THE CONVEX FEASIBLE SET ALGORITHM FOR REAL TIME OPTIMIZATION IN MOTION PLANNING

CHANGLIU LIU∗, CHUNG-YEN LIN†, AND MASAYOSHI TOMIZUKA†

Abstract. With the development of robotics, there are growing needs for real time motion planning. However, due to obstacles in the environment, the planning problem is highly non-convex, which makes it difficult to achieve real time computation using existing non-convex optimization algorithms. This paper introduces the convex feasible set algorithm (CFS) which is a fast algorithm for non-convex optimization problems that have convex costs and non-convex constraints. The idea is to find a convex feasible set for the original problem and iteratively solve a sequence of subproblems using the convex constraints. The feasibility and the convergence of the proposed algorithm are proved in the paper. The application of this method on motion planning for mobile robots is discussed. The simulations demonstrate the effectiveness of the proposed algorithm.

Key words. Non-convex optimization, non-differentiable optimization, robot motion planning

AMS subject classifications. 90C55, 90C26, 68T40, 93C85

1. Introduction. Although great progresses have been made in robot motion planning [13], the field is still open for research regarding real time planning in dynamic uncertain environment. The applications include but are not limited to real time navigation [8], autonomous driving [12, 16], robot arm manipulation and human robot cooperation [15]. To achieve safety and efficiency, the robot motion should be re-planned from time to time when new information is obtained during operation. This demands that the motion planning algorithm should run fast enough to meet the real time requirement.

In this paper, we focus on optimization-based motion planning methods, which fit into the framework of model-predictive control (MPC) [10], where the optimal trajectory is obtained by solving a constrained optimization at each time step. As there are obstacles in the environment, the constraints are highly non-convex, which makes the problem hard to solve in real time. Various methods have been developed to deal with the non-convexity [20, 25]. One popular way is through convexification [26], e.g. transforming the non-convex problem into a convex one. Some authors tried to transform the non-convex problem to semidefinite programming (SDP) [6]. Some authors proposed to introduce lossless convexification by augmenting the space [2, 9]. And some authors propose successive linear approximation to remove non-convex constraints [18, 19]. However, the first method can only handle quadratic cost functions. The second approach highly depends on the linearity of the system and may not be able to handle the variety of obstacles. And the third approach may not generalize to non-differentiable problems. One of the most famous convexification method is the sequential quadratic programming (SQP) [24, 27], which approximates the non-convex problem as a sequence of quadratic programming (QP) problems and solves them iteratively. The method has been successfully applied to offline robot motion planning [11, 23]. However, as SQP is a generic algorithm, the unique structure of the motion planning problems is neglected, which usually results in failure to meet the real time requirement in engineering applications.

Typically, in a motion planning problem, the constraints are physically deter-
mined, while the objective function is designed to be convex [28, 21]. The non-convexity mainly comes from the physical constraints. Regarding this observation, a fast algorithm called the convex feasible set algorithm (CFS) is proposed in this paper to solve optimization-based motion planning problems with convex objective functions and non-convex constraints.

The main idea of the CFS algorithm is to translate the original problem into a sequence of convex subproblems by finding convex feasible sets within the non-convex constraints, then iteratively solve the convex subproblems until convergence. The idea is similar to SQP in that it tries to solve several convex subproblems iteratively. However, the difference lies in the methods to obtain the convex subproblems. The geometric structure of the original problem is fully considered in CFS. This strategy will make the computation faster than conventional SQP and other non-convex optimization methods such as interior point (ITP) [29], as will be demonstrated later. Moreover, it is guaranteed that local optima will be attained.

It is worth noting that the convex feasible set in the trajectory space can be regarded as a convex corridor. The idea of using convex corridors to simplify the motion planning problems has been discussed in [30, 4]. However, these methods are application-specific without theoretical guarantees. In this paper, we consider general non-convex and non-differentiable optimization problems (which may not only arise from motion planning problems but other problems) and provide theoretical guarantees of the method.

The remainder of the paper will be organized as: in section 2, a benchmark optimization problem is proposed; the proposed algorithm in solving the benchmark problem is discussed in section 3; section 4 shows the feasibility and convergence of the algorithm; section 5 discusses the application of the algorithm on motion planning problems for mobile robots; and section 6 concludes the paper.

2. The Optimization Problem.

2.1. The Benchmark Problem. Consider an optimization problem with a convex cost function but non-convex constraints, i.e.

\[
\min_{x \in \Gamma} J(x),
\]

where \( x \in \mathbb{R}^n \) is the decision variable and the problem follows two assumptions.

**Assumption 1 (Cost).** \( J : \mathbb{R}^n \to \mathbb{R}^+ \) is smooth, strictly convex.

**Assumption 2 (Constraint).** The set \( \Gamma \subset \mathbb{R}^n \) is connected and complete, while its complements \( \Gamma^c := \mathbb{R}^n \setminus \Gamma \) is a disjoint collection of simply-connected open sets with piecewise smooth boundaries and disjoint closures. For every point \( x \) on the boundary \( \partial \Gamma \), there exists an \( n \)-dimensional convex polytope \( P \subset \Gamma \) such that \( x \in P \).

Assumption 1 implies that \( J \) is radially unbounded, i.e. \( J(x) \to \infty \) when \( \|x\| \to \infty \). Assumption 2 specifies the geometric features of the feasible set \( \Gamma \), where the first part deals with the topological features of \( \Gamma \) and \( \Gamma^c \) and the second part ensures that the constraint can be locally convexified. Intuitively, Assumption 2 excludes (1) local and global equality constraints\(^1\), (2) sharp concave corners\(^2\) in \( \Gamma^c \) and (3) intersecting constraints.

---

\(^1\)We say a point \( x \in \Gamma \) in on a local equality constraint when the dimension of the neighborhood of \( x \) is less than \( n \). If \( \Gamma \) is on a \( m \)-dimensional manifold with \( m < n \), we then call it a global equality constraint.

\(^2\)A sharp concave corner is a concave corner such that some pieces of boundary are tangent to each other locally.
boundaries, which are illustrated in Figure 1. In the figure, $\Gamma$ includes the white area and the black boundaries. When there is an equality constraint at $x \in \partial \Gamma$, e.g. $x_4$ in Figure 1, we cannot find an $n$-dimensional convex polytope $P \subset \Gamma$ with $x \in P$ since the dimension of $\Gamma$ is less than $n$ locally. When there are sharp concave corners in the infeasible set, e.g. $x_1$ in Figure 1, the dimension of the convex feasible polytope is strictly less than $n$. On the other hand, the desired polytopes (shown in red) can be defined at non-sharp concave corners such as $x_2$, convex corners such as $x_3$ and smooth boundary points such as $x_5$. When the boundary intersects with itself such as in $x_6$, it is possible to find such polytopes. But the 8-shaped set consists of two disjoint simply-connected open sets whose closures are no longer disjoint, which violates the first part of Assumption 2. Hence intersecting boundaries are not considered in this paper. We will relax this assumption in our future work.

The geometric structure of problem (1) is illustrated in Figure 2a. The contour represents the cost function $J$, while the gray parts represent $\Gamma^c$. There are two disjoint components in $\Gamma^c$. The goal is to find a local optimum (hopefully global optimum) starting from the initial reference point (blue dot). As shown in Figure 2a, the problem is highly non-convex and the non-convexity mainly comes from the constraint. To make the computation more efficient, we propose the convex feasible set algorithm in this paper, which transforms the problem into a sequence of convex optimizations by obtaining a sequence of convex feasible sets inside the non-convex domain $\Gamma$. As shown in Figure 2, the idea is implemented iteratively. At current iteration, a convex feasible set for the current reference point (blue dot) is obtained. The optimal solution in the convex feasible set (black dot) is set as the reference point for the next iteration. The formal mathematical description of this algorithm will be discussed in section 3. The feasibility of this method, i.e. the existence of an $n$-dimensional convex feasible set, is implied by Assumption 2. Nonetheless, in order to compute the convex feasible set efficiently, we still need an analytical description of the constraint, which will be discussed in subsection 2.2.

2.2. Analytical Representation of the Constraints. Analytically, $\Gamma$ can be represented by $N$ continuous and piecewise smooth functions $\phi_i : \mathbb{R}^n \rightarrow \mathbb{R}$, e.g.

$$\Gamma = \bigcap_i \{x : \phi_i(x) \geq 0\} = \bigcap_i \Gamma_i,$$

where $\Gamma_i := \{x : \phi_i(x) \geq 0\}$. $\phi_i$ can be regarded as a potential function of $\Gamma_i$ where the boundary and the interior of $\Gamma_i$ satisfy that $\partial \Gamma_i = \{x : \phi_i(x) = 0\}$ and $\Gamma^c_i = \{x : \phi_i(x) > 0\}$. Note that $\Gamma^c_i$’s are not required to be disjoint and $N$ should be greater than or equal to the number of disjoint components in $\Gamma^c$. The decomposition
from $\Gamma$ to $\Gamma_i$'s is not unique. Neither is the function $\phi_i$ that represent $\Gamma_i$. In many cases, $\phi_i$ can be chosen as a signed distance function to $\partial \Gamma_i$, which will be discussed in subsection 5.2. Since $\phi_i$ is only piecewise smooth, it may not be differentiable at all points. For any $v \in \mathbb{R}^n$, define the directional derivative $\partial_v$ as

$$
\partial_v \phi_i(x) := \lim_{a \to 0^+} \frac{\phi_i(x + av) - \phi_i(x)}{a}.
$$

Let $S(\phi_i, x) := \{v \in \mathbb{R}^n : \partial_v \phi_i(x) + \partial_{-v} \phi_i(x) = 0\}$ denote all the smooth directions of function $\phi$ at point $x$. Define the sub-differential of $\phi_i$ at $x$ as

$$
D\phi_i(x) := \{d \in \mathbb{R}^n : d \cdot v \leq \partial_v \phi_i(x), \forall v \in \mathbb{R}^n\}.
$$

This definition is valid, i.e. the right hand side of (4) is non empty under Assumption 3 below and will be justified later. The elements in $D\phi_i(x)$ are called sub-gradients. When $\phi_i$ is smooth at $x$, $D\phi_i(x)$ reduces to a singleton set which contains only the gradient $\nabla \phi_i(x)$. The definition (4) follows from Clarke (generalized) sub-gradients for non-convex functions [5]. Nonetheless, we make the following assumption on $\phi_i$'s.

