



# The distance between convex sets with Minkowski sum structure: application to collision detection

Xiangfeng Wang<sup>1</sup> · Junping Zhang<sup>2</sup> · Wenxing Zhang<sup>2</sup>

Received: 21 July 2019 / Published online: 13 July 2020  
© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

The distance between sets is a long-standing computational geometry problem. In robotics, the distance between convex sets with Minkowski sum structure plays a fundamental role in collision detection. However, it is typically nontrivial to be computed, even if the projection onto each component set admits explicit formula. In this paper, we explore the problem of calculating the distance between convex sets arising from robotics. Upon the recent progress in convex optimization community, the proposed model can be efficiently solved by the recent hot-investigated first-order methods, e.g., alternating direction method of multipliers or primal-dual hybrid gradient method. Preliminary numerical results demonstrate that those first-order methods are fairly efficient in solving distance problems in robotics.

**Keywords** Distance · Minkowski sum of sets · Projection · Alternating direction method of multipliers · Primal-dual hybrid gradient method · Collision detection

## 1 Introduction

The Minkowski sum of sets (also known as dilation, sumset) is a long-standing and fundamental nomenclature in computational geometry. It plays an important role in the fields, such as mathematical morphology in image processing [44], collision detection in robotics [29], configuration space computation in mechanics [48],

---

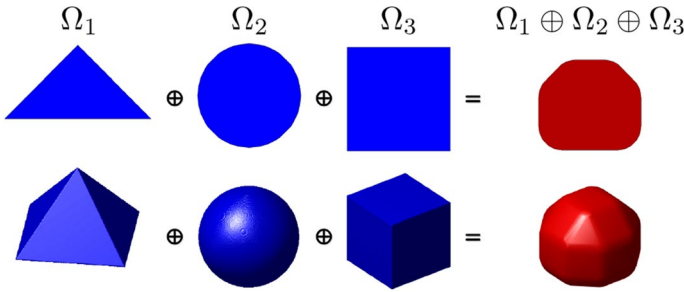
✉ Wenxing Zhang  
zhangwx@uestc.edu.cn

Xiangfeng Wang  
xfwang@sei.ecnu.edu.cn

Junping Zhang  
zjunping1208@126.com

<sup>1</sup> School of Computer Science and Technology, East China Normal University, Shanghai 200062, China

<sup>2</sup> School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China



**Fig. 1** Diagram representations of the Minkowski sum of  $\Omega_1 = \{x \in \mathbb{R}^n \mid \|x\|_1 \leq 1, x_n \geq 0\}$ ,  $\Omega_2 = \{x \in \mathbb{R}^n \mid \|x\|_2 \leq 1\}$  and  $\Omega_3 = \{x \in \mathbb{R}^n \mid \|x\|_\infty \leq 1\}$ . Top row: the case of  $n = 2$ . Bottom row: the case of  $n = 3$

numerical control machining in computer-aided manufacturing [45], and aggregation theory in economics [33]. Given two sets  $\mathcal{A} \subset \mathbb{R}^n$  and  $\mathcal{B} \subset \mathbb{R}^n$  (henceforth, we symbolize set by capital calligraphic), the Minkowski sum of  $\mathcal{A}$  and  $\mathcal{B}$ , denoted by  $\mathcal{A} \oplus \mathcal{B}$ , is defined as

$$\mathcal{A} \oplus \mathcal{B} := \{a + b \mid a \in \mathcal{A}, b \in \mathcal{B}\}. \tag{1.1}$$

Specially, if one of the sets consists of a unique element, e.g.,  $\mathcal{A} = \{a\}$ , we briefly adopt the notation  $a \oplus \mathcal{B}$  to indicate the translation of set  $\mathcal{B}$  toward  $a$ . It follows immediately from (1.1) that the zero set  $\{0\}$  plays the role of identity element under the additive operation  $\oplus$ . Figure 1 displays the diagrams of Minkowski sum of ball, box and convex polytope in two- and three-dimensions.

Let  $\mathcal{F} \subset \mathbb{R}^n$  be a nonempty set. The support function of  $\mathcal{F}$ , denoted by  $\sigma_{\mathcal{F}} : \mathbb{R}^n \rightarrow (-\infty, +\infty]$ , is defined as

$$\sigma_{\mathcal{F}}(x) := \sup \{\langle x, y \rangle \mid y \in \mathcal{F}\}, \tag{1.2}$$

where  $\langle x, y \rangle := x^\top y$  is the inner product endowed on  $\mathbb{R}^n$ . Because of the closedness and sublinearity of  $\sigma_{\mathcal{F}}$  (see Proposition 2.3 in Sect. 2), it essentially indicates a norm on  $\mathbb{R}^n$  whenever  $\mathcal{F}$  fulfils some prerequisites (e.g., compactness, convexity, and symmetry). Accordingly, the generalized distance from an  $x^0 \in \mathbb{R}^n$  to a set  $\mathcal{C} \subset \mathbb{R}^n$ , denoted by  $d_{\mathcal{F}} : \mathbb{R}^n \rightarrow [0, +\infty]$ , can be quantified by the metric  $\sigma_{\mathcal{F}}$  as follows

$$d_{\mathcal{F}}(x^0; \mathcal{C}) := \inf \{\sigma_{\mathcal{F}}(x - x^0) \mid x \in \mathcal{C}\}. \tag{1.3}$$

Note that the optima of (1.3) (if exist) are coined as the generalized projection of  $x^0$  onto  $\mathcal{C}$ , denoted by  $\Pi_{\mathcal{F}}(x^0; \mathcal{C})$ . Particularly, if  $\mathcal{F} = \{x \in \mathbb{R}^n \mid \|x\|_2 \leq 1\}$  is an origin-centered unit ball under the  $\ell^2$ -norm, the  $\sigma_{\mathcal{F}}$ ,  $d_{\mathcal{F}}$  and  $\Pi_{\mathcal{F}}$  reduce to the canonical Euclidean norm, Euclidean distance and Euclidean projection (also known as orthogonal projection), respectively. The (generalized) projection onto set, especially onto the convex set, is crucial for both theoretical and practical purposes in numerical optimization and computational geometry. For some “simple” sets such as the  $\ell^2$ -norm ball, hyperplane, second-order cone and spectral set, the Euclidean projections onto those sets can be trivially derived by closed-form formulae (see e.g.,

[2, Chapter 28] and the references therein). There is also abundant literature on solving projections onto, e.g., the  $\ell^1$ -norm ball [12, 49], manifolds [30],  $\ell^{1,q}$ -norm ball [47], and isotone projection cone [35]. In other words, the distance (1.3) can be efficiently computed for the above “simple” sets. However, the effort on computing the distance (1.3) may be significantly intensive if  $\mathcal{F}$  and  $\mathcal{C}$  are generic convex sets.

In this paper, our particular interest is to compute the generalized distance from an  $x^0 \in \mathbb{R}^n$  to a convex set with Minkowski sum structure, i.e.,

$$d_{\mathcal{F}}(x^0; \mathcal{C}) = \inf \{ \sigma_{\mathcal{F}}(x - x^0) \mid x \in \mathcal{C} := T_1(\Omega_1) \oplus T_2(\Omega_2) \oplus \dots \oplus T_m(\Omega_m) \}, \tag{1.4}$$

where  $\Omega_i \subset \mathbb{R}^{l_i}$  ( $i = 1, 2, \dots, m$ ) are nonempty closed convex sets with  $\sum_{i=1}^m l_i = l$ , and  $T_i(\Omega_i) \subset \mathbb{R}^n$  denotes the image of  $\Omega_i$  under affine mapping  $T_i$ , i.e.,

$$T_i(\Omega_i) = \{ M_i \omega + a_i \mid \omega \in \Omega_i \}, \quad i = 1, 2, \dots, m \tag{1.5}$$

with  $M_i \in \mathbb{R}^{n \times l_i}$  and  $a_i \in \mathbb{R}^n$ . For brevity, the  $-\Omega_i$  denotes the opposite set of  $\Omega_i$ , i.e., the case of  $M_i = -I_n$  and  $a_i = \mathbf{0}$  in (1.5). Due to the closedness and convexity of  $\Omega_i$ , it follows from Lemma 2.2 (see Sect. 2) that those  $T_i(\Omega_i)$  in (1.5) and  $\mathcal{C}$  in (1.4) are closed convex sets. Problem (1.4) has plenty of applications in robotics. For example, to guarantee a robot operate freely in workspace with obstacles, collision detection is conducted to inspect whether the robot and obstacles are overlapped [17]. Concretely, let  $\Omega_i \subset \mathbb{R}^n$  ( $i = 1, 2$ ) be convex sets. The collision detection, which is measured by the distance between  $\Omega_1$  and  $\Omega_2$ , can be formalized as (see e.g., [8, 51] for details)

$$d_{\mathcal{F}}(0; \mathcal{C}) = \inf \{ \sigma_{\mathcal{F}}(x) \mid x \in \mathcal{C} := \Omega_1 \oplus (-\Omega_2) \}, \tag{1.6}$$

which is a special case of (1.4) with specifications  $m = 2$ ,  $x^0 = 0$ ,  $M_1 = -M_2 = -I_n$  and  $a_1 = a_2 = 0$ .

However, the (1.4) is nontrivial to be tackled, even if all  $T_i$ 's are identity mappings. Take the sets in Fig. 1 as examples, although those  $\Omega_i$ 's are simple enough in the sense that the projection onto individual set admits closed-form solutions or can be easily solved up to high precisions, the projection onto their Minkowski sum set (i.e., the  $\Omega_1 \oplus \Omega_2 \oplus \Omega_3$  in Fig. 1) is challenging to acquire. Typically, the projection onto a set with Minkowski sum structure does not possess explicit formula and should be approximated numerically. A large number of methods have been devised for solving (1.6) (a special case of (1.4)) with compact convex sets (especially for polytopes) over the past decades. The existing methods for solving (1.6) can be roughly classified into two categories: optimization-based methods [1, 13, 32, 34, 43, 50, 52] and geometry-based methods [8, 19–21, 31, 39, 51]. Empirically, the optimization-based methods can be adaptive for (1.6) with generic convex sets in high dimension but may be computational intensive, whilst the geometry-based ones usually perform efficiently but are confined to (1.6) with low-dimensional structured sets (e.g., 2D/3D polytopes).

Gilbert et al. [21] proposed a recursion (named as GJK method) for solving (1.6) by approximating  $\mathcal{C}$  as simplices. It can also be extended to solve (1.4) with  $n = 3$  and  $\sigma_{\mathcal{F}}(\cdot) = \|\cdot\|_2$ . Some improvements of GJK method were explored in, e.g., [8,

20, 39]. Recently, Qin and An [40] proposed a fast recursion for solving the dual of (1.4) by using Nesterov’s gradient-like method [36] and smoothing technique [37]. Herein, our paper is devoted to tackling (1.4) with generic component sets by reformulating it as a separable convex optimization, and solving it by the recent state-of-the-art first-order optimization methods with  $O(1/k)$  convergence rate. Moreover, the proposed methods favour the parallel computing system [4, 6] when the number of component sets  $\Omega_i$ ’s in (1.4) is huge-size.

The rest of the paper is organized as follows. In Sect. 2, some basic definitions and propositions are stated for the sequel discussion. In Sect. 3, the reformulations of (1.4) in primal, dual, and primal-dual fashions are presented, followed by the implementation details on handling (1.4) in Sect. 4. In Sect. 5, numerical simulations are conducted to illustrate the performances of alternating direction method of multipliers (ADMM) and primal-dual hybrid gradient method (PDHG). Finally, some concluding remarks are drawn in Sect. 6.

## 2 Preliminaries

In this section, we summarize some basic concepts and their properties that will be useful for the sequel discussions.

