y}^{k+1})\big) = \|\rho \mathbf{B}^\top \hat{\mathbf{A}}(\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k)\|_2 \\
\Rightarrow\ &\mathrm{dist}^2\big(\mathbf{B}^\top \boldsymbol{\lambda}^{k+1}, \partial h(\mathbf{y}^{k+1})\big) = \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|^2_{\rho^2 \hat{\mathbf{A}}^\top \mathbf{B}\mathbf{B}^\top \hat{\mathbf{A}}} \leq \xi_{\max}(\rho^2 \hat{\mathbf{A}}^\top \mathbf{B}\mathbf{B}^\top \hat{\mathbf{A}}) \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2^2 = O(\frac{1}{\epsilon^2}) \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2^2 \\
\Rightarrow\ &\mathrm{dist}\big(\mathbf{B}^\top \boldsymbol{\lambda}^{k+1}, \partial h(\mathbf{y}^{k+1})\big) \leq O(\frac{1}{\epsilon}) \cdot \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2
\end{aligned} \tag{95}$$

According to the optimality condition (38) and the relation (40), we obtain that

$$\|\hat{\mathbf{A}}\hat{\mathbf{x}}^{k+1} - \mathbf{B}\mathbf{y}^{k+1}\|_2 = \frac{1}{\rho}\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|_2 \leq \frac{1}{\rho}\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2 \leq \epsilon \cdot \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2 \leq O(\frac{1}{\epsilon}) \cdot \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2. \tag{96}$$

According to Lemma 3, we have

(ii) If $p \in (0, \frac{1}{2}]$, then

$$O(\frac{1}{\epsilon}) \cdot \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2 \leq O(\frac{1}{\epsilon})\tau^k \leq O(\epsilon) \ \Rightarrow\ k \geq O\big(\log_{\frac{1}{\tau}}(\frac{1}{\epsilon})^2\big),$$

which means that when $k \geq O\big(\log_{\frac{1}{\tau}}(\frac{1}{\epsilon})^2\big)$, we will obtain the ϵ-KKT solution to the perturbed LS-LP problem, *i.e.*, the ϵ-KKT solution to the original LS-LP problem.

(iii) If $p \in (\frac{1}{2}, 1)$, then

$$O(\frac{1}{\epsilon}) \cdot \|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\|_2 \leq O(\frac{1}{\epsilon})k^{-\frac{1-p}{2p-1}} \leq O(\epsilon) \ \Rightarrow\ k \geq O\big((\frac{1}{\epsilon})^{\frac{4p-2}{1-p}}\big),$$

which means that when $k \geq O\big((\frac{1}{\epsilon})^{\frac{4p-2}{1-p}}\big)$, we will obtain the ϵ-KKT solution to the perturbed LS-LP problem, *i.e.*, the ϵ-KKT solution to the original LS-LP problem.