**Assumption 3 (Regularity).** (1) $\phi_i$ is semi-convex for all $i$, i.e. there exists a positive semi-definite $H_i^* \in \mathbb{R}^{n \times n}$ such that for any $x, v \in \mathbb{R}^n$,

$$
\phi_i(x + v) - 2\phi_i(x) + \phi_i(x - v) \geq -v^T H_i^* v.
$$

(2) $0 \notin D\phi_i(x)$ if $x \in \partial \Gamma_i$ or if $D\phi_i(x)$ is a singleton set. (3) for any $x$ such that $I := \{i : \phi_i(x) = 0\} \neq \emptyset$, there exists $v \in \mathbb{R}^n$ such that $\partial_v \phi_i(x) < 0$ for all $i \in I$.

**Lemma 1 (Properties of Semi-Convex Functions).** If $\phi_i$ is semi-convex, for any $x, v, v_1, v_2 \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that $v = v_1 + v_2$, the following inequalities hold,

$$
\begin{aligned}
0 &\leq \partial_v \phi_i(x) + \partial_{-v} \phi_i(x),
\end{aligned}
$$

$$
\begin{aligned}
bd_v \phi_i(x) &\leq \partial_{b v} \phi_i(x),
\end{aligned}
$$

$$
\begin{aligned}
\partial_{v_1} \phi_i(x) &\leq \partial_{v_1} \phi_i(x) + \partial_{v_2} \phi_i(x).
\end{aligned}
$$

The equalities in (6) and (7) are achieved when $\phi_i(x)$ is smooth at direction $v$. The equality in (8) is achieved when $\phi_i(x)$ is smooth at directions $v_1$ and $v_2$. Moreover, for any unit vector $w \in \mathbb{R}^n$, $\partial_w \phi_i(x)$ is locally bounded.

---

3Note that $\partial$ means boundary when followed by a set, e.g. $\partial \Gamma$. It means derivative when followed by a function, e.g. $\partial_v \phi_i$. 

---

Fig. 2: Geometry of problem (1) and the idea of the convex feasible set algorithm.
Proof. If \( \phi_i \) is semi-convex, \( (5) \) implies that for any scalar \( a \) and vector \( v \),
\[
\frac{\phi_i(x + av) - 2\phi_i(x) + \phi_i(x - av)}{a} \geq -av^TH_i^*v.
\]
Let \( a \to 0^+ \). The left hand side approaches \( \partial_v \phi_i(x) + \partial_{-v} \phi_i(x) \), while the right hand side approaches 0 in the limits. Hence \( (6) \) holds. By definition in \( (3) \), \( \partial_{a} \phi_i(x) = b\partial_{v_i} \phi_i(x) \) when \( b \geq 0 \). When \( b < 0 \), by \( (6) \), \(-|b|\partial_{v_i} \phi_i(x) \leq |b|\partial_{-v_i} \phi_i(x) = \partial_{b} \phi_i(x) \). Hence \( (7) \) holds. The equality holds when \( \partial_v \phi_i(x) + \partial_{-v} \phi_i(x) = 0 \), i.e. \( \phi_i(x) \) is smooth at direction \( v \). Moreover, \( (5) \) also implies that
\[
- \frac{a^2}{4} (v_1 - v_2)^TH_i^*(v_1 - v_2) \leq \phi_i(x + av_1) - 2\phi_i(x + \frac{v}{2}) + \phi_i(x + av_2)
\]
\[
= \phi_i(x + av_1) - \phi_i(x) + \phi_i(x + av_2) - \phi_i(x) - 2[\phi_i(x + \frac{v}{2}) - \phi_i(x)].
\]
Divide the both sides by \( a \) and take \( a \to 0^+ \). Then the left hand side approaches 0, while the right hand side approaches \( \partial_{v_i} \phi_i(x) + \partial_{-v_i} \phi_i(x) \). Hence \( (8) \) holds. When \( \phi_i(x) \) is smooth at directions \( v_1 \) and \( v_2 \),
\[
0 \leq \partial_v \phi_i(x) + \partial_{-v} \phi_i(x) \leq \partial_{v_1} \phi_i(x) + \partial_{v_2} \phi_i(x) + \partial_{-v_1} \phi_i(x) + \partial_{-v_2} \phi_i(x) = 0.
\]
The first inequality is due to \( (6) \); the second inequality is due to \( (8) \). Hence the equality in \( (8) \) is attained and \( \phi_i(x) \) is also smooth at \( v \).

Since \( \phi_i \) is semi-convex, for any \( a > 0 \), \( \phi_i(x + aw) \geq \phi_i(x) + a\partial_a \phi_i(x) - \frac{a^2}{2}w^TH_i^*w \). In a neighborhood of \( x \), \( \phi_i(x + aw) \) is bounded, which implies that \( \partial_{a} \phi_i(x) < \infty \). Moreover, by \( (6) \), \( \partial_{a} \phi_i(x) \geq -\partial_{-a} \phi_i(x) > -\infty \). Hence \( \partial_{a} \phi_i(x) \) is bounded locally. \( \square \)

Lemma 1 justifies the definition in \( (4) \). By \( (7) \) and \( (8) \), for any \( v_1, v_2 \in S(\phi_i, x) \), \( a, b \in \mathbb{R} \) and \( v = av_1 + bv_2 \), \( \partial_{v} \phi_i(x) = a\partial_{v_1} \phi_i(x) + b\partial_{v_2} \phi_i(x) \) and \( \partial_{a} \phi_i(x) + \partial_{-a} \phi_i(x) = 0 \). Hence \( v \in S(\phi_i, x) \). We can conclude that \( (1) \) \( S(\phi_i, x) \) is a linear subspace of \( \mathbb{R}^n \) and \( (2) \) the function induced by the directional derivative \( v \mapsto \partial_{v} \phi_i(x) \) is a sub-linear function\(^4\) on \( \mathbb{R}^n \) and a linear function on \( S(\phi_i, x) \). By Hahn-Banach Theorem \([7]\), there exists a vector \( d \in \mathbb{R}^n \) such that \( d \cdot v = \partial_{v} \phi_i(x) \) for \( v \in S(\phi_i, x) \) and \( d \cdot v \leq \partial_{a} \phi_i(x) \) for \( v \in \mathbb{R}^n \). Moreover, as the directional derivative is bounded, the sub-gradients are also bounded. Hence the definition in \( (4) \) is justified.

Geometrically, Assumption 3 implies that there cannot be any concave corners in \( \Gamma^c \) or convex corners in \( \Gamma_i \). Suppose \( \Gamma^c_i \) has a concave corner at \( x \in \partial \Gamma_i \). Since \( 0 \notin \partial \phi_i(x) \), we can choose a unit vector \( v \) such that \( \partial_v \phi_i < 0 \) and \( \partial_{-v} \phi_i < 0 \) as shown in Figure 3a. Then \( (6) \) is violated, which contradicts with the assumption on semi-convexity. Nonetheless, concave corners are allowed in \( \Gamma^c \), but should only be formulated by a combination of several intersecting \( \Gamma^c_i \)'s as shown in Figure 3. Those \( \Gamma^c_i \)'s should be intersecting due to the third statement of Assumption 3. In the example, the set \( \Gamma = \{ x = (x_1, x_2) : \min(|x_1| - 1, |x_2| - 1) \geq 0 \} \) is partitioned into two sets \( \Gamma_1 = \{ x : |x_1| - 1 \geq 0 \} \) and \( \Gamma_2 = \{ x : |x_2| - 1 \geq 0 \} \). Both \( \phi_1 = |x_1| - 1 \) and \( \phi_2 = |x_2| - 1 \) satisfies Assumption 3. Without the partition, \( \phi = \min(|x_1| - 1, |x_2| - 1) \) violates the condition on semi-convexity\(^5\). A method to partition the obstacles and construct the desired \( \phi_i \)'s is discussed in \([14]\). It is our hypothesis that any set \( \Gamma \)

\(^4\)A function \( f \) is called sub-linear if it satisfies positive homogeneity \( f(ax) = af(x) \) for \( a > 0 \), and sub-additivity \( f(x + y) \leq f(x) + f(y) \).

\(^5\)Let \( x = (1, 1) \) and \( v = (\cos \frac{\pi}{4}, \sin \frac{\pi}{4}) \). Then \( \phi_1(x + av) - 2\phi_1(x) + \phi_1(x - av) = -a \cos \frac{\pi}{4} - a \sin \frac{\pi}{4} = -\sqrt{2}a \), which can not be greater than any \( -a^2v^TH_i^*v \) when \( a \) is small.
that satisfies Assumption 2 can be partitioned into $\Gamma_i$’s and represented by $\phi_i$’s that satisfy Assumption 3, which will be verified in our future work.

Assumption 3 will be exploited in computing the convex feasible set in subsection 3.2. Lemma 1 will be used in the proofs in section 4.

2.3. Physical Interpretations. Many motion planning problems can be formulated into (1) when $x$ is regarded as the trajectory as will be discussed in section 5. The dimension of the problem $n$ is proportional to the number of sampling points on the trajectory. If continuous trajectories are considered, then $n \to \infty$ and $\mathbb{R}^n$ approaches the space of continuous functions $C(\mathbb{R})$ in the limit.

In addition to motion planning problems, the proposed method deals with any problem with similar geometric properties as specified in Assumption 1 and Assumption 2. Moreover, problems with global linear equality constraints also fit into the framework if we solve the problem in the low-dimensional linear manifold defined by the linear equality constraints. The case for nonlinear equality constraints is much trickier since convexification on nonlinear manifold is difficult in general. A method to deal with nonlinear equality constraints is discussed in [17].