For any vector  $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ , let  $\|x\|_p := (\sum_{i=1}^n |x_i|^p)^{1/p}$  ( $1 \leq p < \infty$ ) denote the  $\ell^p$ -norm and  $\|x\|_\infty := \max_{i=1, \dots, n} |x_i|$  denote the  $\ell^\infty$ -norm. Given a symmetric matrix  $Q \in \mathbb{R}^{n \times n}$ , the  $Q > 0$  (resp.  $Q \geq 0$ ) denotes the positive definite (resp. positive semi-definite) matrix. For any  $Q > 0$ , we denote by  $\|x\|_Q := (x^T Q x)^{1/2}$  the  $Q$ -norm of  $x$ . We use  $\text{diag}(x)$  to represent a diagonal matrix whose diagonal elements are  $x_i$ ’s. Let  $\mathbf{1}$  and  $I$  denote the all-one and identity matrix (whose dimensions will be clear from the context), respectively.

We now review some basic definitions and properties from convex analysis (see e.g., the monograph [2]).

**Definition 2.1** Let  $\mathcal{C} \subset \mathbb{R}^n$  be a set. Then

- i)  $\mathcal{C}$  is closed if the limit of any convergent sequence in  $\mathcal{C}$  belongs to  $\mathcal{C}$ .
- ii)  $\mathcal{C}$  is compact if any sequence of  $\mathcal{C}$  has a cluster point in  $\mathcal{C}$  (or equivalently,  $\mathcal{C}$  is closed and bounded).
- iii) The convex hull of  $\mathcal{C}$ , denoted by  $\text{conv}(\mathcal{C})$ , is the intersection of all convex sets containing  $\mathcal{C}$  (or equivalently, the smallest convex set containing  $\mathcal{C}$ ). Particularly, the  $\text{conv}(\mathcal{C})$  is a polytope if  $\mathcal{C}$  comprises a finite number of elements.
- iv) The decomposition of  $\mathcal{C}$ , denoted by  $\mathfrak{D}(\mathcal{C})$ , is defined as

$$\mathfrak{D}(\mathcal{C}) := \{ \mathcal{C}_i \subset \mathbb{R}^n \mid \bigcup_{i=1}^m \mathcal{C}_i = \mathcal{C}, \text{ and } \mathcal{C}_i \cap \mathcal{C}_j = \emptyset \text{ for any } i \neq j \}.$$

Furthermore, the convex decomposition of  $\mathcal{C}$ , denoted by  $\widehat{\mathfrak{D}}(\mathcal{C})$ , is a decomposition of  $\mathcal{C}$  with only convex component sets, i.e.,

$$\widehat{\mathfrak{D}}(\mathcal{C}) := \{C_i \subset \mathbb{R}^n \mid C_i \in \mathfrak{D}(\mathcal{C}), \text{ and all } C_i\text{'s are convex sets}\}.$$

Besides the commutative and associative properties, the Minkowski sum operation “ $\oplus$ ” admits the following important propositions.

**Lemma 2.2** *Let  $C_i \subset \mathbb{R}^i (i = 1, 2)$  be nonempty sets (possibly nonconvex).*

- i)  $\text{conv}(C_1 \oplus C_2) = \text{conv}(C_1) \oplus \text{conv}(C_2)$ .
- ii) *If  $C_i (i = 1, 2)$  are convex (resp. compact) sets and  $T_i : \mathbb{R}^i \rightarrow \mathbb{R}^n (i = 1, 2)$  are affine mappings, then  $T_1(C_1) \oplus T_2(C_2)$  is a convex (resp. compact) set.*

More generally, the above lemma remains true on the occasion of a finite number of sets or affine mappings.

Let  $f : \mathbb{R}^n \rightarrow (-\infty, +\infty]$ . The domain and epigraph of  $f$  are defined by  $\text{dom}(f) := \{x \in \mathbb{R}^n \mid f(x) < +\infty\}$  and  $\text{epi}(f) := \{(x, y) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leq y\}$ , respectively. If the  $\text{dom}(f)$  is nonempty, then  $f$  is said to be proper. If the  $\text{epi}(f)$  is closed, then  $f$  is said to be closed. The indicator function of a set  $C \subset \mathbb{R}^n$ , denoted by  $\iota_C$ , is defined as

$$\iota_C(x) := \begin{cases} 0, & \text{if } x \in C, \\ +\infty, & \text{otherwise.} \end{cases} \tag{2.1}$$

The following properties and examples of support function  $\sigma_{\mathcal{F}}$  can be referred to, e.g., [2].

**Proposition 2.3** *Let  $\mathcal{F} \subset \mathbb{R}^n$  be a nonempty set. Then,*

- i)  $\sigma_{\mathcal{F}}$  is closed and convex.
- ii)  $\sigma_{\mathcal{F}}$  is finite everywhere if and only if  $\mathcal{F}$  is a bounded set.
- iii)  $\sigma_{\mathcal{F}}$  is Lipschitz continuous with  $\sup_{x \in \mathcal{F}} \|x\|$  as Lipschitz constant if  $\mathcal{F}$  is a compact convex set.
- iv)  $\sigma_{\mathcal{F}}$  is sublinear, i.e.,  $\sigma_{\mathcal{F}}(\alpha_1 x_1 + \alpha_2 x_2) \leq \alpha_1 \sigma_{\mathcal{F}}(x_1) + \alpha_2 \sigma_{\mathcal{F}}(x_2)$  for any scalars  $\alpha_i \geq 0$  and vectors  $x_i \in \mathbb{R}^n (i = 1, 2)$ .
- v)  $\sigma_{\mathcal{F}}$  is a norm on  $\mathbb{R}^n$  if  $\mathcal{F}$  is compact, convex, symmetric (i.e.,  $\mathcal{F} = -\mathcal{F}$ ) and contains the origin as an interior point.

**Example 2.1** Let  $\mathcal{F} \subset \mathbb{R}^n$  be a nonempty set.

- i) If  $\mathcal{F} = \{x \in \mathbb{R}^n \mid \|x\|_{\ddagger} \leq 1\}$  is a unit ball associated with a norm  $\|\cdot\|_{\ddagger}$ , then the support function  $\sigma_{\mathcal{F}}(x) = \|x\|_{\ddagger}$ , where  $\|\cdot\|_{\ddagger}$  is the dual norm of  $\|\cdot\|_{\ddagger}$ . For instance, if  $\mathcal{F} = \{x \in \mathbb{R}^n \mid \|x\|_Q \leq 1\}$  with  $Q > 0$ , then  $\sigma_{\mathcal{F}}(x) = \|x\|_{Q^{-1}}$ ; if  $\mathcal{F} = \{x \in \mathbb{R}^n \mid \|x\|_p \leq 1\}$  with  $p \in [1, \infty]$ , then  $\sigma_{\mathcal{F}}(x) = \|x\|_{\frac{p}{p-1}}$ . Herein, the  $\frac{p}{p-1}$  takes value  $\infty$  (resp. 1) when  $p = 1$  (resp.  $p = \infty$ ).

- ii) If  $\mathcal{F}$  is a closed convex cone, then the support function  $\sigma_{\mathcal{F}}(x) = \iota_{\mathcal{F}^\circ}(x)$ , where  $\mathcal{F}^\circ := \{x \in \mathbb{R}^n \mid x^\top y \leq 0, \forall y \in \mathcal{F}\}$  denotes the polar cone of  $\mathcal{F}$ .
- iii) If  $\mathcal{F} = \{x \in \mathbb{R}^n \mid Ax = b\}$  is an affine set with  $A \in \mathbb{R}^{m \times n}$ , then the support function

$$\sigma_{\mathcal{F}}(x) = \begin{cases} b^\top z, & \text{if } x = A^\top z, \\ +\infty, & \text{otherwise.} \end{cases} \tag{2.2}$$

As aforementioned, the generalized projection  $\Pi_{\mathcal{F}}(\cdot; \mathcal{C})$  in (1.3) reduces to the canonical Euclidean projection when  $\mathcal{F} = \{x \in \mathbb{R}^n \mid \|x\|_2 \leq 1\}$ . For notational brevity, we use  $\Pi(\cdot; \mathcal{C})$  for the Euclidean projection operator. The followings are some examples of Euclidean projections possessing closed-form formulae. The interested reader is referred to, e.g., [2, Chapter 28], for more examples.

**Example 2.2** Let  $\mathcal{C} \subset \mathbb{R}^n$  be a nonempty closed convex set.

- i) If  $\mathcal{C} = \{x \in \mathbb{R}^n \mid a \leq x \leq b\}$  is a box with  $a \in \mathbb{R}^n$  and  $b \in \mathbb{R}^n$ , then  $[\Pi(x; \mathcal{C})]_i = \text{median}\{a_i, x_i, b_i\}$  for  $i = 1, 2, \dots, n$ .
- ii) If  $\mathcal{C} = \{x \in \mathbb{R}^n \mid \|x\|_2 \leq \alpha\}$  is a ball with  $\alpha > 0$ , then  $\Pi(x; \mathcal{C}) = \min\left(1, \frac{\alpha}{\|x\|_2}\right)x$ .
- iii) If  $\mathcal{C} = \{x \in \mathbb{R}^n \mid Ax = b\}$  is an affine set with  $A \in \mathbb{R}^{m \times n}$  and  $b \in \mathbb{R}^m$ , then  $\Pi(x; \mathcal{C}) = x - A^\dagger(Ax - b)$ , where  $A^\dagger$  is the Moore–Penrose pseudo-inverse<sup>1</sup> of  $A$ . Particularly, if  $\mathcal{C} = \{x \in \mathbb{R}^n \mid a^\top x = b, a \neq 0 \in \mathbb{R}^n\}$  is a nonvertical hypeplane, then  $\Pi(x; \mathcal{C}) = x - \frac{a^\top x - b}{\|a\|_2^2} a$ .
- iv) If  $\mathcal{C} = \{(x, t) \in \mathbb{R}^n \times \mathbb{R} \mid \|x\|_2 \leq \alpha t\}$  is a second-order cone with  $\alpha > 0$ , then

$$\Pi(x; \mathcal{C}) = \begin{cases} (x, t), & \text{if } \|x\|_2 \leq \alpha t; \\ (0, 0), & \text{if } \|x\|_2 \leq -t/\alpha; \\ \frac{\alpha\|x\|_2 + t}{(\alpha^2 + 1)\|x\|_2} (\alpha x, \|x\|_2), & \text{otherwise.} \end{cases}$$

The followings are some propositions of Euclidean projection onto convex set with Minkowski sum structures.

**Proposition 2.4** Let  $\alpha > 0$  be a scalar,  $\mathcal{A}$  and  $\mathcal{C}$  be nonempty closed convex sets in  $\mathbb{R}^n$ , and  $x, x^0$  be vectors in  $\mathbb{R}^n$ .

- i) If  $\mathcal{A} = x^0 \oplus \mathcal{C}$ , then  $\Pi(x; \mathcal{A}) = x^0 + \Pi(x - x^0; \mathcal{C})$ .
- ii) If  $\mathcal{A} \perp \mathcal{C}$  (i.e.,  $a^\top c = 0$  for any  $a \in \mathcal{A}$  and  $c \in \mathcal{C}$ ), then  $\Pi(x; \mathcal{A} \oplus \mathcal{C}) = \Pi(x; \mathcal{A}) + \Pi(x; \mathcal{C})$ .

<sup>1</sup> The Moore–Penrose pseudo-inverse exists and is unique for any  $A \in \mathbb{C}^{m \times n}$ . Let  $\text{rank}(A) = r$  and  $A = U\Sigma V^*$  be the singular value decomposition with  $U \in \mathbb{C}^{m \times r}$ ,  $\Sigma \in \mathbb{R}^{r \times r}$  and  $V \in \mathbb{C}^{n \times r}$ . Then  $A^\dagger = V\Sigma^{-1}U^*$ . Particularly,  $A^\dagger = (A^*A)^{-1}A^*$  when  $A$  is of full column rank, and  $A^\dagger = A^*(AA^*)^{-1}$  when  $A$  is of full row rank.

iii) If  $\mathcal{A} = \mathcal{B} \oplus \mathcal{C}$  with  $\mathcal{B} = \{x \in \mathbb{R}^n \mid \|x\|_2 \leq \alpha\}$ , then  $\Pi(x; \mathcal{A}) = \Pi(x; \mathcal{C}) + \min \left( 1, \frac{\alpha}{\|x - \Pi(x; \mathcal{C})\|_2} \right) (x - \Pi(x; \mathcal{C}))$ .