References

1. Attouch, H., Bolte, J.: On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. Mathematical Programming **116**(1-2), 5–16 (2009)
2. Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-lojasiewicz inequality. Mathematics of Operations Research **35**(2), 438–457 (2010)
3. Besag, J.: On the statistical analysis of dirty pictures. Journal of the Royal Statistical Society. Series B (Methodological) pp. 259–302 (1986)
4. Bolte, J., Daniilidis, A., Lewis, A.: The łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. SIAM Journal on Optimization **17**(4), 1205–1223 (2007)
5. Bolte, J., Daniilidis, A., Lewis, A., Shiota, M.: Clarke subgradients of stratifiable functions. SIAM Journal on Optimization **18**(2), 556–572 (2007)
6. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning **3**(1), 1–122 (2011)
7. Elidan, G., Globerson, A., Heinemann, U.: Pascal 2011 probabilistic inference challenge. http://www.cs.huji.ac.il/project/PASCAL/index.php (2012)
8. Fu, Q., Banerjee, H.W.A.: Bethe-admm for tree decomposition based parallel map inference. In: Uncertainty in Artificial Intelligence, p. 222. Citeseer (2013)
9. Globerson, A., Jaakkola, T.S.: Fixing max-product: Convergent message passing algorithms for map lp-relaxations. In: NIPS, pp. 553–560 (2008)
10. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV, pp. 1–8. IEEE (2009)
11. Jaimovich, A., Elidan, G., Margalit, H., Friedman, N.: Towards an integrated protein–protein interaction network: A relational markov network approach. Journal of Computational Biology **13**(2), 145–164 (2006)
12. Johnson, J.K., Malioutov, D.M., Willsky, A.S.: Lagrangian relaxation for map estimation in graphical models. arXiv preprint arXiv:0710.0013 (2007)
13. Jojic, V., Gould, S., Koller, D.: Accelerated dual decomposition for map inference. In: ICML, pp. 503–510 (2010)
14. Kappes, J.H., Andres, B., Hamprecht, F.A., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B.X., Kröger, T., Lellmann, J., et al.: A comparative study of modern inference techniques for structured discrete energy minimization problems. International Journal of Computer Vision **115**, 155–184 (2015)
15. Kappes, J.H., Savchynskyy, B., Schnörr, C.: A bundle approach to efficient map-inference by lagrangian relaxation. In: CVPR, pp. 1688–1695. IEEE (2012)
16. Karush, W.: Minima of functions of several variables with inequalities as side constraints. M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago (1939)
17. Kelley, J.: The cutting-plane method for solving convex programs. Journal of the Society for Industrial and Applied Mathematics pp. 703–712 (1960)
18. Koller, D., Nir, F. (eds.): Probabilistic Graphical Models: Principles and Techniques. MIT Press, Cambridge, MA (2009)
19. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. IEEE transactions on pattern analysis and machine intelligence **28**(10), 1568–1583 (2006)
20. Komodakis, N., Paragios, N., Tziritas, G.: Mrf optimization via dual decomposition: Message-passing revisited. In: ICCV, pp. 1–8. IEEE (2007)
21. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Transactions on information theory **47**(2), 498–519 (2001)
22. Kuhn, H.W., Tucker, A.W.: Nonlinear programming. In: Traces and emergence of nonlinear programming, pp. 247–258. Springer (2014)
23. Land, A.H., Doig, A.G.: An automatic method of solving discrete programming problems. Econometrica: Journal of the Econometric Society pp. 497–520 (1960)
24. Laurent, M., Rendl, F.: Semidefinite programming and integer programming. Centrum voor Wiskunde en Informatica (2002)
25. Li, G., Pong, T.K.: Global convergence of splitting methods for nonconvex composite optimization. SIAM Journal on Optimization **25**(4), 2434–2460 (2015)
26. Lojasiewicz, S.: Une propriété topologique des sous-ensembles analytiques réels. Les équations aux dérivées partielles **117**, 87–89 (1963)
27. Martins, A.F., Figeuiredo, M.A., Aguiar, P.M., Smith, N.A., Xing, E.P.: An augmented lagrangian approach to constrained map inference. In: ICML (2011)
28. Martins, A.F., Figueiredo, M.A., Aguiar, P.M., Smith, N.A., Xing, E.P.: Ad3: Alternating directions dual decomposition for map inference in graphical models. Journal of Machine Learning Research **16**, 495–545 (2015)
29. Meshi, O., Globerson, A.: An alternating direction method for dual map lp relaxation. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 470–483. Springer (2011)
30. Meshi, O., Mahdavi, M., Schwing, A.: Smooth and strong: Map inference with linear convergence. In: NIPS, pp. 298–306 (2015)
31. Otten, L., Dechter, R.: Anytime and/or depth-first search for combinatorial optimization. AI Communications **25**(3), 211–227 (2012)
32. Otten, L., Ihler, A., Kask, K., Dechter, R.: Winning the pascal 2011 map challenge with enhanced and/or branch-and-bound. In: IN NIPS WORKSHOP DISCML. Citeseer (2012)
33. Savchynskyy, B., Schmidt, S., Kappes, J., Schnörr, C.: Efficient mrf energy minimization via adaptive diminishing smoothing. arXiv preprint arXiv:1210.4906 (2012)
34. Schwing, A.G., Hazan, T., Pollefeys, M., Urtasun, R.: Globally convergent dual map lp relaxation solvers using fenchel-young margins. In: NIPS, pp. 2384–2392 (2012)
35. Schwing, A.G., Hazan, T., Pollefeys, M., Urtasun, R.: Globally convergent parallel map lp relaxation solver using the frank-wolfe algorithm. In: ICML, pp. 487–495 (2014)
36. Sontag, D.A.: Approximate inference in graphical models using lp relaxations. Ph.D. thesis, Massachusetts Institute of Technology (2010)
37. Sontag, D.A., Li, Y., et al.: Efficiently searching for frustrated cycles in map inference. In: UAI (2012)
38. Wainwright, M.J., Jaakkola, T.S., Willsky, A.S.: Map estimation via agreement on trees: message-passing and linear programming. IEEE transactions on information theory **51**(11), 3697–3717 (2005)
39. Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. Foundations and Trends in Machine Learning **1**(1-2), 1–305 (2008)
40. Wang, Y., Yin, W., Zeng, J.: Global convergence of admm in nonconvex nonsmooth optimization. Journal of scientific programming (2017)
41. Wu, B., Ghanem, B.: ℓ_p-box admm: A versatile framework for integer programming. IEEE transactions on pattern analysis and machine intelligence **41**(7), 1695–1708 (2019)
42. Xu, Y., Yin, W.: A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM Journal on imaging sciences **6**(3), 1758–1789 (2013)