3. Solving the Optimization Problem.

3.1. The Convex Feasible Set Algorithm. To solve the problem (1) efficiently, we propose the convex feasible set algorithm. As introduced in subsection 2.1, a convex feasible set $\mathcal{F}$ for the set $\Gamma$ is a convex set such that $\mathcal{F} \subset \Gamma$. $\mathcal{F}$ is not unique. We define the desired $\mathcal{F}$ in subsection 3.2. As $\Gamma$ can be covered by several (may be infinitely many) convex feasible sets, we can efficiently search the non-convex space $\Gamma$ for solutions by solving a sequence of convex optimizations constrained in a sequence of convex feasible sets. The idea is implemented iteratively as shown in Figure 2a. At iteration $k$, given a reference point $x^{(k)}$, a convex feasible set $\mathcal{F}^{(k)} := \mathcal{F}(x^{(k)}) \subset \Gamma$ is computed around $x^{(k)}$. Then a new reference point $x^{(k+1)}$ will be obtained by solving the resulting convex optimization problem

$$x^{(k+1)} = \arg \min_{x \in \mathcal{F}^{(k)}} J(x).$$

The optimal solution will be used as the reference point for the next step. The iteration will terminate if either the change in solution is small, e.g.

$$\|x^{(k+1)} - x^{(k)}\| \leq \epsilon_1$$

for some small $\epsilon_1 > 0$, or the descent in cost is small, e.g.

$$J(x^{(k)}) - J(x^{(k+1)}) \leq \epsilon_2$$

for some small $\epsilon_2 > 0$. We will show in section 4 that these two conditions are equivalent and both of them imply convergence. The process is summarized in Algorithm 1.
**Algorithm 1 The Convex Feasible Set Algorithm**

Initialize initial guess $x^{(0)}$, $k := 0$;

while True do

Find a convex feasible set $\mathcal{F}^{(k)} \subset \Gamma$ for $x^{(k)}$;

Solve the convex optimization problem (10) for $x^{(k+1)}$;

if (11) or (12) is satisfied then

Break the while loop;

$k := k + 1$;

end while

return $x^{(k+1)}$;

---

### 3.2. Finding the Convex Feasible Set

Since the constraint $\Gamma$ can be represented by several piecewise smooth functions (2), we try to find a convex feasible set $\mathcal{F}_i$ for each constraint $\Gamma_i = \{x : \phi_i(x) \geq 0\}$.

**Case 1: $\phi_i$ is concave.** Then $\Gamma_i$ is convex. The convex feasible set is chosen to be itself,

$$\mathcal{F}_i = \Gamma_i.$$  

**Case 2: $\phi_i$ is convex.** Then $\Gamma_i$ is convex. The convex feasible set $\mathcal{F}_i$ with respect to a reference point $x^r \in \mathbb{R}^n$ is defined as

$$\mathcal{F}_i(x^r) := \{x : \phi_i(x^r) + \hat{\nabla} \phi_i(x^r)(x - x^r) \geq 0\},$$

where $\hat{\nabla} \phi_i(x^r) \in D\phi_i(x^r)$ is a sub-gradient. When $\phi_i$ is smooth at $x^r$, $\hat{\nabla} \phi_i(x^r)$ equals to the gradient $\nabla \phi_i(x^r)$. Otherwise, the sub-gradient is chosen according to the method discussed in subsection 3.3. Since $\phi_i$ is convex, $\phi_i(x) \geq \phi_i(x^r) + \partial_{x-x^r} \phi_i(x) \geq \phi_i(x^r) + d \cdot (x - x^r)$ for all $d \in D\phi_i(x^r)$. Hence $\mathcal{F}_i(x^r) \subset \{x : \phi_i(x) \geq 0\} = \Gamma_i$ for all $x^r \in \mathbb{R}^n$.

**Case 3: $\phi_i$ is neither concave nor convex.** Considering Assumption 3, the convex feasible set with respect to the reference point $x^r$ is defined as

$$\mathcal{F}_i(x^r) := \{x : \phi_i(x^r) + \hat{\nabla} \phi_i(x^r)(x - x^r) \geq \frac{1}{2} (x - x^r)^T H_i^*(x - x^r)\},$$

where $\hat{\nabla} \phi_i(x^r) \in D\phi_i(x^r)$, which is chosen according to the method discussed in subsection 3.3. Since $\phi_i$ is semi-convex, $\phi_i(x) \geq \phi_i(x^r) + \partial_{x-x^r} \phi_i(x) - \frac{1}{2} (x - x^r)^T H_i^*(x - x^r) \geq \phi_i(x^r) + d \cdot (x - x^r) - \frac{1}{2} (x - x^r)^T H_i^*(x - x^r)$ for all $d \in D\phi_i(x^r)$. Hence $\mathcal{F}_i(x^r) \subset \{x : \phi_i(x) \geq 0\} = \Gamma_i$ for all $x^r \in \mathbb{R}^n$.

Considering (13), (14) and (15), the convex feasible set for $\Gamma$ at $x^r$ is defined as

$$\mathcal{F}(x^r) := \bigcap_i \mathcal{F}_i(x^r).$$

### 3.3. Choosing the Optimal Sub-Gradients

The sub-gradients in (14) and (15) should be chosen such that the steepest descent of $J$ in the set $\Gamma$ is always included in the convex feasible set $\mathcal{F}$.

Let $B(x, r)$ denote the unit ball centered at $x \in \mathbb{R}^n$ with radius $r$. At point $x^r$, a search direction $v \in \partial B(0, 1)$ is feasible if for all $i$, one of the three conditions holds:
Case two

C. LIU, C. LIN AND M. TOMIZUKA

and so on is chosen as: choosing the one with the smallest first entry, the smallest second entry, can choose $v$ is exploited in Proposition 3.

The feasibility and convergence of Algorithm 1 will be shown. The main result is summarized in the following Theorem:

Define the set of feasible search directions as $C(x^*)$, which is non empty since we can choose $v$ to be $d/\|d\|$ for any nonzero $d \in D\phi$, $D\phi$ always contain a nonzero element by the second statement in Assumption 2. Then the direction of the steepest descent is $v^* := \arg \min_{v \in C(x^*)} \nabla J \cdot v$. If $v^*$ is not unique, the tie breaking mechanism is chosen as: choosing the one with the smallest first entry, the smallest second entry, and so on. Then the optimal sub-gradient is chosen to be $\hat{\nabla} \phi_i := \arg \min_{d \in DF_i} \nabla J \cdot d/\|d\|$, where $DF_i$ is the feasible set of sub-gradients for $\phi_i$ such that $DF_i := D\phi_i$ when $\phi_i > 0$; $DF_i := \{d \in D\phi_i | d \cdot v^* \geq 0\}$ when $\phi_i = 0$; and $DF_i := \{d \in D\phi_i | d \cdot v^* > 0\}$ when $\phi_i < 0$. The set $DF_i$ is non empty by definition of $C(x^*)$. To avoid singularity, let $\|d\| = 1$ when $d = 0$. Figure 4 illustrates the above procedure in choosing the optimal sub-gradient, where the short arrow shows the direction of the steepest descent of $J$, the shaded sector shows the range of sub-differentials, the long arrow denotes the optimal sub-gradient and the shaded half-space is the convex feasible set $F_i$. In case one, $v^*$ is in the same direction of $\hat{\nabla} \phi_i$, while the two are perpendicular to each other in case two.

4. Properties of the Convex Feasible Set Algorithm. In this section, the feasibility and convergence of Algorithm 1 will be shown. The main result is summarized in the following Theorem:

Theorem 1 (Convergence of Algorithm 1). Under Algorithm 1, the sequence $\{x^{(k)}\}$ will converge to some $x^* \in \Gamma$ for any initial guess $x^{(0)}$ such that $F^{(0)} \neq \emptyset$. When the limit is attained, e.g. there exists a constant $K \in \mathbb{N}$ such that $x^{(k)} = x^*$ for all $k > K$, $x^*$ is a strong local optimum of (1). Otherwise, the sequence $\{J(x^{(k)})\}$ is strictly decreasing and $x^*$ is at least a weak local optimum of (1).

We say that $x^*$ is a strong local optimum of (1) if $J$ is nondecreasing along any feasible search direction, e.g. $\nabla J(x^*) v \geq 0$ for all $v \in C(x^*)$, as shown in Figure 5a. We say that $x^*$ is a weak local optimum of (1) if the KKT condition is satisfied, i.e. $\nabla J(x^*) + \sum_{i=1}^{N} \lambda_i d_i = 0$ for some $d_i \in D\phi_i(x^*)$, as shown in Figure 5b. $\lambda_i$ is a Lagrange multiplier such that $\lambda_i \leq 0$ and $\lambda_i \phi_i(x) = 0$ (complementary slackness) for all $i = 1, \cdots, N$. A strong local optimum is always a weak local optimum. The two are equivalent when all $\phi_i$'s are smooth at $x^*$.

Note that the tie braking mechanism can be any as long as it makes $v^*$ unique. The uniqueness is exploited in Proposition 3.
4.1. Preliminary Results. Before proving Theorem 1, we present some preliminary results that are useful towards proving the theorem. Lemma 2 states that given a feasible reference point \( x^* \), \( F(x^*) \) is a convex set containing \( x^* \) with nontrivial interior. This conclusion naturally leads to the hypothesis that a suboptimal reference \( x^* \) can be improved by optimizing \( J \) in the convex feasible set \( \mathcal{F}(x^*) \). We will show in Proposition 3 that if the solution can not be improved using Algorithm 1, then \( x^* \) is already a strong local optimum of (1). Otherwise, if we can keep improving the result using Algorithm 1, this process will generate a Cauchy sequence that converges to a weak local optimum of (1), which will be shown in Proposition 5. Given these results, the conclusion in the theorem follows naturally.

We say that a reference point \( x^* \in \mathbb{R}^n \) is feasible if \( x^* \in \Gamma \); and \( x^* \in \Gamma \) is a fixed point of Algorithm 1 if

\[
(17) \quad x^* = \arg \min_{x \in \mathcal{F}(x^*)} J(x).
\]

**Lemma 2 (Feasibility).** If \( x^* \in \Gamma \), then \( x^* \in \mathcal{F}(x^*) \) and \( \mathcal{F}(x^*) \neq \emptyset \).

**Proof.** When \( x^* \) is feasible, \( x^* \in \mathcal{F}_i(x^*) \) for all \( i \) according to the definitions in (13), (14) and (15). Hence \( x^* \in \mathcal{F}(x^*) \).

Claim 1: if \( x^* \in \Gamma^0 \), then \( x^* \in \mathcal{F}(x^*) \). The condition \( x^* \in \Gamma^0 \) implies that \( \phi_i(x^*) > 0 \) for all \( i = 1, \ldots, N \). Then the inequality in (14) and (15) are not tight at \( x^* \). Hence \( x^* \in \mathcal{F}_i(x^*) \) in either of the three cases. Since \( N \) is finite, \( x^* \in \bigcap_{i=1}^{N} \mathcal{F}_i(x^*) = \mathcal{F}(x^*) \).

Claim 2: if \( x^* \in \partial \Gamma \), there exists a non trivial \( v \in \mathbb{R}^n \) such that \( x^* + v \in \mathcal{F}(x^*) \).