**Proof** The proofs of i) and ii) can be referred to [2, Propositions 28.1 and 28.6], respectively. The assertion iii) can be proved as follows.

By combining the definition of Euclidean projection with the Minkowski sum structure of  $\mathcal{A}$ , we have  $\Pi(x; \mathcal{A}) = \arg \min_{y \in \mathcal{A}} \|y - x\|_2 = w_1^* + w_2^*$ , where  $(w_1^*, w_2^*)$  is an optimum of

$$\min_{w_1 \in \mathcal{B}, w_2 \in \mathcal{C}} \|w_1 + w_2 - x\|_2^2.$$

By invoking the first-order optimality conditions, we have

$$w_1^* = \Pi(w_1^* - \beta_1(w_1^* + w_2^* - x); \mathcal{B}), \quad \forall \beta_1 > 0, \tag{2.3}$$

$$w_2^* = \Pi(w_2^* - \beta_2(w_1^* + w_2^* - x); \mathcal{C}), \quad \forall \beta_2 > 0. \tag{2.4}$$

Without loss of generality, by setting  $\beta_1 = 1$  in (2.3), it follows from Example 2.5 ii) that

$$w_1^* = \Pi(x - w_2^*; \mathcal{B}) = \begin{cases} \frac{\alpha(x - w_2^*)}{\|x - w_2^*\|_2}, & \text{if } \|x - w_2^*\|_2 > \alpha, \\ x - w_2^*, & \text{otherwise.} \end{cases} \tag{2.5}$$

In the case of  $\|x - w_2^*\|_2 > \alpha$ , by substituting  $w_1^* = \frac{\alpha(x - w_2^*)}{\|x - w_2^*\|_2}$  into (2.4) and setting  $\beta_2 = (1 - \frac{\alpha}{\|x - w_2^*\|_2})^{-1}$ , we obtain  $w_2^* = \Pi(x; \mathcal{C})$ , and thus  $w_1^* = \frac{\alpha(x - \Pi(x; \mathcal{C}))}{\|x - \Pi(x; \mathcal{C})\|_2}$ . Finally, we get

$$w_1^* + w_2^* = \begin{cases} \Pi(x; \mathcal{C}) + \frac{\alpha(x - \Pi(x; \mathcal{C}))}{\|x - \Pi(x; \mathcal{C})\|_2}, & \text{if } \|x - \Pi(x; \mathcal{C})\|_2 > \alpha, \\ x, & \text{otherwise,} \end{cases} \tag{2.6}$$

which completes the proof. □

The set  $\mathcal{A}$  in Proposition 2.6 iii) is widely known as the offsetting of  $\mathcal{C}$  by a radius  $\alpha$ . As a special case of Proposition 2.6 i), if  $\mathcal{C} = \{x \in \mathbb{R}^n \mid \|x\|_2 \leq \alpha\}$ , it follows that  $\mathcal{A}$  is an  $x^0$ -centred  $\ell^2$ -norm ball with radius  $\alpha$ . Accordingly, the following corollary holds.

**Corollary 2.5** *If  $\mathcal{A} = \{x \in \mathbb{R}^n \mid \|x - x^0\|_2 \leq \alpha\}$  with  $\alpha > 0$ , then  $\Pi(x; \mathcal{A}) = x^0 + \min \left( 1, \frac{\alpha}{\|x - x^0\|_2} \right) (x - x^0)$ .*

We now present some preliminaries involving functions and operators, which will be useful in the sequel discussion.

The subdifferential of  $f$ , denoted by  $\partial f : \mathbb{R}^n \rightarrow 2^{\mathbb{R}^n}$ , is defined as

$$\partial f(x) := \{ \xi \in \mathbb{R}^n \mid f(y) \geq f(x) + \xi^T(y - x), \forall y \in \mathbb{R}^n \}, \quad \forall x \in \text{dom}(f). \quad (2.7)$$

If  $f : \mathbb{R}^n \rightarrow (-\infty, +\infty]$  is a convex function, then the subdifferential  $\partial f(x)$  is a non-empty compact convex set for all  $x \in \mathbb{R}^n$  (see e.g., [2] for details). We state below some examples of subdifferential.

**Example 2.3** Let  $C \subset \mathbb{R}^n$  be a nonempty closed convex set. Then,

$$\begin{aligned} \partial \iota_C(x) &= \begin{cases} N_C(x) := \{ \xi \in \mathbb{R}^n \mid \xi^T(y - x) \leq 0, \forall y \in C \}, & \text{if } x \in C, \\ \emptyset, & \text{otherwise,} \end{cases} \\ \partial \sigma_C(x) &= \begin{cases} C, & \text{if } x = 0, \\ F_C(x) := \{ \xi \in C \mid \xi^T x = \sigma_C(x) \}, & \text{otherwise,} \end{cases} \end{aligned}$$

where  $N_C$  and  $F_C$  are the so-called normal cone and exposed face of  $C$ .

**Example 2.4** Let  $\| \cdot \|_{\dagger}$  be a norm of  $\mathbb{R}^n$  and  $\| \cdot \|_{\ddagger}$  be its dual norm. Then

$$\partial \|x\|_{\dagger} = \begin{cases} \{ \xi \in \mathbb{R}^n \mid \|\xi\|_{\ddagger} \leq 1 \}, & \text{if } x = 0, \\ \{ \xi \in \mathbb{R}^n \mid \|\xi\|_{\ddagger} \leq 1, \xi^T x = \|x\|_{\dagger} \}, & \text{otherwise.} \end{cases} \quad (2.8)$$

In particular, the subdifferential of  $\ell^p$ -norm ( $p \in [1, +\infty]$ ) at origin is the unit ball associated with its dual norm, i.e.,  $\partial \|0\|_p = \{ \xi \in \mathbb{R}^n \mid \|\xi\|_{\frac{p}{p-1}} \leq 1 \}$ , whilst the subdifferential of  $\ell^p$ -norm at  $x \neq 0$  can be stated as

$$\begin{aligned} \partial \|x\|_1 &= \left\{ \xi \in \mathbb{R}^n \mid \xi_i = \frac{x_i}{|x_i|} \text{ for } x_i \neq 0, \text{ and } \xi_i \in [-1, 1] \text{ for } x_i = 0 \right\}, \\ \partial \|x\|_p &= \left\{ \xi \in \mathbb{R}^n \mid \xi_i = \frac{x_i |x_i|^{p-2}}{\|x\|_p^{p-1}} \text{ for } x_i \neq 0, \text{ and } \xi_i = 0 \text{ for } x_i = 0 \right\}, \\ \partial \|x\|_{\infty} &= \text{conv} \left( \frac{x_i e_i}{|x_i|} \mid i \in \mathcal{I}(x) \right), \end{aligned}$$

where  $\mathcal{I}(x) := \{ i \mid |x_i| = \|x\|_{\infty} \}$  and  $e_i \in \mathbb{R}^n$  accounts for the canonical basis vector (i.e., the vector whose only nonzero element is 1 in the  $i$ th coordinate). The conjugate of  $f$ , denoted by  $f^*$ , is defined as

$$f^*(\lambda) := \sup \{ x^T \lambda - f(x) \mid x \in \text{dom}(f) \}, \quad \forall \lambda \in \mathbb{R}^n. \quad (2.9)$$

Note that  $f^*$  is always a closed convex function (even if  $f$  is not closed convex). Furthermore, if  $f$  is a closed convex proper function, then  $f^{**} = f$ . It follows by (2.9) that the indicator function  $\iota_C$  and support function  $\sigma_C$  are conjugate with each other. Let  $f : \mathbb{R}^n \rightarrow (-\infty, +\infty]$  be a closed convex proper function. The proximity of  $f$ , denoted by  $\text{prox}_f$ , is defined as



$$\text{prox}_f(x) := \underset{y \in \mathbb{R}^n}{\text{argmin}} \left\{ f(y) + \frac{1}{2} \|y - x\|_2^2 \right\}, \quad \forall x \in \mathbb{R}^n. \tag{2.10}$$

The followings are some properties of proximity which will be useful for the sequel analysis. The interested reader is referred to [11, Table 10.2] for more instances and properties of proximity operators in Hilbert space.

**Proposition 2.6** *Let  $\alpha > 0$  and  $\gamma \in \mathbb{R}$  be scalars,  $b \in \mathbb{R}^n$  be a vector,  $A \in \mathbb{R}^{n \times m}$  be a matrix,  $C \subset \mathbb{R}^n$  be a nonempty closed convex set, and  $\varphi : \mathbb{R}^n \rightarrow (-\infty, +\infty]$  be a closed convex proper function.*

i) *If  $f(x) = \varphi(Ax + b)$  with  $A$  satisfying  $AA^T = \alpha I_n$ , then*

$$\text{prox}_f(x) = x - \alpha^{-1} A^T (Ax + b - \text{prox}_{\alpha\varphi}(Ax + b)). \tag{2.11}$$

ii) *If  $f(x) = \varphi(x) + \frac{\alpha}{2} \|x\|_2^2 + b^T x + \gamma$ , then  $\text{prox}_f(x) = \text{prox}_{\frac{\varphi}{\alpha+1}} \left( \frac{x-b}{\alpha+1} \right)$ .*

iii) *Recall that  $\varphi^*$  is the conjugate of  $\varphi$ , then  $x = \text{prox}_{\alpha\varphi}(x) + \alpha \text{prox}_{\frac{\varphi^*}{\alpha}} \left( \frac{x}{\alpha} \right)$  for any  $x \in \mathbb{R}^n$ .*

iv) *If  $f(x) = \iota_C(x)$  is indicator function, then  $\text{prox}_f(x) = \Pi(x; C)$ ; and if  $f(x) = \sigma_C(x)$  is support function, then  $\text{prox}_f(x) = x - \Pi(x; C)$ .*

Note that Proposition 2.10 iii) is the celebrated Moreau’s identity, which indicates the relationship of proximities between  $\varphi$  and its conjugate.

### 3 Reformulations of (1.4) into variant forms

In this section, we first reformulate (1.4) into variant forms, and then deduce the equivalent monotone inclusion problems by employing the first-order optimality conditions. It will be demonstrated that the problem (1.4) can be tractably solved by a large family of splitting methods.

We start the discussion by restating (1.4) as follows

$$\min \sigma_{\mathcal{F}}(x - x^0) \tag{3.1a}$$

$$\text{subject to } x \in T_1(\Omega_1) \oplus T_2(\Omega_2) \oplus \dots \oplus T_m(\Omega_m). \tag{3.1b}$$

The existence of optima of (3.1) follows immediately by the Weierstrass’ theorem (see e.g., [2, Chapter 11]).

**Proposition 3.1** *Problem (3.1) admits at least one optimum if one of the following conditions is satisfied: (i)  $T_i(\Omega_i)$  ( $i = 1, 2, \dots, m$ ) are compact sets; (ii)  $0 \in \text{int}(\mathcal{F})$ .*

**Proof**

- i) By the compactness of  $T_i(\Omega_i)$  and Lemma 2.2 ii), we have that the Minkowski sum of  $T_i(\Omega_i)$  is a compact set. Moreover, as  $\sigma_{\mathcal{F}}$  is closed (see Proposition 2.3), the assertion holds by the Weierstrass' theorem.
- ii) If  $0 \in \text{int}(F)$ , there exists a neighbourhood of 0, denoted by  $\mathcal{N}_\delta(0)$ , satisfying  $\mathcal{N}_\delta(0) \subset \mathcal{F}$ . For any  $x \in \mathbb{R}^n$ , we have  $y^0 := \frac{\delta(x-x^0)}{2\|x-x^0\|_2} \in \mathcal{N}_\delta(0)$ . Accordingly,

$$\sigma_{\mathcal{F}}(x - x^0) = \sup_{y \in \mathcal{F}} y^\top(x - x^0) \geq \sup_{y \in \mathcal{N}_\delta(0)} y^\top(x - x^0) \geq (y^0)^\top(x - x^0) = \frac{\delta}{2} \|x - x^0\|_2,$$

which indicates that  $\sigma_{\mathcal{F}}(\cdot - x^0)$  is coercive (i.e.,  $\lim_{\|x\|_2 \rightarrow \infty} \sigma_{\mathcal{F}}(x - x^0) = +\infty$ ). Again, the assertion (ii) follows from the Weierstrass' theorem. □

Because of the affine property of  $T_i$ , it is obvious that Proposition 3.1 (i) holds if all  $\Omega_i$ 's are compact sets. Under some stringent premise, the uniqueness of optimum of (3.1) can be guaranteed. For instance, if  $\Omega$  and  $\mathcal{F}$  in (3.1) are convex, normally smooth and round,<sup>2</sup> then (3.1) admits a unique optimum [40].