Let \( I := \{ i : \phi_i(x) = 0 \} \). By the third statement in Assumption 3, there exists a unit vector \( w \in \mathbb{R}^n \) such that \( \partial_w \phi_i(x^*) < 0 \) for all \( i \in I \). Fix \( i \in I \). Let \( a > 0 \) be sufficiently small. When \( \phi_i \) is concave, \( \phi_i(x^* - aw) \approx \partial_{-aw} \phi_i(x^*) \geq -a \partial_w \phi_i(x^*) > 0 \) where the first inequality is due to (6). When \( \phi_i \) is convex, \( \phi_i(x^*) + \nabla \phi_i(x^*) \cdot (-aw) \geq -a \partial_w \phi_i(x^*) > 0 \) where the first inequality is due to \( \nabla \phi_i(x^*) \cdot w \leq \partial_w \phi_i(x^*) \) by definition (4). When \( \phi_i \) is neither concave nor convex, \( \phi_i(x^*) + \nabla \phi_i(x^*) \cdot (-aw) \geq -a \partial_w \phi_i(x^*) > \frac{a^2}{2} w^T H^*_w w \). Hence \( x^* - aw \in \mathcal{F}_i(x^*) \) for any sufficiently small \( a \). Since \( I \) is finite, we can find a constant \( \epsilon \) such that \( x^* - aw \in \mathcal{F}_i(x^*) \) for all \( i \in I \) and \( 0 < a < \epsilon \). On the other hand \( x^* \in \Gamma^0 \) for all \( j \notin I \). According to Claim 1, \( x^* \in \bigcap_{j \notin I} \mathcal{F}_j(x^*) \). There exists a constant \( \epsilon_0 > 0 \) such that \( B(x^*, \epsilon_0) \subset \bigcap_{j \notin I} \mathcal{F}_j(x^*) \). Define \( v = -\min(\epsilon_0, \epsilon) w \). According to previous discussion, \( x^* + v \in \mathcal{F}(x^*) \).

Claim 1 and Claim 2 imply that \( \mathcal{F}(x^*) \) has nonempty interior.

**Proposition 3 (Fixed point).** If \( x^* \) is a fixed point of Algorithm 1, then \( x^* \) is a strong local optimum of (1).

**Proof.** We need to show that \( \nabla J(x^*) \cdot v \geq 0 \) for all \( v \in C(x^*) \). If \( \nabla J(x^*) = 0 \), then \( x^* \) is the global optimum of (1). Consider the case \( \nabla J(x^*) \neq 0 \). Claim that...
\( x^* \in \partial \Gamma \). First of all, \( x^* \) is a feasible point, since \( x^* \in \mathcal{F}(x^*) \subset \Gamma \). Moreover, since \( J \) is strictly convex, the optimal point \( x^* \) must be on the boundary of \( \mathcal{F}(x^*) \), i.e. \( x^* \in \partial \mathcal{F}(x^*) \) for some \( i \). According to (13), (14) and (15), \( \partial \mathcal{F}(x^*) \) equals to \( \{ x : \phi_i(x) = 0 \} \) in case 1, \( \{ x : \phi_i(x^*) + \nabla \phi_i(x^*)(x - x^*) = 0 \} \) in case 2, and \( \{ x : \phi_i(x^*) + \nabla \phi_i(x^*)(x - x^*) = \frac{1}{2}(x - x^*)^T H_i^*(x - x^*) \} \) in case 3. Then \( x^* \in \partial \mathcal{F}(x^*) \) implies that \( \phi_i(x^*) = 0 \). Hence \( x^* \in \partial \Gamma \). Let \( I = \{ i : \phi_i(x^*) = 0 \} = \{ i : x^* \in \mathcal{F}_i(x^*) \} \).

Consider \( v^* := \arg \min_{v \in C(x^*)} \nabla J(x^*) \cdot v \). If the minimum is not unique, use the tie breaking mechanism discussed in subsection 3.3. Claim that \( \tilde{v} \) reference point is more intricate, which is deeply related to the choice of the function \( \phi_i \)’s. The design considerations of \( \phi_i \)’s such that a nonempty convex feasible set always exists will be addressed in section 7.

Consider \( v^* = \arg \min_{v \in C(x^*)} \nabla J(x^*) \cdot v \). If the minimum is not unique, use the tie breaking mechanism discussed in subsection 3.3. Claim that \( \nabla \phi_i \cdot v^* \geq 0 \) for all \( i \in I \). For \( i \in I \) such that \( \phi_i \) is smooth at \( x^* \), the definition of \( C(x^*) \) implies that \( \nabla \phi_i(x^*) \cdot v^* = \nabla \phi_i(x^*) \cdot v^* \geq 0 \). For \( i \in I \) such that \( \phi_i \) is not smooth at \( x^* \), \( \nabla \phi_i \cdot v^* \geq 0 \) since \( \nabla \phi_i \in D \mathcal{F}_i := \{ d \in D \phi_i | \| d \cdot v^* \| \geq 0 \} \). On the other hand, since \( x^* \) is the optimal solution of the smooth optimization \( \min_{x \in \mathcal{F}(x^*)} J(x) \), the KKT condition is satisfied

\[
\nabla J(x^*) + \sum_{i=1}^{N} \lambda_i \nabla \phi_i(x^*) = 0.
\]

The complementary slackness condition implies that \( \lambda_i \leq 0 \) for \( i \in I \) and \( \lambda_j = 0 \) for \( j \notin I \). Hence

\[
\nabla J(x^*) \cdot v^* = -\sum_{i=1}^{N} \lambda_i \nabla \phi_i(x^*) \cdot v^* = -\sum_{i \in I} \lambda_i \nabla \phi_i(x^*) \cdot v^* \geq 0.
\]

Thus \( J(x^*) \cdot v \geq 0 \) for all \( v \in C(x^*) \). So \( x^* \) is a strong local optimum of (1).

**Remark 1.** Lemma 2 and Proposition 3 imply that a feasible \( x^* \) can always be improved by optimizing over the convex feasible set \( \mathcal{F}(x^*) \) if \( x^* \) itself is not a local optimum. However, the existence of nonempty convex feasible set for an infeasible reference point is more intricate, which is deeply related to the choice of the function \( \phi_i \)’s. The design considerations of \( \phi_i \)’s such that a nonempty convex feasible set always exists will be addressed in section 5.

**Lemma 4 (Strong descent).** For any feasible \( x^{(k)} \), the descent of the objective function satisfies that \( \nabla J(x^{(k+1)})(x^{(k)} - x^{(k+1)}) \geq 0 \). Moreover, if \( J(x^{(k+1)}) = J(x^{(k)}) \), then \( x^{(k+1)} = x^{(k)} \).

**Proof.** Claim that \( \mathcal{F}(x^{(k)}) \) is a subset of the half space \( H := \{ x | \nabla J(x^{(k+1)}) \cdot (x - x^{(k+1)}) \geq 0 \} \) as shown by the shaded area in Figure 6. If not, there must be some \( \bar{x} \in \mathcal{F}(x^{(k)}) \) such that \( \nabla J(x^{(k+1)})(\bar{x} - x^{(k+1)}) < 0 \). Let \( v := \bar{x} - x^{(k+1)} \). Since \( \mathcal{F}(x^{(k)}) \) is convex, then \( x^{(k+1)} + av \in \mathcal{F}(x^{(k)}) \) for \( a \in [0, 1] \). Since \( J \) is smooth, there exists a positive constant \( c > 0 \) such that \( J(x^{(k+1)} + av) \leq J(x^{(k+1)}) + a \nabla J(x^{(k+1)}) \cdot v + ca^2 \| v \|^2 \). When \( a \) is sufficiently small, the right hand side of the inequality is strictly smaller than \( J(x^{(k+1)}) \). Then \( J(x^{(k+1)} + av) < J(x^{(k+1)}) \), which contradicts with the fact that \( x^{(k+1)} \) is the minimum of \( J \) in the convex feasible set \( \mathcal{F}(x^{(k)}) \). Hence
Moreover, since $J$ is strictly convex, $J(x) > J(x^{(k+1)})$ for all $x \in H \setminus \{x^{(k+1)}\}$. Hence if $J(x^{(k+1)}) = J(x^{(k)}), x^{(k)} = x^{(k+1)}$.

**Proposition 5** (Convergence of strictly descending sequence). Consider the sequence $\{x^{(k)}\}$ generated by Algorithm 1. If $J(x^{(1)}) > J(x^{(2)}) > \cdots$, then the sequence $\{x^{(k)}\}$ converges to a weak local optimum $x^*$ of (1).

**Proof.** Since $J \geq 0$, the monotone sequence $\{J(x^{(k)})\}_{k=1}^\infty$ converges to some value $a \geq 0$. If $a = 0$, the sequence $\{x^{(k)}\}$ converges to the global optima. Consider the case $a > 0$. Since $J$ is strictly convex, the set $\{x : J(x) \leq J(x^{(1)})\}$ is compact. Then there exists a subsequence of $\{x^{(k)}\}$ that converges to $x^*$ such that $J(x^*) = a$. Since $\Gamma$ is closed, $x^* \in \Gamma$.

We need to show that the whole sequence $\{x^{(k)}\}$ converges to $x^*$. Suppose not, then there exists $\delta > 0$ such that $\forall K > 0$, there exists $k > K$ s.t. $\|x^{(k)} - x^*\| \geq \delta$. For any $\epsilon \in (0, \delta)$, there exists $k, j \in \mathbb{N}$ such that $\|x^{(k)} - x^*\| \leq \epsilon$ and $\|x^{(k+j)} - x^*\| \geq \delta$. Since $J$ is strictly convex, there exists $c > 0$ such that

$$J(x^{(k)}) \geq J(x^{(k+1)}) + \nabla J(x^{(k+1)})(x^{(k)} - x^{(k+1)}) + c \|x^{(k)} - x^{(k+1)}\|^2 \geq J(x^{(k)}) + j \|x^{(k)} - x^{(k+1)}\|^2 \geq J(x^{(k+j)}) + c \|x^{(k)} - x^*\| \geq a + c(\delta - \epsilon)^2,$$

which contradicts with the fact that $J(x^{(k)}) \to a$ as $\epsilon \to 0$. Note that the second inequality is due to $\nabla J(x^{(k+1)})(x^{(k)} - x^{(k+1)}) \geq 0$ in Lemma 4. The third inequality is by induction. The fourth inequality and the fifth inequality follow from $\Delta$-inequality. Hence we conclude that $\lim_{k \to \infty} x^{(k)} = x^*$.

Then we need to show that $x^*$ is a local optimum. The proof can be divided into two steps. First, we show that there is a subsequence of the convex feasible sets $\{F^{(k)}\}$ that converges point-wise to a suboptimal convex feasible set $\mathcal{G}$ at point $x^*$. Sub-optimality refers to the fact that the sub-gradients are not chosen according to subsection 3.3. Then we show that $x^*$ is the minimum of $J$ in $\mathcal{G}$. Thus the KKT condition is satisfied at $x^*$ and $x^*$ is a weak local optimum of (1).

Consider any $\phi_i, \|\phi_i(x^{(k)}) - \phi_i(x^*)\| = O(\|x^{(k)} - x^*\|)$ since the sub-gradient is locally bounded by Lemma 1. If $\phi_i$ is smooth at $x^*$, then $\nabla \phi_i(x^{(k)})$ converges to $d_i := \nabla \phi_i(x^*)$. If $\phi_i$ is not smooth at $x^*$, then $\nabla \phi_i$ is locally bounded, and there is a subsequence $\{\nabla \phi_i(x^{(k)})\}_{k \in \mathbb{N}}$ that converges to some $d_i \in \mathbb{R}^n$. Claim that $d_i \in D\phi_i(x^*)$. By definition (4), for any $v \in \mathbb{R}^n$, $\nabla \phi_i(x^{(k)}) \cdot v \leq \partial_v \phi_i(x^{(k)})$. Since $\phi_i$ is piece-wise smooth, then either $\partial_v \phi_i(x^{(k)}) \to \partial_v \phi_i(x^*)$ or $\partial_v \phi_i(x^{(k)}) \to -\partial_v \phi_i(x^*)$. Since $-\partial_v \phi_i(x^*) \leq \partial_v \phi_i(x^*)$ by (6), we have the following inequality,

$$d_i \cdot v = \lim_{k \to \infty} \nabla \phi_i(x^{(k)}) \cdot v \leq \lim_{k \to \infty} \partial_v \phi_i(x^{(k)}) \leq \partial_v \phi_i(x^*).$$

Hence by definition (4), $d_i \in D\phi_i(x^*)$. Then we can choose a subsequence $\{x^{(k_n)}\}_{k_n \in \mathbb{N}}$ such that $\phi_i(x^{(k_n)})$ converges to $\phi_i(x^*)$ and $\nabla \phi_i(x^{(k_n)})$ converges to $d_i \in D\phi_i(x^*)$ for all $i = 1, \cdots, N$. For simplicity and without loss of generality, we use the same notation for the subsequence as the original sequence in the following discussion.

Define a new convex feasible set $\mathcal{G}_i$ such that $\mathcal{G}_i = \Gamma_i$ if $\phi_i$ is concave, $\mathcal{G}_i = \{x : \phi_i(x^*) + d_i(x - x^*) \geq 0\}$ if $\phi_i$ is convex, and $\mathcal{G}_i = \{x : \phi_i(x^*) + d_i(x - x^*) \geq 0\}$ if $\phi_i$ is smooth at $x^*$. Hence if $\phi_i$ is not smooth at $x^*$, then $\{x \in H \cap \mathcal{G}_i : \phi_i(x) \leq \phi_i(x^*)\}$ is locally bounded by Lemma 1. If $\phi_i$ is smooth at $x^*$, then $\nabla \phi_i(x^*)$ converges to $d_i \in \mathbb{R}^n$. Claim that $\mathcal{G}_i$ is closed.

Hence if $\phi_i$ is smooth at $x^*$, then $\{x \in H \cap \mathcal{G}_i : \phi_i(x) \leq \phi_i(x^*)\}$ is locally bounded by Lemma 1. If $\phi_i$ is not smooth at $x^*$, then $\{x \in H \cap \mathcal{G}_i : \phi_i(x) \leq \phi_i(x^*)\}$ is not smooth at $x^*$. Since $\mathcal{G}_i$ is locally bounded by Lemma 1, $\{x \in H \cap \mathcal{G}_i : \phi_i(x) \leq \phi_i(x^*)\}$ is locally bounded by Lemma 1. Hence $\{x \in H \cap \mathcal{G}_i : \phi_i(x) \leq \phi_i(x^*)\}$ is locally bounded by Lemma 1.
\( \frac{1}{2}(x-x^*)^T H_i(x-x^*) \) if otherwise. Let \( G = \bigcap_{i=1}^{N} G_i \). Claim that \( G = \lim_{k \to \infty} F^{(k)} \) where \( \lim_{k \to \infty} F^{(k)} := \bigcap_{k \to \infty} \bigcup_{j=k}^{\infty} F^{(k)} \). Consider \( z \in G \). If \( \phi_i \) is concave, then \( z \in F^{(k)}_i := F^{(k)}(x^{(k)}) \) for all \( k \) according to (14). If \( \phi_i \) is convex, then

\[
\lim_{k \to \infty} \left[ \phi_i(x^{(k)}) + \nabla \phi_i(x^{(k)})(z-x^{(k)}) \right] = \phi_i(x^*) + d_i(z-x^*) \leq 0
\]

which implies that \( z \in \lim_{k \to \infty} F^{(k)}_i \). Similarly, we can show that if \( \phi_i \) is neither convex or concave, \( z \) also lies in the limits of \( F^{(k)}_i \). Hence \( z \in \bigcap_{i=1}^{N} \lim_{k \to \infty} F^{(k)}_i = \lim_{k \to \infty} F^{(k)} \). Since \( z \) is arbitrary, \( G \subset \lim_{k \to \infty} F^{(k)}_i \). For any \( z \in \lim_{k \to \infty} F^{(k)}_i \), then there exists a sequence \( \{z_k\} \) that converges to \( z \) such that \( z_k \in F^{(k)}_i \). For any \( i \), if \( \phi_i \) is concave, then \( z_k \in F^{(k)}_i = G_i \). Since \( G_i \) is closed, \( z \in G_i \). If \( \phi_i \) is convex, then

\[
\phi_i(x^*) + d_i(z-x^*) = \lim_{k \to \infty} \left[ \phi_i(x^{(k)}) + \nabla \phi_i(x^{(k)})(z_k-x^{(k)}) \right] \geq 0
\]

which implies that \( z \in G_i \). Similarly, we can show that \( z \in G_i \) if \( \phi_i \) is neither convex or concave. Hence \( z \in G \). And we verify that \( G = \lim_{k \to \infty} F^{(k)}_i \).

Claim that \( x^* = \arg\min_{x \in F^{(k)}} J(x) \). Suppose not, then there exists \( z \in G \) such that \( J(z) < J(x^*) \). For all \( \epsilon > 0 \), there exists \( y \in F^{(k)}_i \) for some \( k \in \mathbb{N} \) such that \( \|z-y\| < \epsilon \). Since \( J \) is smooth, then \( J(y) - J(z) < O(\epsilon) \). Thus \( J(y) < J(x^*) \) when \( \epsilon \) is sufficiently small. This contradicts with \( J(y) > J(x^{(k)}) \) for \( \forall k \in \mathbb{N} \). Hence \( J(z) \geq J(x^*) \) for all \( z \in G \). If there exists \( z \in G \) such that \( J(z) = J(x^*) \), then \( \frac{z+x^*}{2} \in G \) and \( J\left(\frac{z+x^*}{2}\right) < \frac{J(z)+J(x^*)}{2} = J(x^*) \) since \( G \) is convex and \( J \) is strictly convex. However, this contradicts with the conclusion that \( J(z) \geq J(x^*) \) for all \( z \in G \). Hence \( x^* \) is the unique minimum of \( J \) in the set \( G \). And the KKT condition is satisfied, i.e.

\[
\nabla J(x^*) + \sum_{i=1}^{N} \lambda_i d_i = 0 \text{ for } d_i \in D \phi_i(x^*) \text{. Then } x^* \text{ is a weak local optimum.}
\]

It is worth noting that if all \( \phi_i \)'s are smooth at \( x^* \), \( G = F(x^*) \). Then \( x^* \) is a fixed point, thus a strong local optimum by Proposition 3.

**Remark 2.** Lemma 4 and Proposition 5 justify the adoption of the terminate condition (12), which is indeed equivalent to the standard terminate condition (11), e.g. convergence in the objective function implies convergence in the solution.

### 4.2. Proof of the Main Result.

**Proof of Theorem 1.** If \( F^{(0)} \) is nonempty, then \( x^{(1)} \in \Gamma \) can be obtained by solving the convex optimization (10). By Lemma 2, \( F^{(1)} \) has nonempty interior, then \( x^{(2)} \in \Gamma \) can be obtained. By induction, we can conclude that \( x^{(i)} \in F^{(i)} \subset \Gamma \) for all \( i \geq 1 \). Moreover, as a better solution is found at each iteration, then \( J(x^{(1)}) \geq J(x^{(2)}) \geq \cdots \). This leads to two cases. The first case is that \( J(x^{(K)}) = J(x^{(K+1)}) \) for some \( K \), while the second case is that the cost keeps decreasing strictly, e.g. \( J(x^{(1)}) > J(x^{(2)}) > \cdots \). In the former case, the condition \( J(x^{(K)}) = J(x^{(K+1)}) \) is equivalent to \( x^{(K)} = x^{(K+1)} \) by Lemma 4. By induction, the algorithm converges, e.g. \( x^{(k)} = x^{(k+1)} \) and \( J(x^{(k)}) = J(x^{(k+1)}) \) for all \( k \geq K \). Moreover, as \( x^* := x^{(K)} \) is a fixed point, it is a strong local optimum by Proposition 3. If the cost keeps decreasing, e.g. \( J(x^{(1)}) > J(x^{(2)}) > \cdots \), then the sequence \( \{x^{(k)}\} \) converges to a weak local optimum \( x^* \) by Proposition 5.

### 5. Application on Motion Planning for Mobile Robots.

In this section, Algorithm 1 is applied to motion planning problem for mobile robots [22]. Its application to other systems can be found in [14]. The problem will be formulated in subsection 5.1 and then transformed into the benchmark form (1) and (2).
major difficulty in transforming the problem lies in finding the semi-convex function $\phi_i$ to describe the constraints. The method to construct $\phi_i$ in 2D is discussed in subsection 5.2 and the resulting convex feasible set is illustrated through examples in subsection 5.3. The performance of Algorithm 1 is shown in subsection 5.4 and compared to other non-convex optimization algorithms such as interior point method (ITP) and sequential quadratic programming (SQP).