By recalling the affine mappings  $T_i$  in (1.5), the constraint (3.1b) can be described as

$$x = \sum_{i=1}^m (M_i \omega_i + a_i), \quad \omega_i \in \Omega_i, \quad i = 1, 2, \dots, m. \tag{3.2}$$

Hereafter, for notational convenience, we denote  $\omega \in \mathbb{R}^l$ ,  $a \in \mathbb{R}^n$ ,  $M \in \mathbb{R}^{n \times l}$  and  $\Omega \subset \mathbb{R}^l$  as follows

$$\omega := \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_m \end{pmatrix}, \quad a := \sum_{i=1}^m a_i, \quad M := (M_1, M_2, \dots, M_m) \quad \text{and} \quad \Omega := \Omega_1 \times \dots \times \Omega_m. \tag{3.3}$$

**3.1 Primal reformulation**

With notations in (3.3), the constraint (3.2) can be narrowed down to  $x = M\omega + a$  with  $\omega \in \Omega$ . Accordingly, problem (3.1) can be cast as

$$\min \sigma_{\mathcal{F}}(x - x^0) \quad \text{subject to} \quad x = M\omega + a, \quad \omega \in \Omega, \tag{3.4}$$

or concisely,

---

<sup>2</sup> Let  $\text{bd}(C)$  and  $N_C(x)$  denote the topological boundary and normal cone (see also Example 2.8 for definition) of a set  $C \subset \mathbb{R}^n$ , respectively. The  $C$  is called normally smooth if, for any  $x \in \text{bd}(C)$ , there exists an  $a_x \in \mathbb{R}^n$  such that  $N_C(x) = \text{cone}\{a_x\}$ . The  $C$  is said to be round if  $N_C(x) \neq N_C(y)$  for any  $x, y \in \text{bd}(C)$  and  $x \neq y$ .

$$\min_{\omega \in \mathbb{R}^l} \sigma_{\mathcal{F}}(M\omega + a - x^0) + \iota_{\Omega}(\omega), \tag{3.5}$$

where  $\iota_{\Omega}$  is the indicator function defined in (2.1). For simplicity, the (3.5) is termed as the primal reformulation of (1.4). The first-order optimality conditions of (3.5) read

$$0 \in M^T \partial \sigma_{\mathcal{F}}(M\omega + a - x^0) + \partial \iota_{\Omega}(\omega), \tag{3.6}$$

where  $\partial \iota_{\Omega}$  and  $\partial \sigma_{\mathcal{F}}$  are subdifferential operators (see Example 2.8) of  $\iota_{\Omega}$  and  $\sigma_{\mathcal{F}}$ , respectively. Particularly, if the dynamic set  $\mathcal{F} = \{x \in \mathbb{R}^n \mid \|x\|_{Q^{-1}} \leq 1\}$  with  $Q > 0$  (or equivalently,  $\sigma_{\mathcal{F}}(x) = \|x\|_Q$ ), the (3.5) can be equivalently cast as

$$\min_{\omega \in \mathbb{R}^l} \frac{1}{2} \|M\omega + a - x^0\|_Q^2 + \iota_{\Omega}(\omega), \tag{3.7}$$

which implies that the projection onto Minkowski sum set (1.4) is substantially a constrained least squares problem. Obviously, the objective function of (3.7) is strongly convex if  $M$  is of full column rank, which implies that the (3.7) admits unique optimum. Moreover, a corollary can be immediately deduced for particular  $M$ .

**Corollary 3.2** *If  $\mathcal{F} = \{x \in \mathbb{R}^n \mid \|x\|_{Q^{-1}} \leq 1\}$  with  $Q > 0$ , i.e.,  $\sigma_{\mathcal{F}}(x) = \|x\|_Q$ , and  $M \in \mathbb{R}^{n \times l}$  satisfies  $M^T Q M = \text{diag}\{\frac{1}{v_1} I_{l_1}, \frac{1}{v_2} I_{l_2}, \dots, \frac{1}{v_m} I_{l_m}\}$  with  $v_i > 0$  for all  $i = 1, 2, \dots, m$  (i.e., the columns of  $M$  are  $Q$ -conjugate vectors), then the optimum of (3.5), denoted by  $\omega^* = (\omega_1^*, \omega_2^*, \dots, \omega_m^*)$ , admits the explicit formula*

$$w_i^* = \Pi(v_i M_i^T Q(x^0 - a); \Omega_i), \quad i = 1, 2, \dots, m. \tag{3.8}$$

**Proof** By the first-order optimality conditions of (3.7), we derive

$$0 \in M^T Q(M\omega^* + a - x^0) + \partial \iota_{\Omega}(\omega^*).$$

Furthermore, it follows from the assumption  $M_i^T Q M_i = \frac{1}{v_i} I_{l_i}$  and the fact  $\Omega = \Omega_1 \times \dots \times \Omega_m$  that

$$v_i M_i^T Q(x^0 - a) \in w_i^* + v_i \partial \iota_{\Omega_i}(w_i^*), \quad i = 1, 2, \dots, m,$$

which indicates that  $w_i^* = (I_{l_i} + v_i \partial \iota_{\Omega_i})^{-1}(v_i M_i^T Q(x^0 - a)) = \Pi(v_i M_i^T Q(x^0 - a); \Omega_i)$ . □

The corollary above indicates that if the column vectors of  $M_i$  are  $Q$ -conjugate, then the projection onto a convex set with Minkowski sum structure admits closed-form solution.

### 3.2 Dual reformulation

On the other hand, by deploying Fenchel–Rockafellar duality (see e.g., [2, Chapter 15]), the dual of (3.5) can be formulated as

$$\max_{\lambda \in \mathbb{R}^n} \lambda^\top (a - x^0) - \iota_{\mathcal{F}}(\lambda) - \sigma_{\Omega}(-M^\top \lambda). \tag{3.9}$$

Accordingly, the first-order optimality conditions of (3.9) read

$$0 \in a - x^0 - \partial \iota_{\mathcal{F}}(\lambda) + M \partial \sigma_{\Omega}(-M^\top \lambda). \tag{3.10}$$

The explicit formula of  $\sigma_{\Omega}$  can be available if  $\Omega$  is special set, e.g, ball, closed convex cone, and affine set (see Example 2.4). For example, if  $\Omega$  is a convex cone, the (3.9) can be cast as  $\max_{\lambda \in \mathbb{R}^n} \{ \lambda^\top (a - x^0) - \iota_{\mathcal{F}}(\lambda) - \iota_{\Omega^{\circ}}(-M^\top \lambda) \}$ . Furthermore, if  $\mathcal{F} = \{x \in \mathbb{R}^n \mid \|x\|_{\infty} \leq \alpha\}$  and  $\Omega = \{x \in \mathbb{R}^n \mid Ax \leq 0\}$  be a polyhedral cone, the  $\Omega^{\circ} = \{x \in \mathbb{R}^n \mid x = A^\top z, z \geq 0\}$  is exactly the finite generated cone by columns of  $A$ . Thus, the (3.9) can be reduced to

$$\max_{\lambda \in \mathbb{R}^n} \lambda^\top (a - x^0) \quad \text{subject to} \quad \|\lambda\|_{\infty} \leq \alpha, \quad M^\top \lambda + A^\top z = 0, \quad z \geq 0,$$

which is essentially a linear programming in variables  $(\lambda, z)$ .

### 3.3 Primal-dual reformulation

Moreover, by applying the conjugate (see (2.9) for definition) of  $\sigma_{\mathcal{F}}$  to (3.5), we derive a saddle point problem

$$\min_{\omega \in \mathbb{R}^l} \max_{\lambda \in \mathbb{R}^n} \psi(\omega, \lambda) := \lambda^\top (M\omega + a - x^0) - \iota_{\mathcal{F}}(\lambda) + \iota_{\Omega}(\omega). \tag{3.11}$$

Again, by the first-order optimality conditions, the saddle point problem (3.11) amounts to

$$\begin{cases} 0 \in \partial \iota_{\Omega}(\omega) + M^\top \lambda, \\ 0 \in \partial \iota_{\mathcal{F}}(\lambda) - M\omega - a + x^0, \end{cases}$$

or equivalently,

$$\begin{pmatrix} 0 \\ a - x^0 \end{pmatrix} \in \begin{pmatrix} \partial \iota_{\Omega} & M^\top \\ -M & \partial \iota_{\mathcal{F}} \end{pmatrix} \begin{pmatrix} \omega \\ \lambda \end{pmatrix}. \tag{3.12}$$

As the (3.11) is linearly coupled in  $\omega$  and  $\lambda$ , there always exists a saddle point  $(\omega^*, \lambda^*)$  satisfying (3.12), where  $\omega^*$  and  $\lambda^*$  are solutions of primal and dual problem, respectively (see e.g., [3, Chapter 3]).

**Remark 3.3** In a word, the (1.4) can be equivalently reformulated as variant optimization forms, whose optimality conditions correspond to finding zeros of a series of

monotone inclusion problems, e.g., (3.6), (3.10) and (3.12). To solve those monotone inclusion problems, numerous splitting schemes can be employed, e.g., proximal point method (PPA), forward-backward splitting (FBS), forward-backward-forward splitting (FBFS), Douglas-Rachford splitting (DRS) and Peaceman-Rachford splitting (PRS), to name a few. The interested reader is referred to, e.g., [15, 38, 42], for overviews and recent advances in splitting schemes. Among those splitting schemes for tackling monotone inclusion problems, the most popular ones include the application of PPA to (3.12) which results in the well-known PDHG method [9, 24], the application DRS to (3.10) which results in ADMM [18, 23], the application DRS to (3.6) which results in the Spingarn’s method [46], etc. In a nutshell, the (1.4) can be tractably solved by a large family of first-order optimization methods except for the computational geometry based algorithms.

### 4 Solvers for projection onto Minkowski sum set

As discussed in Sect. 3, the (1.4) can be reformulated into variant manners, then solved by numerous state-of-the-art methods. The analysis and numerical comparisons on all those methods are beyond the scope of this paper. Herein, we focus on ADMM [23] and PDHG [24] for solving (1.4).

#### 4.1 ADMM based solver

We first consider a generic linearly constrained convex optimization problem with separable structure

$$\min \theta_1(x_1) + \theta_2(x_2) \tag{4.1a}$$

$$\text{subject to } A_1x_1 + A_2x_2 = b, x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2, \tag{4.1b}$$

where  $\theta_i : \mathbb{R}^{n_i} \rightarrow (-\infty, +\infty]$  are all closed convex proper functions (possibly non-smooth);  $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$  ( $i = 1, 2$ ) are nonempty closed convex sets;  $A_i \in \mathbb{R}^{l \times n_i}$  ( $i = 1, 2$ ) are full column rank matrices;  $b \in \mathbb{R}^l$ ; and  $n_1 + n_2 = n$ . A large family of instances arising from the area such as compressive sensing, image processing, computer vision and machine learning can be boiled down to the abstract model (4.1). A state-of-the-art algorithm for solving (4.1) is ADMM originated in [18, 23]. Given  $(x_2^k, \lambda^k) \in \mathbb{R}^{n_2} \times \mathbb{R}^l$ , the recursion of ADMM for solving the abstract model (4.1) can be stated as follows:

$$\begin{cases} x_1^{k+1} = \underset{x_1 \in \mathcal{X}_1}{\operatorname{argmin}} L_\beta(x_1, x_2^k, \lambda^k), \\ x_2^{k+1} = \underset{x_2 \in \mathcal{X}_2}{\operatorname{argmin}} L_\beta(x_1^{k+1}, x_2, \lambda^k), \\ \lambda^{k+1} = \lambda^k - \gamma \beta (A_1x_1^{k+1} + A_2x_2^{k+1} - b), \end{cases} \tag{4.2}$$

where  $\beta > 0$  is a penalty parameter and  $\gamma \in (0, \frac{\sqrt{5}+1}{2})$  is a relaxation factor. Herein, the  $L_\beta : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^l \rightarrow \mathbb{R}$  denotes the augmented Lagrangian function of (4.1), which is defined as

$$L_\beta(x_1, x_2, \lambda) := \theta_1(x_1) + \theta_2(x_2) - \lambda^\top(A_1x_1 + A_2x_2 - b) + \frac{\beta}{2}\|A_1x_1 + A_2x_2 - b\|_2^2, \tag{4.3}$$

where  $\lambda \in \mathbb{R}^l$  is the Lagrange multiplier. As ADMM allows us to independently cope with individual sub-function  $\theta_i$  by the aid of augmented Lagrangian function, it has gained much popularity in diverse applications (see, e.g., the overviews [6, 22] and references therein). The convergence of ADMM has been restated in the tutorial [14], and its  $O(1/k)$  convergence rate was established in [25].