5.1. Problem Statement. Suppose a mobile robot needs to plan a trajectory in $\mathbb{R}^2$ from start point $x_0$ to the goal point $G$ as shown in Figure 7. Let $x_q \in \mathbb{R}^2$ be the position of the robot at time step $q$. Define the decision variable as $x := [x_1^T \cdots x_h^T]^T \in \mathbb{R}^{2h}$ where $h$ is the planning horizon. The whole trajectory is denoted as $x = \begin{bmatrix} x_0^T & \cdots & x_h^T \end{bmatrix}^T \in \mathbb{R}^{2(h+2)}$. Define a sequence of selection functions $l_q : \mathbb{R}^{2h} \rightarrow \mathbb{R}^2$ as $x_q = l_q(x)$. The sampling time is $t_s$. Let the velocity at $q$ be $v_q := \frac{x_q - x_{q-1}}{t_s}$ and the acceleration at $q$ be $a_q := \frac{v_q - v_{q-1}}{t_s}$.

The Cost Function. The cost function of the problem is designed as

$$J(x) = w_1 \|x - x^*\|_Q^2 + w_2 \|x\|_S^2,$$

where $w_1, w_2 \in \mathbb{R}^+$. The first term $\|x - x^*\|_Q^2 := (x - x^*)^T Q (x - x^*)$ penalizes the distance from the target trajectory to the reference trajectory. The second term $\|x\|_S^2 := x^T S x$ penalizes the properties of the target trajectory itself, e.g. length of the trajectory and magnitude of acceleration. The matrices $Q, S \in \mathbb{R}^{(h+2) \times 2(h+2)}$ can be constructed from the following components: 1) matrix for position $Q_1 := I_{2(h+2)}$; 2) matrix for velocity $Q_2 := V^T V$ and 3) matrix for acceleration $Q_3 := A^T A$. Note that $V \in \mathbb{R}^{2(h+1) \times 2(h+2)}$ and $A \in \mathbb{R}^{2h \times 2(h+2)}$ are defined as

$$V = \frac{1}{t_s} \begin{bmatrix} I_2 & -I_2 & 0 & \cdots & 0 \\ 0 & I_2 & -I_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & I_2 & -I_2 \end{bmatrix}, \quad A = \frac{1}{t_s^2} \begin{bmatrix} I_2 & -2I_2 & I_2 & 0 & \cdots & 0 \\ 0 & I_2 & -2I_2 & I_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & I_2 & -2I_2 & I_2 \end{bmatrix},$$

which take finite differences of the trajectory $x$ such that $V x$ returns the velocity vector and $A x$ returns the acceleration vector. Then $Q := \sum_{i=1}^{3} c_i^Q Q_i$ and $S := \sum_{i=1}^{3} c_i^S Q_i$ where $c_i^Q$ and $c_i^S$ are positive constants. Assumption 1 is satisfied.

The Constraints. The obstacles in the environment are denoted as $O_j \in \mathbb{R}^2$ for $j \in \mathbb{N}$. Each $O_j$ is simply connected and open. For example, there are seven such obstacles in Figure 7 where five of them are static and two are dynamic. Let

\[\text{Note that the obstacles may not be physical, but denote the infeasible area in the state space.}\]
$\mathcal{O}_j$ be the obstacle $j$ centered at the origin. The area occupied by the obstacle $j$ at time step $q$ is defined as $\mathcal{O}_j(q)$. Define a linear isometry $T_{j,q} : \mathbb{R}^2 \to \mathbb{R}^2$ such that $\mathcal{O}_j = T_{j,q}(\mathcal{O}_j(q))$. Then $x_q \notin \mathcal{O}_j(q)$ is equivalent to $T_{j,q}(x_q) \notin \mathcal{O}_j$. Hence the constraint for the optimization is

$$x \in \Gamma := \{ x \in \mathbb{R}^{2h} : T_{j,q}(l_q(x)) \notin \mathcal{O}_j, \forall j, \forall q = 1, \cdots, h \}. \tag{19}$$

It is easy to verify that Assumption 2 is satisfied.

5.2. Transforming the Problem. In order to apply Algorithm 1, the constraint (19) needs to be written in the form of (2), e.g. the function $\phi_i$’s need to be constructed. Assume $\mathcal{O}_j$ does not contain concave corner. If it does, we need to partition it to several overlapping obstacles that do not have concave corners, as discussed in subsection 2.2. For each $\mathcal{O}_j$, we first construct a simple function $\varphi_j : \mathbb{R}^2 \to \mathbb{R}$ and then use $\varphi_j$ to construct $\phi_i$. The function $\varphi_j$ is continuous, piecewise smooth and semi-convex such that $\mathcal{O}_j = \{ x \in \mathbb{R}^2 : \varphi_j(x) < 0 \}$ and $\partial \mathcal{O}_j = \{ x \in \mathbb{R}^2 : \varphi_j(x) = 0 \}$. We call $\varphi_j$ as a safety index in the following discussion, since it typically measures the distance to an obstacle. As shown in Figure 7, there are three types of obstacles. The construction of $\varphi_j$ in each case is discussed below.

**Convex obstacle.** A convex obstacle refers to the case that $\mathcal{O}_j$ is compact and convex. In this case, $\varphi_j$ is defined to be the signed distance function to $\mathcal{O}_j$, i.e.

$$\varphi_j(x) := \begin{cases} \min_{y \in \partial \mathcal{O}_j} \| x - y \| & x \notin \mathcal{O}_j \\ -\min_{y \in \partial \mathcal{O}_j} \| x - y \| & x \in \mathcal{O}_j \end{cases}. \tag{20}$$

**Boundary obstacle.** A boundary obstacle refers to a non-compact $\mathcal{O}_j$ such that there is an affine parameterization of $\partial \mathcal{O}_j$, e.g. if we rotate and align the obstacle properly, there exists a continuous and piecewise smooth semi-convex function $f : \mathbb{R} \to \mathbb{R}$ such that $p_2 = f(p_1)$ describes $\partial \mathcal{O}_j$ where $x = (p_1, p_2) \in \mathbb{R}^2$. Then $\varphi_j$ is defined as the directional distance along $(0, 1)$ to the boundary, i.e.

$$\varphi_j(x) := f(p_1) - p_2. \tag{21}$$

**Non-convex obstacle.** A non-convex obstacle refers to the case that $\mathcal{O}_j$ is compact, but non-convex. As the signed distance function for a non-convex set is not semi-convex, the strategy is to introduce directional distance. Denote the smallest convex envelop of the obstacle $\mathcal{O}_j$ as $\hat{\mathcal{O}}_j$. Then $\varphi_j$ is defined as a signed directional distance function which computes the minimum distance from a point $x$ to $\partial \mathcal{O}_j$ in the direction that is perpendicular to $\partial \hat{\mathcal{O}}_j$ as shown in Figure 8b. Let $\delta$ be a correspondence function that maps a point $y \in \partial \hat{\mathcal{O}}_j$ to the closest $z \in \partial \mathcal{O}_j$ such that $z - y$ is perpendicular to $\partial \hat{\mathcal{O}}_j$ at $y$. It is assumed that $\delta$ is bijective and $\| y - \delta(y) \|$ is semi-convex on $y$. Then

$$\varphi_j(x) = \begin{cases} \min_{y \in \partial \mathcal{O}_j} \| x - y \| + \| y - \delta(y) \| & x \notin \hat{\mathcal{O}}_j \\ -\min_{y \in \partial \mathcal{O}_j} \| x - y \| - \| y - \delta(y) \| & x \in \hat{\mathcal{O}}_j \end{cases}. \tag{22}$$

According to (20), (21) and (22), the contours of the safety indices are shown in Figure 8. Based on the safety indices, (19) can be re-written as

$$x \in \Gamma = \bigcap_{j,q} \{ x \in \mathbb{R}^{2h} : \varphi_j(T_{j,q}(l_q(x))) \geq 0 \}. \tag{23}$$
Define \( \phi_{j,q}(x) := \varphi_j(T_{j,q}(l_q(x))) \) for all \( j \) and \( q \). It is easy to verify that the first and the second arguments in Assumption 3 holds for \( \phi_{j,q} \). The third argument depends on how the obstacles are partitioned. Let \( \Gamma_{j,q} := \{ x \in \mathbb{R}^2 : \phi_{j,q}(x) \geq 0 \} \). The convex feasible set \( \mathcal{F} := \bigcap_{j,q} \Gamma_{j,q} \) can be constructed according to the discussion in subsection 3.2. And Algorithm 1 can be applied.

5.3. The Convex Feasible Set - Examples. In this part, the configuration of convex feasible sets will be illustrated with examples. Since \( \phi_{j,q} \) only depends on \( x_q, F_{j,q}(x') \) only poses constraints on \( x_q \). For example, according to (14), the convex feasible set for a convex \( \phi_{j,q} \) is

\[
\mathcal{F}_{j,q}(x') = \{ x : \varphi_j(T_{j,q}(x'_q)) + \nabla \varphi_j(T_{j,q}(x'_q)) \nabla T_{j,q}(x'_q)(x_q - x'_q) \geq 0 \}.
\]

For simplicity, define \( \mathcal{F}^q_{j,q}(x') := l_q(\mathcal{F}_{j,q}(x')) \in \mathbb{R}^2 \). In the following discussion, we will first illustrate \( \mathcal{F}^q_{j,q}(x') \) in \( \mathbb{R}^2 \) and then \( \bigcap_q \mathcal{F}_{j,q}(x') \) in \( \mathbb{R}^{2h} \).

The convex feasible set in \( \mathbb{R}^2 \). We illustrate the convex feasible set for one obstacle at one time step. For simplicity, sub-scripts \( j \) and \( q \) are removed and \( T_{j,q} \) is assumed to be identity. Consider a polygon obstacle with vertices \( a = (1,1), b = (-1,1), c = (-1,-1) \) and \( d = (1,-1) \) as shown in Figure 9a. Let \( \|x\|_\infty \) denote the \( l_\infty \) norm, i.e. \( \|x\|_\infty := \max\{|p_1|,|p_2|\} \). Then according to (20),

\[
\varphi(x) = \begin{cases} 
\max\{-1 - p_1, p_1 - 1, p_2 - 1, -1 - p_2\} & \|x\|_\infty \leq 1 \\
\min\{|x - a|,|x - b|,|x - c|,|x - d|\} & |p_1| > 1, |p_2| > 1 \\
\min\{|p_1 - 1|,|p_1 + 1|\} & |p_1| > 1, |p_2| < 1 \\
\min\{|p_2 - 1|,|p_2 + 1|\} & |p_1| < 1, |p_2| > 1 
\end{cases}
\]

For a reference point \( x' = (-1.5,-1.5), \varphi(x') = \frac{\sqrt{2}}{2} \) and \( \nabla \varphi(x') = \frac{\sqrt{2}}{2} [-1,-1] \).