Now, we present the implementation details of ADMM on solving (1.4). Essentially, the primal reformulation of (3.1) falls into the framework of (4.1) with the following specifications:

- the variables  $x_1 := x, x_2 := \omega$ , and the abstract sets  $\mathcal{X}_1 = \mathbb{R}^n$  and  $\mathcal{X}_2 = \Omega$ ;
- the objective functions  $\theta_1(x_1) := \sigma_{\mathcal{F}}(x - x^0)$  and  $\theta_2(x_2) := 0$ ;
- the matrices and vector in linear constraint (4.1b) are  $A_1 := I, A_2 := -M$  and  $b := a$ .

Therefore, the resulting subproblems when applying ADMM (or rather, the iterative scheme (4.2)) to (3.4) read

$$\begin{cases} x^{k+1} = \operatorname{argmin}_{x \in \mathbb{R}^n} \{ \sigma_{\mathcal{F}}(x - x^0) + \frac{\beta}{2} \|x - M\omega^k - a - \beta^{-1}\lambda^k\|_2^2 \}, \\ \omega^{k+1} = \operatorname{argmin}_{\omega \in \Omega} \|\omega^{k+1} - M\omega - a - \beta^{-1}\lambda^k\|_2^2, \\ \lambda^{k+1} = \lambda^k - \beta(x^{k+1} - M\omega^{k+1} - a). \end{cases} \tag{4.4}$$

Unfortunately, the  $\omega$ -subproblem in (4.4) is a least squares problem with constraint  $\Omega = \Omega_1 \times \dots \times \Omega_m$ , which probably requires internally nested iteration to derive an approximate solution (albeit the projection onto individual  $\Omega_i$  admits closed-form solution). Note that the compelling numerical performance of ADMM is established in the sense that all resulting subproblems have close-form solutions or can be efficiently approximated by some subroutines. Accordingly, it is sensible to reformulate (3.4) by an alternative fashion. Concretely, by further introducing an auxiliary variable  $z \in \mathbb{R}^l$ , the (3.4) is equivalent to

$$\min \sigma_{\mathcal{F}}(x - x^0) \quad \text{subject to } x = Mz + a, \omega = z, \omega \in \Omega. \tag{4.5}$$

Likewise, the problem (4.5) falls into the abstract model (4.1) with the following specifications:

- the variables  $x_1 := (x, w), x_2 := z$ , and the abstract sets  $\mathcal{X}_1 := \mathbb{R}^n \times \Omega, \mathcal{X}_2 := \mathbb{R}^n$ ;
- the objective functions  $\theta(x_1) := \sigma_{\mathcal{F}}(x - x^0)$  and  $\theta_2(x_2) := 0$ ;
- The matrices and vector with respect to the linear constraint (4.1b) are

$$A_1 := \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}, \quad A_2 := (-Mi) \quad \text{and} \quad b := \begin{pmatrix} a \\ 0 \end{pmatrix}.$$

ADMM is thus applicable to (4.5) with above specifications. More precisely, the augmented Lagrangian function of (4.5) can be specified as

$$L_\beta(x_1, x_2, \lambda) = \sigma_{\mathcal{F}}(x - x^0) - \lambda_1^\top(x - Mz - a) + \frac{\beta}{2} \|x - Mz - a\|_2^2 - \lambda_2^\top(\omega - z) + \frac{\beta}{2} \|\omega - z\|_2^2,$$

where  $\lambda = (\lambda_1, \lambda_2) \in \mathbb{R}^n \times \mathbb{R}^l$  is the Lagrange multiplier. Accordingly, the resulting subproblems when applying ADMM recursion (4.2) to (1.4) read

$$\begin{cases} x^{k+1} = \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ \sigma_{\mathcal{F}}(x - x^0) + \frac{\beta}{2} \|x - Mz^k - a - \beta^{-1} \lambda_1^k\|_2^2 \right\}, \\ \omega^{k+1} = \operatorname{argmin}_{\omega \in \Omega} \|\omega - z^k - \beta^{-1} \lambda_2^k\|_2^2, \\ z^{k+1} = \operatorname{argmin}_{z \in \mathbb{R}^n} \left\{ \|x^{k+1} - Mz - a - \beta^{-1} \lambda_1^k\|_2^2 + \|\omega^{k+1} - z - \beta^{-1} \lambda_2^k\|_2^2 \right\}, \\ \lambda_1^{k+1} = \lambda_1^k - \beta(x^{k+1} - Mz^{k+1} - a), \\ \lambda_2^{k+1} = \lambda_2^k - \beta(\omega^{k+1} - z^{k+1}). \end{cases} \quad (4.6)$$

These resulting subproblems (correspondingly, the  $x$ -,  $\omega$ -, and  $z$ -subproblems) can be further delineated as follows:

- The  $x$ -subproblem can be equivalently solved by the proximity operator

$$\begin{aligned} x^{k+1} &= \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ \sigma_{\mathcal{F}}(x - x^0) + \frac{\beta}{2} \|x - r^k\|_2^2 \right\} \\ \text{[By (2.10)]} &= \operatorname{prox}_{\frac{1}{\beta} \sigma_{\mathcal{F}}(-x^0)}(r^k) \\ \text{[Proposition 2.6 i)]} &= x^0 + \operatorname{prox}_{\frac{\sigma_{\mathcal{F}}}{\beta}}(r^k - x^0) \\ \text{[Proposition 2.6 iii)]} &= r^k - \frac{1}{\beta} \Pi(\beta(r^k - x^0); \mathcal{F}), \end{aligned} \quad (4.7)$$

where  $r^k := Mz^k + a + \lambda_1^k/\beta$ . Accordingly, the computational effort on solving (4.7) is dominated by computing the Euclidean projection onto dynamic set  $\mathcal{F}$ . In the upcoming numerical simulations, we shall consider the cases that  $\mathcal{F} \subset \mathbb{R}^n$  is the  $\ell^1$ -,  $\ell^2$ - and  $\ell^\infty$ -norm unit ball.

- The  $\omega$ -subproblem is equivalent to

$$\omega^{k+1} = \operatorname{argmin}_{\omega \in \Omega} \|\omega - z^k - \beta^{-1} \lambda_2^k\|_2^2 = \Pi(z^k + \beta^{-1} \lambda_2^k; \Omega), \quad (4.8)$$

which corresponds to the Euclidean projection onto  $\Omega$ . Attributed to the Cartesian product  $\Omega = \Omega_1 \times \Omega_2 \times \dots \times \Omega_m$ , the  $\omega^{k+1} = (\omega_1^{k+1}, \omega_2^{k+1}, \dots, \omega_m^{k+1})$  can thus be derived simultaneously by

$$\begin{aligned} \omega_i^{k+1} &= \operatorname{argmin}_{\omega_i \in \Omega_i} \|\omega_i - z_i^k - \beta^{-1}(\lambda_2^k)_i\|_2^2 \\ &= \Pi(z_i^k + \beta^{-1}(\lambda_2^k)_i; \Omega_i), \quad i = 1, 2, \dots, m. \end{aligned} \tag{4.9}$$

- The  $z$ -subproblem amounts to

$$z^{k+1} = \operatorname{argmin}_{z \in \mathbb{R}^n} \{ \|x^{k+1} - Mz - a - \beta^{-1}\lambda_1^k\|_2^2 + \|\omega^{k+1} - z - \beta^{-1}\lambda_2^k\|_2^2 \},$$

which is a positive definite linear system

$$(M^T M + I)z = M^T(x^{k+1} - a - \beta^{-1}\lambda_1^k) + \omega^{k+1} - \beta^{-1}\lambda_2^k. \tag{4.10}$$

There are numerous algorithms catering for the above linear system in numerical computation, e.g., preconditioned conjugate gradient (PCG) method. Summarily, the updates of  $x$ ,  $\omega$  and  $z$  when implementing ADMM can be readily conducted. The projections onto dynamic set  $\mathcal{F}$  in (4.7) and those component sets  $\Omega_i$  in (4.9) can be handled simultaneously. Moreover, a pleasing property of ADMM is that there is only one user-dependent parameter  $\beta$  to be tuned throughout numerical implementation.

### 4.2 PDHG solver

As discussed in Sect. 4.1, ADMM can perform efficiently on solving (1.4) with lower dimensions, e.g., the collision detection in robotics. However, its pitfalls are becoming increasingly evident when the dimension of (1.4) is enlarged. The computational effort on implementing ADMM in each iteration is possibly dominated by solving the linear system (4.10). An alternative is the PDHG method explored in [9, 10, 16, 24, 26], which can be viewed as the application of PPA to the monotone inclusion problem (3.12). Recently, a novel viewpoint was addressed that PDHG can also be interpreted as the application of DRS to a reformulation of primal problem [38].

Given  $(\omega^k, \lambda^k) \in \mathbb{R}^n \times \mathbb{R}^l$ , the recursion of PDHG [24] on solving the primal-dual reformulation of (1.4) (i.e., the (3.11)), is specified as follows

$$\begin{cases} \tilde{\omega}^k = \operatorname{argmin}_{\omega \in \mathbb{R}^l} \left\{ \psi(\omega, \lambda^k) + \frac{1}{2\tau_1} \|\omega - \omega^k\|_2^2 \right\}, \\ \tilde{\lambda}^k = \operatorname{argmax}_{\lambda \in \mathbb{R}^n} \left\{ \psi(2\tilde{\omega}^k - \omega^k, \lambda) - \frac{1}{2\tau_2} \|\lambda - \lambda^k\|_2^2 \right\}, \\ \omega^{k+1} = \omega^k + \rho(\tilde{\omega}^k - \omega^k), \\ \lambda^{k+1} = \lambda^k + \rho(\tilde{\lambda}^k - \lambda^k), \end{cases} \tag{4.11}$$

where  $\psi$  is the saddle point function defined in (3.11), the  $\tau_1 > 0$ ,  $\tau_2 > 0$  are parameters satisfying  $\tau_1 \tau_2 \|M^T M\|_2 \leq 1$ , and  $\rho \in (0, 2)$  is a relaxation factor. By elaborating the subproblems in (4.11), we obtain the following concrete recursion for solving (1.4) by PDHG



$$\begin{cases} \tilde{\omega}^k = \Pi(\omega_i^k - \tau_1 M_i^T \lambda^k; \Omega_i), & i = 1, 2, \dots, m \\ \tilde{\lambda}^k = \Pi(\lambda^k + \tau_2 (M(2\tilde{\omega}^k - \omega^k) + a - x^0); \mathcal{F}), \\ \omega^{k+1} = \omega^k + \rho(\tilde{\omega}^k - \omega^k), \\ \lambda^{k+1} = \lambda^k + \rho(\tilde{\lambda}^k - \lambda^k). \end{cases} \tag{4.12}$$

Intuitively, the recursion of PDHG for solving (1.4) is much easier than that of ADMM. The computational effort of (4.12) is merely dominated by solving the projections onto dynamic set  $\mathcal{F}$  and component sets  $\Omega_i$ 's. Compared to ADMM, PDHG is fairly favourable for handling large-scale problems as the linear systems are not involved throughout iteration. The only drawback of PDHG may be the challenge in tuning parameters  $(\tau_1, \tau_2)$  to obtain compelling numerical performance.

**Remark 4.1** In contrast, Gilbert's method [19] aims to seek the optima of convex quadratic function  $f$  on compact set  $\mathcal{C}$ , i.e.,  $\min_{x \in \mathcal{C}} f(x)$ . The recursion reads

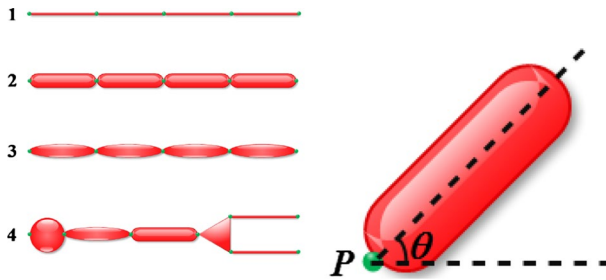
$$x^{k+1} = x^k + \alpha_k [s_{\mathcal{C}}(-\nabla f(x^k)) - x^k], \tag{4.13}$$

where  $\alpha_k \in (0, 1]$  is the stepsize computed by a prior criterion, and  $s_{\mathcal{C}}(y) := \{x \in \mathcal{C} \mid x^T y = \sup_{z \in \mathcal{C}} z^T y\}$  denotes the contact mapping of  $\mathcal{C}$ . The recursion (4.13) amounts to minimizing  $f$  on a polyhedron approximation of  $\mathcal{C}$  by projection gradient method, or can be deemed as a variant of Frank-Wolfe method (see e.g., [40] for discussion). An improvement of (4.13) can be referred to, e.g. [21], for solving problem (1.4) in 3D case. The worst-case convergence rate of (4.13) is proven to be  $O(\frac{1}{\epsilon})$ . Alternatively, Qin and An [40] deployed Nesterov gradient-like method [36, Chapter 2.2] to the dual reformulation of (1.4), which admits the worst-case  $O(\frac{1}{\sqrt{\epsilon}} \ln(\frac{1}{\epsilon}))$  convergence rate. Both methods are essentially the gradient-like methods in either primal or dual variable. Numerically, as shown in [40], Gilbert's method may converge slowly near optima due to zigzagging phenomenon.

ADMM and PDHG are state-of-the-art splitting methods involving both primal and dual variables in recursion, and admit the worst-case  $O(\frac{1}{\epsilon})$  convergence rate for generic convex programming. PDHG is typically more popular for solving large-scale practical problems as its recursion only involves matrix-vector operations.

## 5 Numerical simulations

In this section, we conduct some numerical experiments involving the abstract model (1.4) so as to demonstrate the performances of PDHG and ADMM. Essentially, the model (1.4) captures numerous important real-world applications in robotics [29] and mechanics [48]. We test a 2D path planning problem in robotics and some high-dimensional synthetic problems involving (1.4). All codes are written by MATLAB 7.9 and all experiments are conducted on a Lenovo personal computer with Intel Core (TM) CPU 2.30GHZ and 8G memory.



**Fig. 2** Left: Diagram representations of manipulators with variant shapes of links. Right: A link  $\mathcal{A}(P, \theta)$  with Minkowski sum structure

### 5.1 Path planning

Path planning plays a fundamental role in robotics and has been extensively studied in real-world applications. It aims to ensure the agents (e.g., manipulator, robot) move freely in a (static or dynamic) workspace without hitting obstacles. We herein consider the scenarios of an agent (more precisely, a manipulator) with a single link and  $m$  degree-of-freedom (DoF) links in a static workspace.

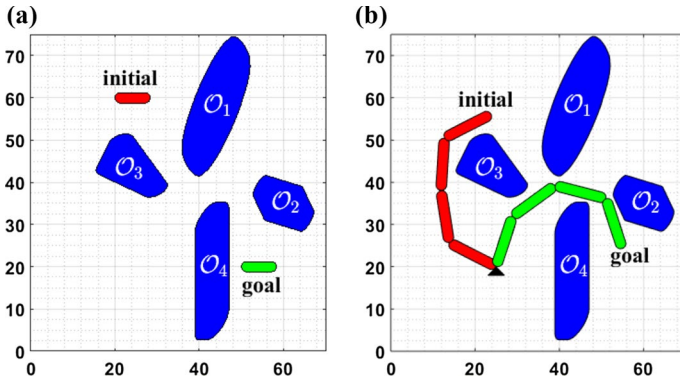
Let  $\Omega \subset \mathbb{R}^n$  denote an  $n$ -dimensional ( $n = \{2, 3\}$  for practical applications in robotics) workspace and  $\mathcal{O}_i \subset \Omega$  ( $i = 1, 2, \dots, l$ ) are closed sets representing stationary obstacles. Without loss of generality, all  $\mathcal{O}_i$ 's are assumed to be convex.<sup>3</sup> A manipulator with  $m$  DoF can be described by a collection of rigid bodies  $\mathcal{A}_i \subset \mathbb{R}^n$  ( $i = 1, 2, \dots, m$ ). Figure 2 shows the diagram representations of some manipulators with variant shapes. The links (also called arms) of manipulator, which are concatenated by the joints (green points in Fig. 2), are typically mimicked by simple sets such as line segment, polytope, and ellipsoid.

In path planning, a probabilistic roadmap is constructed as an undirected graph [27], followed by a collision-free path by repetitively checking the distances between obstacles and links. A mathematical model can be formulated by calculating the distance function

$$d(\mathcal{O}, \mathcal{A}) := \inf \{ \sigma_{\mathcal{F}}(x, y) \mid x \in \mathcal{O}, y \in \mathcal{A} \}, \tag{5.1}$$

where  $\sigma_{\mathcal{F}}$  is the generalized distance metric defined in (1.2). The  $\mathcal{O} := \bigcup_{i=1}^l \mathcal{O}_i$  and  $\mathcal{A} := \bigcup_{i=1}^m \mathcal{A}_i$  denote the collection of stationary obstacles and rigid links, respectively. Unfortunately, the problem (5.1) is prohibitively difficult to be tackled by numerical optimization methods because the constraint  $\mathcal{O}$  or  $\mathcal{A}$  is typically nonconvex, even if all component set  $\mathcal{O}_i$ 's or  $\mathcal{A}_i$ 's are convex. An equivalent formulation of (5.1) is to compute the distances between a series of component sets, i.e.,

<sup>3</sup> If an obstacle  $\mathcal{O}_i$  has nonconvex feature, it can be approximate by convex decomposition (see Definition 2.1). Some literature on deriving convex decomposition of a given set can be referred to, e.g., [5, 28, 41].



**Fig. 3** Examples of path planning in a 2D workspace with four obstacles. **a** Single-link manipulator; **b** 4-DoF manipulator

$$d(\mathcal{O}, \mathcal{A}) = \min\{d(\mathcal{O}_i, \mathcal{A}_j) \mid i = 1, 2, \dots, l; j = 1, 2, \dots, m\}. \tag{5.2}$$

Under the convexity assumption on  $\mathcal{O}_i$ 's and  $\mathcal{A}_i$ 's, the nonconvex optimization problem (5.1) can be possibly handled by solving  $ml$  convex optimization problems. There is abundant literature on solving (5.2) with  $\mathcal{O}_i$ 's and  $\mathcal{A}_i$ 's being ellipsoids or polytopes. Those special shapes are typically resulted by approximating obstacles or links via principal component analysis and linearization technique. The ability to cope with only the ellipsoidal and polytonal obstacles may be not accurate enough. Herein, we further assume that the component sets  $\mathcal{O}_i$ 's and  $\mathcal{A}_i$ 's can be described as the Minkowski sum of convex sets. This assumption is admissible as the manipulators typically admit special geometry shape, e.g., polyhedron and ellipsoid (which can be easily described as Minkowski sum of simplex and ball). In order to formulate a generic shape (possibly non-convex) as the Minkowski sum of "simple" set, a possible idea is first approximating the shape by a union of polyhedrons (e.g., [7, 41]) or ellipsoids (e.g., by using principle component analysis), and then deriving the Minkowski sum description for those polyhedrons and ellipsoids.

Figure 3 shows an environment of 2D workspace  $\Omega = [0, 75]^2$  with four stationary obstacles. The manipulators with single- and 4-DoF links are illustrated in Fig. 3 (a) and (b), respectively. The base of 4-DoF manipulator is fixed at the point (25, 20) and all links of manipulator admit the same shapes. Accordingly, the link  $\mathcal{A}_i$  can be described as a rigid transformation of stencil body, denoted by  $\mathcal{A}(P, \theta)$  (see Fig. 2b). The stencil body is hinged by the location of joint (denoted by  $P$ ), and the phase of dominant axis (denoted by  $\theta$ ). The data for reproducing obstacles  $\mathcal{O}_i$ 's and the stencil body  $\mathcal{A}(P, \theta)$  are reported in Table 1. All obstacles and links are represented as the Minkowski sum of set  $T_i(\Omega_i)$ , where the sets  $\Omega_i$  ( $i = 1, 2, 3$ ) are defined by  $\Omega_1 = \{x \in \mathbb{R}^2 \mid \|x\|_1 \leq 1, x_2 \geq 0\}$ ,  $\Omega_2 = \{x \in \mathbb{R}^2 \mid \|x\|_2 \leq 1\}$  and  $\Omega_3 = \{x \in \mathbb{R}^2 \mid \|x\|_\infty \leq 1\}$ . In Table 1, the parameter  $w$  for  $M_9$  indicates the length of link. We take  $w = 3$  and  $w = 5$  for sigle- and 4-DoF manipulators, respectively. The aim of manipulator in Fig. 4 is to move freely from initial configuration (red location) to goal configuration (green location) without hitting obstacles.

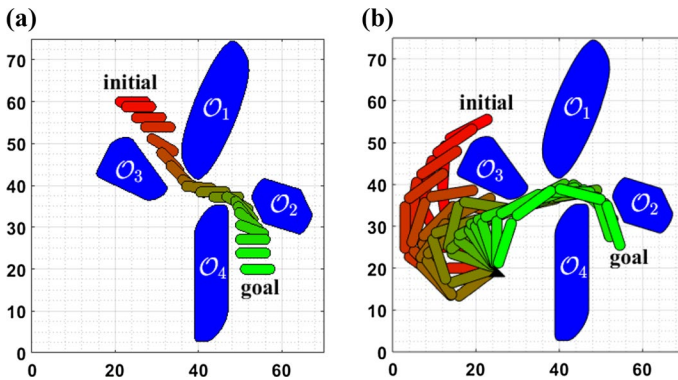
**Table 1** Data of the obstacles  $\mathcal{O}_i (i = 1, 2, 3, 4)$  and links for path planning problem

	Formulation*	$a_i$ and $M_i$ for affine mapping $T_i$ in (1.5)
Obstacle	$\mathcal{O}_1 = T_1(\Omega_2) \oplus T_2(\Omega_2)$	$a_1 = [21; 23], M_1 = [4.8, 4.3; 4.3, 13.3]$
	$\mathcal{O}_2 = T_3(\Omega_3) \oplus T_4(\Omega_2)$	$a_2 = [23; 35], M_2 = [1.5, -0.9; -0.9, 2.5]$
	$\mathcal{O}_3 = T_5(\Omega_1) \oplus T_6(\Omega_2)$	$a_3 = [15; 20], M_3 = [4.3, -1.3; -1.3, 2.8]$
	$\mathcal{O}_4 = T_7(\Omega_3) \oplus T_8(\Omega_2)$	$a_4 = [45; 15], M_4 = [1.5, 0.9; 0.9, 2.5]$
Link	$\mathcal{A}(P, \theta) = T_9(\Omega_3) \oplus T_{10}(\Omega_2)$	$a_5 = [12; 23], M_5 = [5.5, -2.6; -2.6, 8.5]$
		$a_6 = [12; 18], M_6 = [3.0, 0.5; 0.5, 2.0]$
		$a_7 = [28; 12], M_7 = [2.8, 1.3; 1.3, 4.2]$
		$a_8 = [15; 7], M_8 = [1.0, 0; 0, 12]$
		$M_9 = [w, 0; 0, 0.005]G_\theta^{(**)}, a_9 = P^{(***)}$
		$M_{10} = 1.25I_2, a_{10} = 0$

\*The sets  $\Omega_i \subset \mathbb{R}^2 (i = 1, 2, 3)$  are defined by  $\Omega_1 = \{x \in \mathbb{R}^2 \mid \|x\|_1 \leq 1, x_2 \geq 0\}$ ,  $\Omega_2 = \{x \in \mathbb{R}^2 \mid \|x\|_2 \leq 1\}$  and  $\Omega_3 = \{x \in \mathbb{R}^2 \mid \|x\|_\infty \leq 1\}$ .