The convex feasible set is \( \mathcal{F}(x') = \{ x : \varphi(x') + \nabla \varphi(x')(x - x') \geq 0 \} = \{ x : \frac{\sqrt{2}}{2}(-p_1 - p_2 + 2) \geq 0 \} \). The bowl-shaped surface in Figure 9a illustrates the safety index \( \varphi \). The plane that is tangent to the safety index satisfies the function \( \frac{\sqrt{2}}{2}(-p_1 - p_2 + 2) \). Since \( \varphi \) is convex, the tangent plane is always below \( \varphi \). The convex feasible set is just the projection of the plane onto the zero level set.

Consider the case that points \( c \) and \( d \) are not connected by a straight line, but a
concave curve \( p_2 = -(p_1)^2 \) as shown in Figure 9b. Then according to (22),

\[
\varphi(x) = \begin{cases} 
\max\{-(p_1)^2 - p_2, p_1 - 1, p_2 - 1, -1 - p_2\} & \|x\|_{\infty} \leq 1, p_2 \geq -(p_1)^2 \\
\min\{\|x-a\|, \|x-b\|, \|x-c\|, \|x-d\|\} & |p_1| > 1, |p_2| > 1 \\
\min\{|p_1 - 1|, |p_1 + 1|\} & |p_1| > 1, |p_2| < 1 \\
-(p_1)^2 - p_2 & p_2 > 1 \\
-p_2 & p_2 \leq -(p_1)^2
\end{cases}
\]

Then the curvature of \( \varphi \) is bounded below by \( -H^* = [-2, 0; 00] \). For a reference point \( x^r = (0, -1) \), \( \varphi(x^r) = 1 \) and \( \nabla \varphi(x^r) = [0, -1] \). The convex feasible set is \( \mathcal{F}(x^r) = \{x : \varphi(x^r) + \nabla \varphi(x^r)(x - x^r) \geq \frac{1}{2}(x - x^r)^TH^*(x - x^r)\} = \{x : -p_2 - (p_1)^2 \geq 0\} \). The bowl-shaped surface in Figure 9b illustrates the safety index \( \varphi \). The parabolic surface that is tangent to the safety index represents the function \( -p_2 - (p_1)^2 \). The convex feasible set is the projection of the surface onto the zero level set.

**The convex feasible set in higher dimension.** We illustrate the convex feasible set for one obstacle over the entire time horizon. The obstacle is shown in Figure 10a, which is a translated version of the obstacle in Figure 9a. The reference trajectory is shown in red, which violates the safety constraint. Figure 10b shows the convex feasible sets computed for each time step. Those sets formulate a corridor around the time-augmented obstacle. A new trajectory will be computed in the corridor. Although the projection of the corridor into \( \mathbb{R}^2 \) is not convex, each time slice of the corridor is convex. In \( \mathbb{R}^{2h} \), those slices are stuck together orthogonally, hence formulate a convex subset of \( \mathbb{R}^{2h} \).

When there are multiple obstacles, the existence of a corridor that bypasses all
time-augmented obstacles under the proposed algorithm is hard to guarantee if the reference trajectory is infeasible as pointed out in Remark 1. In Lemma 6, we show the existence of the convex feasible set for infeasible reference trajectory under certain geometric conditions.

**Lemma 6 (Feasibility).** If all obstacles are convex and have disjoint closures, i.e. \( \mathcal{O}_i(q) \cap \mathcal{O}_j(q) = \emptyset \) for all \( q \) and \( i \neq j \), then the convex feasible set \( \mathcal{F}(x^r) \) has nonempty interior for all \( x^r \in \mathbb{R}^n \) when \( \phi_{j,q} \) is chosen according to (20) and (23).

**Proof.** Considering Lemma 2, we only need to show the case that \( x^r \) is not feasible, e.g. some \( \phi_{j,q}(x^r) \) is negative. Fix \( q \), since \( \mathcal{O}_j(q) \)'s are disjoint, only one \( \phi_{j,q} \) can be negative. Without loss of generality, suppose \( \phi_{1,q}(x^r) < 0 \) and \( \phi_{j,q}(x^r) \geq 0 \) for \( j \geq 2 \). Since the safety index \( \varphi_j \) is a signed distance function in (20) and \( T_{j,q} \) is an isometry, \( \| \nabla \varphi_j \| = 1 \) and \( \| T_{j,q} \| = 1 \). Then the ball \( B(x^r_q, \varphi_j(T_{j,q}(x^r_q))) \) is a subset of \( \mathcal{F}^q_{j,q}(x^r) \) according to (24) for all \( j \) and \( q \). Hence \( B(x^r_q, d^*) \subseteq \bigcap_{j \geq 2} \mathcal{F}^q_{j,q}(x^r) \) where \( d^* = \min_{j \geq 2} \varphi_j(T_{j,q}(x^r_q)) \) is the minimum distance to \( \bigcup_j \mathcal{O}_j(q) \). According to (20), \( \mathcal{F}^q_{1,q}(x^r) \) is tangent to \( \partial \mathcal{O}_1(q) \) at \( x^* \) such that \( \varphi_1(T_{1,q}(x^r_q)) = -\| x^r_q - x^* \| \) as shown in Figure 11. Since \( \mathcal{O}_1(q) \cap \bigcup_j \mathcal{O}_j(q) = \emptyset \) for \( j \geq 2 \), \( d^* > -\varphi_1(T_{1,q}(x^r_q)) = \| x^r_q - x^* \| \), which implies that the set \( \bigcap_j \mathcal{F}^q_{j,q}(x^r) \) has nonempty interior. Then \( \bigcap_j \mathcal{F}^q_{j,q}(x^r) \) has nonempty interior where \( \oplus \) means direct sum.

**Remark 3.** Lemma 6 implies that \( \mathcal{F}(0) \) is nonempty for any \( x^{(0)} \). Then by Theorem 1, the algorithm converges to a local optimum for any \( x^{(0)} \in \mathbb{R}^n \) if all obstacles are convex and have disjoint closures.

### 5.4. Performance and Comparison

The performance of Algorithm 1 will be illustrated through two examples, which will also be compared to the performance of existing non-convex optimization methods such as interior point (ITP) and sequential quadratic programming (SQP). For simplicity, only convex obstacles and convex boundaries are considered⁹. Algorithm 1 is implemented in both Matlab and C++. The convex optimization problem (10) is solved using the interior-point-convex method in `quadprog` in Matlab and the interior point method in Knitro [3] in C++. For comparison, (1) is also solved directly using ITP and SQP methods in `fmincon` [1] in Matlab and in Knitro in C++. To create fair comparison, the gradient and the hessian of the objective function \( J \) and the optimal sub-gradients \( \nabla \phi_i \) of the constraint function \( \phi_i \)'s are also provided to the ITP and SQP solvers.

In the examples, \( x_0 = (0,0) \) and \( G = (9,0) \). The planning horizon \( h \) goes from 30 to 100. \( t_s = (h + 1)^{-1} \). The cost function (18) penalizes the average acceleration

\[ \text{Fig. 11: Existence of convex feasible set for an infeasible reference point.} \]
along the trajectory, e.g. \( Q = 0 \) and \( S = h^{-1} A^T A \). When \( h \to \infty \), \( J(x) \to \int_0^1 \| \dot{x} \|^2 dt \) where \( x : [0, 1] \to \mathbb{R}^n \) is a continuous trajectory with \( x(0) = x_0 \) and \( x(1) = G \). The initial reference \( x^{(0)} \) is chosen to be a straight line connecting \( x_0 \) and \( G \) with equally sampled waypoints. In the first scenario, there are three disjoint convex obstacles as shown in Figure 12. In the second scenario, there are three disjoint infeasible sets, two of which contain concave corners. Then they are partitioned into five overlapping convex obstacles as shown in Figure 13. In the constraint, a distance margin of 0.25 to the obstacles is required.

The computation time under different solvers is listed in Table 1. The first column shows the horizon. In the first row, \( h = 100 \) in scenario 1 and \( h = 60 \) in scenario 2. In the remaining rows, \( h \) is the same in the two scenarios. In the second column, “-M” means the algorithm is run in Matlab and “-C” means the algorithm is run in C++. Under each scenario, the first column shows the final cost. The second column is the total number of iterations. The third and fourth columns are the total computation time and the average computation time per iteration respectively (only the entries that are less than 100 ms are shown). It does happen that the algorithms find different local optima, though CFS-M and CFS-C always find the same solution. In terms of computation time, Algorithm 1 always outperforms ITP and SQP, since it requires less time per iteration and fewer iterations to converge. This is due to the fact CFS does not require additional line search after solving (10) as is needed in ITP and SQP, hence saving time during each iteration. CFS requires fewer iterations to converge since it can take unconstrained step length \( \| x^{(k+1)} - x^{(k)} \| \) in the convex feasible set as will be shown later. Moreover, Algorithm 1 scales much better than ITP and SQP, as the computation time and time per iteration in CFS-C go up almost linearly with respect to \( h \) (or the number of variables).

The computation time of CFS consists of two parts: 1) the processing time, i.e. the time to compute \( F \) and 2) the optimization time, i.e. the time to solve (10). As shown in Figure 14, the two parts grow with \( h \). In Matlab, the processing time dominates, while the optimization time dominates in C++.

To better illustrate the advantage of Algorithm 1, the runtime statistics across all methods when \( h = 100 \) in the first scenario is shown in Figure 15. The first log-log figure shows the cost \( J(x^{(k)}) \) versus iteration \( k \), while the second semi-log figure shows the feasibility error \( \max \{ 0, -\phi_1(x^{(k)}), \ldots, -\phi_N(x^{(k)}) \} \) versus \( k \). At the beginning, the cost \( J = 0 \) and the feasibility error is 0.75. In Algorithm 1, \( x^{(k)} \) becomes feasible at the first iteration while the cost \( J(x^{(k)}) \) jumps up. In the following iterations, the cost goes down and converges to the optimum value. In ITP-C, the problem becomes feasible at the third iteration. In SQP-C, it is the fifth iteration. In order to make the problem feasible, the cost jumps much higher in ITP-C than in CFS-C. Once the problem is feasible, it also takes more iterations for ITP-C and SQP-C to converge than CFS-C. On the other hand, ITP-M and SQP-M have very small step length in the beginning. The problem only becomes feasible after 100 iterations. But once the problem is feasible, the performance of ITP-M and SQP-M is similar to that of ITP-C and SQP-C. Note that the cost below 1 is not shown in the figure.