\*\*The  $G_\theta \in \mathbb{R}^{2 \times 2}$  denotes the two-dimensional Givens transform, which represents an anti-clockwise rotation with respect to an angle  $\theta$ .

\*\*\*The  $P \in \mathbb{R}^2$  indicates the joint of link which is closed to the base of manipulator



**Fig. 4** The motion tracks of manipulators from initial to goal configurations. **a** Single-link manipulator; **b** 4-DoF manipulator

In numerical simulation, we first generate a probabilistic roadmap by the method in [27], then collision-free paths for moving manipulators from initial to goal configurations are probed by repetitively computing the distances  $d(\mathcal{O}_i, \mathcal{A}_j)$  in (5.2) (more precisely, the  $d(\mathcal{O}_i, \mathcal{A}(P, \theta))$  with data in Table 1). The dynamic set  $\mathcal{F}$  is taken as  $\mathcal{F} = \{x \in \mathbb{R}^2 \mid \|x\|_2 \leq 1\}$  for this real-world application, which is equivalent to choosing  $\sigma_{\mathcal{F}}(x) = \|x\|_2$ . Throughout, we choose the penalty parameter  $\beta \equiv 1$  for ADMM, and  $(\tau_1, \tau_2) = (1, 1.1/\|M^T M\|_2)$  for PDHG. The maximum number of iteration is set as 100. The motion tracks of manipulators, which is indicated by the colour changes from initial to goal configurations, are shown in Fig. 4. The computational time to realize this path planning problem is less than 5 s for single-link

manipulator and 10 s for 4-DoF manipulator on our personal laptop computing system.

### 5.2 High-dimensional problems

We now consider the problem (1.4)–(1.5) with synthetic data in high dimensions so as to demonstrate the numerical performances of ADMM and PDHG. Numerical comparisons with some benchmark methods in the literature are also tested, including the Gilbert’s method [19], Qin and An’s method [40]. The datasets of (1.4)–(1.5) are synthesized as follows:

- All  $M_i$ ’s in (1.5) are taken as square matrices, i.e.,  $l_i \equiv n$  for all  $i = 1, 2, \dots, m$ .
- The  $M_i$  and  $a_i$  in (1.5) are randomly generated with entries being independent and identically distributed (i.i.d) variables. Concretely, the entries of  $M_i$  obey the standard normal distribution, and the entries of  $a_i$  follow the uniform distribution on  $[mn, mn + 10]$ .
- The component sets  $\{\Omega_i\}_{i=1}^m$  in (1.4) are randomly taken among some abstract sets, including the unit  $\ell^p$ -norm balls with  $p = \{1, 2, \infty\}$ , the nonnegative orthant, and the second-order cone with  $\alpha = 1$  (see Example 2.5 iv) for definition).

We test the (1.4)–(1.5) on scenarios of  $\sigma_{\mathcal{F}}(\cdot) = \|\cdot\|_p$  ( $p = \{1, 2, \infty\}$ ) with different number of component sets (i.e., the  $m$ ) and variant dimension of variables (i.e., the  $n$ ). Both ADMM and PDHG methods are conducted with their implementation details elucidated in Sect. 4. As Gilbert’s method [19], Qin and An’s method [40] can also be applicable for solving (1.4) with  $\sigma_{\mathcal{F}}(\cdot) = \|\cdot\|_2$ , we test both methods on (1.4)–(1.5) with Euclidean norm. For the involved parameters in test methods, we choose  $\alpha = 0.5$  for Gilbert’s method;  $\mu = 0.001$  for Qin and An’s method;  $\beta = 0.8$ , and  $\gamma = 1.6$  for ADMM;  $\tau_1 = 1$ ,  $\tau_2 = 1.1/\|M^T M\|_2$  and  $\rho = 1.5$  for PDHG. As for the  $nm$ -by- $nm$  linear system (4.10) in ADMM recursion, we first economize the arithmetic operations on deriving the inverse of coefficient matrix  $I + M^T M$  by aid of the Sherman-Morrison formula, then cope with an  $n$ -by- $n$  linear system by using the build-in MATLAB package `linsolve`. All initial points required for test methods are taken as zeros vectors. The stopping criterion is measured by the primal-dual relative gap, which is defined as

$$\text{relgap} = \frac{|\text{Pobj} - \text{Dobj}|}{1 + |\text{Pobj}| + |\text{Dobj}|} < \text{Tol}, \tag{5.3}$$

where  $\text{Pobj}$  and  $\text{Dobj}$  represent the primal and dual objective function values of (1.4), respectively. According to the discussion in Sect. 3, the values of  $\text{Pobj}$  and  $\text{Dobj}$  can be respectively computed by  $\sigma_{\mathcal{F}}(x - x^0)$  and  $\lambda_1^T(a - x^0) - \sigma_{\Omega}(\lambda_2)$  when implementing ADMM recursion, whilst the values of  $\text{Pobj}$  and  $\text{Dobj}$  can be respectively computed by  $\sigma_{\mathcal{F}}(Mw + a - x^0)$  and  $\lambda^T(a - x^0) - \sigma_{\Omega}(-M^T \lambda)$  when implementing PDHG recursion.

We report in Tables 2, 3 and 4 the number of iterations and computing time in seconds (“No. Iteration(CPU)”) when the stopping criterion (5.3) reaches the

**Table 2** Numerical results on solving (1.4)–(1.5) with distance merit  $\sigma_{\mathcal{F}}(\cdot) = \|\cdot\|_2$

$n$	$m$	Tol = $10^{-6}$				Tol = $10^{-8}$			
		Gilbert	Qin&An	ADMM	PDHG	Gilbert	Qin&An	ADMM	PDHG
50	50	213 (0.5)	92 (0.2)	48 (0.1)	41 (0.1)	1594 (3.2)	1132 (2.1)	414 (0.9)	369 (0.48)
	100	197 (1.0)	88 (0.4)	54 (0.3)	45 (0.2)	1528 (6.8)	986 (3.5)	354 (1.7)	292 (0.78)
	500	188 (4.7)	86 (1.9)	52 (1.3)	45 (0.9)	1487 (36.8)	1048 (22.5)	759 (18.1)	604 (12.5)
100	50	192 (1.1)	92 (0.5)	54 (0.3)	37 (0.2)	1675 (8.7)	1123 (5.6)	365 (2.16)	301 (1.2)
	100	211 (2.9)	89 (1.1)	44 (0.6)	35 (0.3)	1544 (21.4)	947 (11.2)	573 (6.7)	468 (4.51)
	500	204 (8.9)	86 (3.3)	43 (1.8)	37 (1.2)	1640 (71.4)	1002 (38.5)	427 (17.8)	352 (10.8)
500	50	187 (11.3)	79 (4.6)	28 (2.5)	24 (1.3)	1496 (89.4)	958 (55.6)	262 (22.7)	217 (12.9)
	100	192 (26.3)	83 (10.4)	29 (4.3)	24 (2.7)	1521 (207.9)	932 (115.8)	270 (40.0)	227 (25.7)
	500	183 (125.4)	78 (44.3)	33 (23.8)	24 (13.1)	1431 (979.8)	965 (547.8)	285 (204.7)	235 (127.2)

**Table 3** Numerical results on solving (1.4)–(1.5) with distance merit  $\sigma_{\mathcal{F}}(\cdot) = \|\cdot\|_1$

$n$	$m$	Tol = $10^{-4}$		Tol = $10^{-5}$		Tol = $10^{-6}$	
		ADMM	PDHG	ADMM	PDHG	ADMM	PDHG
50	50	59 (0.3)	45 (0.1)	208 (0.6)	191 (0.3)	671 (1.3)	612 (0.9)
	100	48 (0.4)	41 (0.2)	156 (0.9)	145 (0.4)	622 (2.9)	589 (1.7)
	500	69 (1.5)	36 (0.5)	160 (3.2)	125 (1.9)	462 (8.7)	434 (6.9)
100	50	80 (0.6)	50 (0.2)	300 (1.9)	192 (0.6)	672 (3.8)	588 (1.8)
	100	61 (0.8)	45 (0.4)	188 (2.1)	166 (1.4)	695 (7.2)	684 (5.7)
	500	70 (2.7)	41 (1.4)	176 (6.6)	132 (3.9)	528 (20.1)	459 (14.7)
500	50	86 (7.1)	44 (2.4)	309 (24.2)	170 (9.1)	887 (66.9)	739 (40.1)
	100	64 (10.3)	42 (4.3)	187 (27.8)	144 (15.2)	647 (92.5)	593 (63.9)
	500	83 (56.1)	38 (186)	192 (126.9)	121 (66.5)	522 (342.2)	418 (231.4)

**Table 4** Numerical results on solving (1.4)–(1.5) with distance merit  $\sigma_{\mathcal{F}}(\cdot) = \|\cdot\|_{\infty}$

$n$	$m$	Tol = $10^{-8}$		Tol = $10^{-9}$		Tol = $10^{-10}$	
		ADMM	PDHG	ADMM	PDHG	ADMM	PDHG
50	500	95 (2.1)	77 (1.4)	227 (4.7)	201 (3.6)	952 (18.7)	762 (13.7)
	800	89 (2.9)	74 (2.1)	215 (6.7)	177 (4.8)	942 (28.3)	672 (18.3)
	1000	81 (3.2)	69 (2.2)	211 (8.0)	176 (5.8)	812 (30.6)	613 (20.2)
100	500	54 (2.3)	33 (1.1)	160 (6.5)	106 (3.8)	544 (22.1)	415 (14.9)
	800	47 (3.1)	34 (2.1)	139 (9.1)	101 (5.4)	528 (34.2)	447 (22.5)
	1000	50 (4.2)	36 (2.3)	129 (10.6)	100 (6.5)	469 (37.4)	414 (27.2)
500	500	30 (20.4)	27 (13.1)	68 (47.4)	60 (28.6)	167 (135.1)	113 (57.5)
	800	32 (36.2)	27 (20.8)	63 (71.4)	60 (51.3)	168 (197.5)	125 (100.6)
	1000	37 (56.2)	27 (29.1)	68 (86.2)	61 (64.5)	160 (241.1)	118 (131.5)

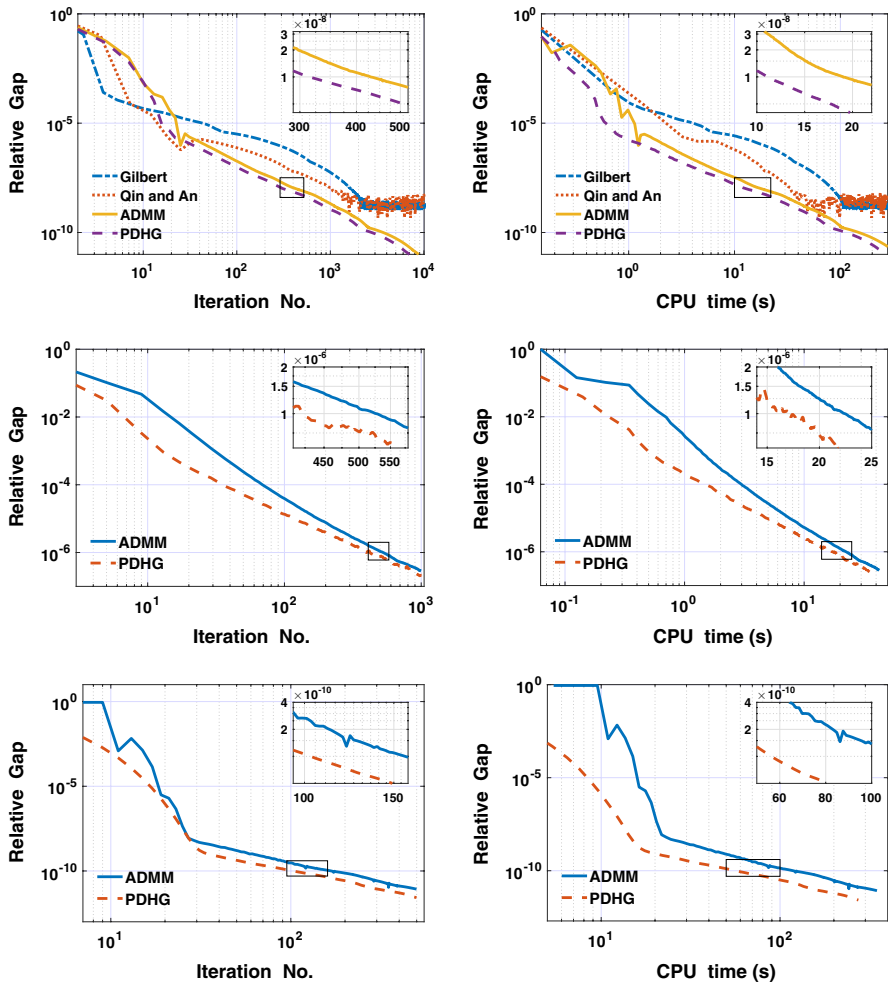


Fig. 5 Evolutions of primal-dual gap values w.r.t. iterations and computing time for solving (1.4). Top row: the case of  $\sigma_{\mathcal{F}}(x) = \|x\|_2$  with  $(n, m) = (100, 500)$ . Center row: the case of  $\sigma_{\mathcal{F}}(x) = \|x\|_1$  with  $(n, m) = (100, 500)$ . Bottom row: the case of  $\sigma_{\mathcal{F}}(x) = \|x\|_\infty$  with  $(n, m) = (500, 500)$

pre-determined tolerances. Note that the number of iterations and computing time reported in Tables 2, 3 and 4 are averaged by running 50 i.i.d random datasets. Table 2 shows that: (i) the performances of ADMM and PDHG outperform substantially those of Gilbert’s method, Qin and An’s method; (ii) ADMM is competitive to PDHG when the dimensionality (i.e., the  $n$ ) and the number of component sets (i.e., the  $m$ ) in problem (1.4) are relative small. The superiority of PDHG would be more evident when the dimensionality or the number of component sets in (1.4) increases. The number of iterations between ADMM and PDHG differs slightly, whilst the computing time of both methods is quite disparate. The reason is that ADMM recursion involves solving generic linear systems which are usually time-consuming in

high-dimensional scenarios, whilst the recursion of PDHG merely involves matrix-vector multiplications whose arithmetic operations are relatively low. Both ADMM and PDHG are also tested for solving problem (1.4) with nonsmooth distance merits, e.g.,  $\sigma_{\mathcal{F}}(x) = \|x\|_1$  and  $\|x\|_{\infty}$ . The numerical results are reported in Tables 3 and 4, which demonstrate the compelling performances of PDHG. In Fig. 5, we plot the evolutions of primal-dual relative gap with respect to the number of iterations and computing time in seconds for some test datasets with variant support function  $\sigma_{\mathcal{F}}$ . The figures show that both ADMM and PDHG perform well for solving the problem (1.4). PDHG is more numerically preferable for solving problem (1.4) in large-scale.

## 6 Concluding remarks

In this paper, we explore the distance between convex sets with Minkowski sum structure (or equivalently, the projection onto Minkowski sum of convex sets), which is a fundamental nomenclature in computational geometry. By reformulating the problem into a separable convex optimization with linear constraints, we cope with the resulted model by some recent advanced optimization methods. Attributed to the skilfully first-order optimization methods, the distance can be efficiently solved in primal, dual, or primal-dual forms. Moreover, those methods are also favourable for solving large-scale computational geometry problems by aid of parallel computing techniques.

**Acknowledgements** The authors would like to thank the anonymous referees for their valuable comments, which helped improving the paper substantially. X. Wang was supported by NSFC 11871279 and STCSM 19ZR1414200. W. Zhang was supported by NSFC 11971003.

## References

1. Aliyu, M.D.: A vertex algorithm for collision detection. *Eur. J. Oper. Res.* **120**, 174–180 (2000)
2. Bauschke, H.H., Combettes, P.L.: *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, New York (2011)
3. Bertsekas, D.P.: *Convex Analysis and Optimization*. Athena Scientific, Belmont (2003)
4. Bertsekas, D.P., Tsitsiklis, J.N.: *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Upper Saddle River (1989)
5. Böröczky, K., Reitzner, M.: Approximation of smooth convex bodies by random circumscribed polytopes. *Ann. Appl. Probab.* **14**, 239–273 (2004)
6. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**, 1–122 (2011)
7. Bronstein, E.M.: Approximation of convex sets by polytopes. *J. Math. Sci.* **153**, 727–762 (2008)
8. Cameron, S.: Enhancing GJK: computing minimum and penetration distances between convex polyhedra. In: *IEEE International Conference on Robotics and Automation*, pp. 3112–3117 (2002)
9. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**, 120–145 (2012)
10. Chambolle, A., Ehrhardt, M.J., Richtárik, P., Schönlieb, C.: Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM J. Optim.* **28**, 2783–2808 (2018)
11. Combettes, P.L., Pesquet, J.C.: Proximal splitting methods in signal processing. In: Bauschke, H.H., et al. (eds.) *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pp. 185–212. Springer, Berlin (2011)



12. Condat, L.: Fast projection onto the simplex and the  $\ell_1$  ball. *Math. Program.* **158**, 575–585 (2016)
13. Dax, A.: The distance between two convex sets. *Linear Algebra Appl.* **416**, 184–213 (2006)
14. Eckstein, J.: Augmented Lagrangian and alternating direction methods for convex optimization: a tutorial and some illustrative computational results. RUTCOR Research Report, (2012)
15. Eckstein, J.: Splitting methods for monotone operators with applications to parallel optimization. Ph.D. Thesis, MIT (1989)
16. Esser, E., Zhang, X.Q., Chan, T.F.: A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imaging Sci.* **3**, 1015–1046 (2010)
17. Fogel, E., Halperin, D., Wein, R.: *CGAL Arrangements and Their Applications*. Springer, Berlin (2012)
18. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximations. *Comput. Math. Appl.* **2**, 17–40 (1976)
19. Gilbert, E.: An iterative procedure for computing the minimum of a quadratic form on a convex set. *SIAM J. Control* **6**, 61–80 (1966)
20. Gilbert, E., Foo, C.: Computing the distance between general convex objects in three-dimensional space. *IEEE Trans. Robot. Autom.* **6**, 53–61 (1990)
21. Gilbert, E., Johnson, D., Keerthi, S.: A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Trans. Robot. Autom.* **4**, 193–203 (1988)
22. Glowinski, R.: On alternating direction methods of multipliers: a historical perspective. *Model. Simul. Optim. Sci. Technol. Comput. Methods Appl. Sci.* **34**, 59–82 (2014)
23. Glowinski, R., Marrocco, A.: Sur l’approximation par éléments finis d’ordre un et la résolution par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires. *Revue Fr. Autom. Inform. Rech. Opér., Anal. Numér.* **2**, 41–76 (1975)
24. He, B.S., Yuan, X.M.: Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. *SIAM J. Imaging Sci.* **5**, 119–149 (2012)
25. He, B.S., Yuan, X.M.: On the  $O(1/n)$  convergence rate of the Douglas–Rachford alternating direction method. *SIAM J. Numer. Anal.* **50**, 700–709 (2012)
26. He, B.S., You, Y.F., Yuan, X.M.: On the convergence of primal-dual hybrid gradient algorithm. *SIAM J. Imaging Sci.* **7**, 2526–2537 (2014)
27. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration space. *IEEE Trans. Robot. Autom.* **12**, 566–580 (1996)
28. Keil, J.M.: Decomposing a polygon into simpler components. *SIAM J. Comput.* **14**, 799–817 (1985)
29. Latombe, J.C.: *Robot Motion Planning*. Kluwer Academic Publishers, Boston (1991)
30. Lewis, A.S., Malick, J.: Alternating projections on manifolds. *Math. Oper. Res.* **33**, 216–234 (2008)
31. Lin, M., Canny, J.: A fast algorithm for incremental distance calculation. In: *IEEE International Conference on Robotics and Automation*, pp. 266–275 (1991)
32. Liu, Z., Fathi, Y.: An active index algorithm for the nearest point problem in a polyhedral cone. *Comput. Optim. Appl.* **49**, 435–456 (2011)
33. Mayer, A., Zelenyuk, V.: Aggregation of Malmquist productivity indexes allowing for reallocation of resources. *Eur. J. Oper. Res.* **238**, 774–785 (2014)
34. Mitchell, B.F., Dem’Yanov, V.F., Malozemov, V.N.: Finding the point of a polyhedron closest to the origin. *SIAM J. Control* **12**, 19–26 (1974)
35. Németh, A., Németh, S.: How to project onto an isotone projection cone. *Linear Algebra Appl.* **433**, 41–51 (2010)
36. Nesterov, Y.: A method for unconstrained convex minimization problem with the rate of convergence  $O(1/k^2)$ . *Dokl. Akad. Nauk SSSR* **269**, 543–547 (1983)
37. Nesterov, Y.: Smooth minimization of non-smooth functions. *Math. Program.* **103**, 127–152 (2005)
38. O’Connor, D., Vandenberghe, L.: On the equivalence of the primal-dual hybrid gradient method and Douglas–Rachford splitting. *Math. Program.* **179**, 85–108 (2020)
39. Ong, C.J., Gilbert, E.: Fast versions of the Gilbert–Johnson–Keerthi distance algorithm: additional results and comparisons. *IEEE Trans. Robot. Autom.* **17**, 531–539 (2001)
40. Qin, X.L., An, N.T.: Smoothing algorithms for computing the projection onto a Minkowski sum of convex sets. *Comput. Optim. Appl.* **74**, 821–850 (2019)
41. Rosin, P.L.: Shape partitioning by convexity. *IEEE Trans. Syst. Man, Cybern. A* **30**, 202–210 (2000)
42. Ryu, E., Boyd, S.: A primer on monotone operator methods. *Appl. Comput. Math.* **15**, 3–43 (2016)
43. Sekitani, K., Yamamoto, Y.: Recursive algorithm for finding the minimum norm point in a polytope and a pair of closest points in two polytopes. *Math. Program.* **61**, 233–249 (1993)
44. Serra, J.: *Image Analysis and Mathematical Morphology*. Academic Press, Cambridge (1983)

45. Smid, P.: CNC Programming Handbook. Industrial Press, New York (2008)
46. Spingarn, J.E.: Applications of the method of partial inverses to convex programming: decomposition. *Math. Program.* **32**, 199–223 (1985)
47. Sra, S.: Fast projections onto  $\ell_{1,q}$ -norm balls for grouped feature selection. In: *Machine Learning and Knowledge Discovery in Databases*, pp. 305–317 (2011)
48. Sussman, G.J., Wisdom, J.: *Structure and Interpretation of Classical Mechanics*. MIT Press, Cambridge (2002)
49. van den Berg, E., Friedlander, M.P.: Probing the Pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.* **31**, 890–912 (2008)
50. Wolfe, P.: Finding the nearest point in a polytope. *Math. Program.* **11**, 128–149 (1976)
51. Zheng, Y., Yamane, K.: Generalized distance between compact convex sets: algorithms and applications. *IEEE Trans. Robot.* **31**, 988–1003 (2015)
52. Zhu, X.Y., Tso, S.K.: A pseudodistance function and its applications. *IEEE Trans. Robot.* **20**, 344–352 (2004)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.