The optimal trajectories computed by Algorithm 1 for different \( h \) in the first scenario is shown in Figure 12. Those trajectories converge to a continuous trajectory when \( h \) goes up. Figure 13 illustrates the trajectories before convergence in CFS-C and ITP-C in scenario 2 when \( h = 50 \). For ITP-C, the trajectories are shown every

\[10^\text{In the C++ solver, the constraint is limited by 300. Hence when there are five obstacles, the maximum allowed } h \text{ is 60.}\]
Table 1: Comparison among different algorithms.

| h   | Method | Cost  | Iter | Time  | dT   | Cost  | Iter | Time  | dT   |
|-----|--------|-------|------|-------|------|-------|------|-------|------|
| 100 or 60 | SQP-M | 1358.9 | 470 | 50.6s | - | 5387.5 | 198 | 60.1s | - |
|     | ITP-M | 1358.9 | 18 | 1.8s | 98.8ms | 5413.2 | 6 | 344.9ms | 57.5ms |
|     | SQP-C | 1347.6 | 123 | 47.3s | - | 5489.6 | 52 | 10.2s | - |
|     | ITP-C | 1341.7 | 306 | 2.9s | 9.5ms | 5349.6 | 123 | 595.0ms | 4.8ms |
|     | CFS-M | 1358.9 | 18 | 74.4ms | 4.1ms | 5413.2 | 6 | 27.3ms | 4.6ms |
|     | SQP-C | 1347.6 | 123 | 47.3s | - | 5489.6 | 52 | 10.2s | - |
|     | ITP-C | 1341.7 | 306 | 2.9s | 9.5ms | 5349.6 | 123 | 595.0ms | 4.8ms |
|     | CFS-C | 1358.9 | 18 | 74.4ms | 4.1ms | 5413.2 | 6 | 27.3ms | 4.6ms |
| 50  | SQP-M | 2299.5 | 110 | 26.8s | - | 5539.6 | 164 | 40.0s | - |
|     | ITP-M | 1308.3 | 187 | 8.8s | 47.1ms | 5372.8 | 268 | 11.3s | 42.2ms |
|     | SQP-C | 1308.2 | 52 | 3.4s | 65.4ms | 5682.0 | 62 | 5.7s | 91.9ms |
|     | ITP-C | 1275.1 | 131 | 390ms | 3.0ms | 5555.8 | 96 | 495.6ms | 5.2ms |
|     | CFS-C | 1458.2 | 8 | 23.7ms | 3.0ms | 5394.2 | 5 | 21.0ms | 4.2ms |
| 40  | SQP-M | 3391.6 | 97 | 18.6s | - | 5518.1 | 127 | 23.2s | - |
|     | ITP-M | 1317.0 | 150 | 5.2s | 34.7ms | 5549.8 | 156 | 5.6s | 35.9ms |
|     | SQP-C | 1317.0 | 40 | 1.5s | 37.5ms | 5568.9 | 69 | 2.8s | 40.6ms |
|     | ITP-C | 1170.5 | 102 | 240.5ms | 2.4ms | 5399.2 | 91 | 299.4ms | 3.2ms |
|     | CFS-C | 1317.0 | 8 | 16.5ms | 2.1ms | 5399.2 | 6 | 19.0ms | 3.2ms |
| 30  | SQP-M | 1039.2 | 106 | 8.4s | 79.2ms | 5075.8 | 90 | 11.0s | - |
|     | ITP-M | 1039.2 | 109 | 2.8s | 25.7ms | 5162.4 | 127 | 3.2s | 25.2ms |
|     | SQP-C | 1453.3 | 27 | 379.1ms | 14.0ms | 5444.3 | 42 | 1.5s | 35.7ms |
|     | ITP-C | 1039.2 | 59 | 118.5ms | 2.0ms | 5320.8 | 67 | 125.5ms | 1.9ms |
|     | CFS-C | 1039.2 | 12 | 19.0ms | 1.6ms | 5167.3 | 5 | 12.2ms | 2.4ms |

With respect to the results, we conclude that Algorithm 1 is time-efficient, local-optimal and scalable.

6. Conclusion. This paper introduced a fast algorithm for real-time motion planning based on the convex feasible set. The CFS algorithm can handle problems that have convex cost function and non-convex constraints which are usually encountered in robot motion planning. By computing a convex feasible set within the non-convex constraints, the non-convex optimization problem is transformed into a convex optimization. Then by iteration, we can efficiently eliminate the error introduced by the convexification. It is proved in the paper that the proposed algorithm is feasible and stable. Moreover, it can converge to a local optimum if either the initial reference satisfies certain conditions or the constraints satisfy certain conditions. The performance of CFS is compared to that of ITP and SQP. It is shown that CFS outperforms ITP and SQP in computation time, hence better suited for real-time applications. In the future, the method for computing the convex feasible sets for iteration in the first ten iterations and then every ten iterations in the remaining iterations. The step length in CFS-C is much larger than that in ITP-C, which explains why CFS requires fewer iterations to become feasible and fewer iterations to converge. The trajectories are feasible and smooth in every iteration in CFS. Hence even if time is limited, we can safely stop the iteration and get a good enough sub-optimal trajectory before convergence.
Fig. 13: Scenario 2 and the trajectories before convergence for $h = 50$.

Fig. 14: The decomposed time per iteration using Algorithm 1 in scenario 1.

infeasible references in complicated environments will be explored.

REFERENCES

[1] Optimization toolbox, constrained optimization, fmincon, https://www.mathworks.com/help/optim/ug/fmincon.html.
[2] B. Açıkmese, J. M. Carson, and L. Blackmore, Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem, IEEE Transactions on Control Systems Technology, 21 (2013), pp. 2104–2113.
[3] R. H. Byrd, J. Nocedal, and R. A. Waltz, Knitro: An Integrated Package for Nonlinear Optimization, Springer US, Boston, MA, 2006, pp. 35–59.
[4] C. Chen, M. Rickert, and A. Knoll, Path planning with orientation-aware space exploration guided heuristic search for autonomous parking and maneuvering, in IEEE Intelligent Vehicles Symposium (IV), 2015, pp. 1148–1153.
[5] F. H. Clarke, Generalized gradients and applications, Transactions of the American Mathematical Society, 205 (1975), pp. 247–262.
[6] G. Eichfelder and J. Povh, On the set-semidefinite representation of nonconvex quadratic programs over arbitrary feasible sets, Optimization Letters, 7 (2013), pp. 1373–1386.
[7] G. B. Folland, Real analysis: modern techniques and their applications, John Wiley & Sons, 2013.
[8] E. Frazzoli, M. A. Dahler, and E. Feron, Real-time motion planning for agile autonomous vehicles, Journal of Guidance, Control, and Dynamics, 25 (2002), pp. 116–129.
[9] M. W. Harris and B. Açıkmese, Lossless convexification of non-convex optimal control problems for state constrained linear systems, Automatica, 50 (2014), pp. 2304–2311.
[10] T. M. Howard, C. J. Green, and A. Kelly, Receding horizon model-predictive control for mobile robot navigation of intricate paths, in Field and Service Robotics, Springer, 2010, pp. 69–78.
[11] T. A. Johansen, T. I. Fossen, and S. P. Berge, Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming, IEEE Transactions on Control Systems Technology, 12 (2004), pp. 211–216.
[12] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, Real-time motion planning with applications to autonomous urban driving, IEEE Transactions on Control Systems Technology, 17 (2009), pp. 1105–1118.
Fig. 15: The run time statistics in scenario 1.

[13] J.-C. Latombe, Robot motion planning, vol. 124, Springer Science & Business Media, 2012.
[14] C. Liu, C.-Y. Lin, Y. Wang, and M. Tomizuka, Convex feasible set algorithm for constrained trajectory smoothing, in Proceedings of 2017 American Control Conference, 2017.
[15] C. Liu and M. Tomizuka, Algorithmic safety measures for intelligent industrial co-robots, in Proceedings of Robotics and Automation (ICRA), 2016 IEEE International Conference on, IEEE, 2016, pp. 3095 – 3102.
[16] C. Liu and M. Tomizuka, Enabling safe freeway driving for automated vehicles, in 2016 American Control Conference, IEEE, 2016, pp. 3461 – 3467.
[17] C. Liu and M. Tomizuka, Real time trajectory optimization for nonlinear robotic systems: Relaxation and convexification, System & Control Letters, (2016), p. in review.
[18] X. Liu, Autonomous trajectory planning by convex optimization, PhD thesis, Iowa State University, 2013.
[19] X. Liu and P. Lu, Solving nonconvex optimal control problems by convex optimization, Journal of Guidance, Control, and Dynamics, (2014).
[20] J. Nocedal and S. Wright, Numerical optimization, Springer Science & Business Media, 2006.
[21] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, CHOMP: Gradient optimization techniques for efficient motion planning, in Robotics and Automation, 2009. ICRA’09. IEEE International Conference on, IEEE, 2009, pp. 489–494.
[22] A. V. Savkin, A. S. Matveev, M. Hoy, and C. Wang, Safe Robot Navigation Among Moving and Steady Obstacles, Butterworth-Heinemann, 2015.
[23] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel, Finding locally optimal, collision-free trajectories with sequential convex optimization., in Robotics: science and systems, vol. 9, Citeseer, 2013, pp. 1–10.
[24] P. Spellucci, A new technique for inconsistent QP problems in the SQP method, Mathematical Methods of Operations Research, 47 (1998), pp. 355–400.
[25] R. G. Strongin and Y. D. Sergeyev, Global optimization with non-convex constraints: Sequential and parallel algorithms, vol. 45, Springer Science & Business Media, 2013.
[26] M. Tawarmalani and N. V. Sahinidis, Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications, vol. 65, Springer Science & Business Media, 2002.
[27] K. Tone, Revisions of constraint approximations in the successive QP method for nonlinear programming problems, Mathematical Programming, 26 (1983), pp. 144–152.
[28] J. Van Den Berg, P. Abbeel, and K. Goldberg, LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information, The International Journal of Robotics Research, 30 (2011), pp. 895–913.
[29] R. J. Vanderbei and D. F. Shanno, An interior-point algorithm for nonconvex nonlinear programming, Computational Optimization and Applications, 13 (1999), pp. 231–252.
[30] Z. Zhu, E. Schmerling, and M. Pavone, A convex optimization approach to smooth trajectories for motion planning with car-like robots, in Decision and Control (CDC), 2015 IEEE Conference on, IEEE, 2015, pp. 835–842.