



Surrogate-based Distributed Optimisation for Expensive Black-box Functions

DOI:

[10.1016/j.automatica.2020.109407](https://doi.org/10.1016/j.automatica.2020.109407)

Document Version

Accepted author manuscript

[Link to publication record in Manchester Research Explorer](#)

Citation for published version (APA):

Li, Z., Dong, Z., Liang, Z., & Ding, Z. (2020). Surrogate-based Distributed Optimisation for Expensive Black-box Functions. *Automatica*, 125, Article 109407. <https://doi.org/10.1016/j.automatica.2020.109407>

Published in:

Automatica

Citing this paper

Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights

Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy

If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [<http://man.ac.uk/04Y6Bo>] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.



Surrogate-based Distributed Optimisation for Expensive Black-box Functions [★]

Zhongguo Li ^a, Zhen Dong ^a, Zhongchao Liang ^b, Zhengtao Ding ^a

^a*Department of Electrical and Electronic Engineering, University of Manchester, Manchester, M13 9PL, UK*

^b*School of Mechanical Engineering and Automation, Northeastern University, Shenyang, 110819, China*

Abstract

This paper considers distributed optimisation problems with black-box functions using surrogate-assisted methods. Since the cost functions and their derivatives are usually impossible to be expressed by explicit functions due to the complexity of modern systems, function calls have to be performed to obtain those values. Moreover, the cost functions are often expensive to evaluate, and therefore designers prefer to reduce the number of evaluations. In this paper, surrogate-based methods are utilised to approximate the true functions, and conditions for constructing smooth and convex surrogates are established, by which the requirements for explicit functions are eliminated. To improve the quality of surrogate models, a distance-based infill strategy is proposed to balance the exploitation and exploration, which guarantees the density of the decision sequence in a compact set. Then, a distributed optimisation algorithm is developed to solve the reformulated auxiliary sub-problems, and the convergence of the proposed algorithm is established via Lyapunov theory. Simulation examples are provided to validate the effectiveness of the theoretical development and demonstrate the potential significance of the framework.

Key words: Distributed algorithms, expensive optimisation methods, black-box functions, surrogate models, multi-agent systems.

1 Introduction

Over the past decade, distributed optimisation for multi-agent systems has received significant attention, due to its wide applications in engineering, social and economic networks [1–5]. For large-scale and complex systems over networks, the traditional centralised methods are no longer feasible mainly owing to the lack of central authority, massive data communication, limited processing power, robustness and privacy concerns [1]. Distributed algorithms, on the other hand, can be implemented via local interactions among networked agents. Private information are confined to local communication without sharing to remote fusion centres, which consequently relieves the privacy concern and communication burden. More importantly, the distributed approaches do not require central processors, and therefore improve

the robustness and resilience to single endpoint failures.

Recent advances on distributed optimisation have been reported in many state-of-the-art works, e.g., [3, 6–12]. Discrete-time algorithms are proposed in [3, 6] where consensus algorithms with sub-gradients are used in [3], and a primal-dual method is developed with inequality constraints in [6]. More recently, continuous-time algorithms have been proposed to solve the distributed optimisation problems. The saddle-point method is studied for balanced graphs [9], and some variations have been further explored to deal with more general classes of problems [10]. Fixed-time consensus algorithms have been introduced to solve distributed optimisation problems [13–15], which demonstrate a fast convergence to the optimal solutions. Projection-based methods have been proposed to handle the problems with set constraints [11, 16].

The aforementioned works assume that the local objective functions and their derivatives are directly available, which poses a critical limitation to applications of those algorithms. It is usually impractical to obtain the explicit objective functions owing to the complexity of modern systems. Moreover, evaluating the black-box

[★] This work was supported by the Science and Technology Facilities Council through Newton Fund under Grant ST/N006852/1. Corresponding author: Zhengtao Ding.

Email addresses: zhongguo.li@manchester.ac.uk (Zhongguo Li), zhen.dong@manchester.ac.uk (Zhen Dong), liangzc@me.neu.edu.cn (Zhongchao Liang), zhengtao.ding@manchester.ac.uk (Zhengtao Ding).

functions is often time-consuming for many engineering applications, for example, aerospace design optimization [17, 18], where the black-box models are considered as expensive or costly functions [19, 20]. The traditional gradient-based algorithms using real-time stimulations are infeasible to be implemented, since a large number of function evaluations are required. To eliminate the requirement for costly gradients, derivative-free optimisation has been actively studied mainly in centralised settings (referring to [21] for a comprehensive overview). Besides, it should be noted that models obtained from trust-region methods may be local, because they are confined to a specific subregion of interest [22].

The surrogate model, also known as meta model or response surface, has been widely investigated for black-box functions in both single and multi-objective optimisation problems [17, 19]. It approximates the objectives by using interpolation approaches, e.g., radial basis function methods [20], polynomial interpolation [23] and Kriging models [24]. Considering that evaluating the black-box functions is expensive, we have to make full use of the currently available information to choose the next evaluation point wisely based on the present surrogate and the infill strategy. When selecting the next point of black-box evaluation, there are conflicting dual goals that should be considered simultaneously: exploiting the current model by selecting the point having a low objective value, and exploring the undiscovered space to improve the fitness of the surrogate. Infill criteria, also referred as adaptive sampling, have been proposed in the literature aiming to balance the exploitation and exploration, see [17, 25] for a comprehensive overview. In [20, 26, 27], an effective distance-based infill strategy has been developed, which aims to optimise the current surrogate subject to a constraint that takes those points of some distance from previously evaluated points. With this setting, exploitation is reflected by the optimisation of current surrogate, and exploration is incorporated in the constraint. Then, centralised algorithms can be applied to derive the next optimal point, and accordingly to solve the real optimisation problems. However, most of the existing works require additional optimisation toolboxes to solve the auxiliary sub-problems in centralised settings, e.g., [20, 27], which have struggled with the computation and communication complexity in network-connected systems.

There is an urgent need in developing distributed algorithms for multi-agent systems with expensive black-box functions. To date, the existing literature has not focused on this important class of distributed optimisation problems. In this paper, distributed approaches will be combined together with surrogate models to establish an efficient optimisation framework for costly functions. Nevertheless, three critical challenges arise in dealing with this class of problems. Firstly, establishing a surrogate that can approximate the true functions and facilitate the development of distributed algorithms

is difficult. Secondly, developing a suitable compromising strategy that balances between exploitation and exploration over network is challenging. Finally, designing the cooperative algorithm taking the above two objectives into consideration is non-trivial mainly owing to the problem transformation, and the complexity of convergence analysis.

In this work, we first establish that the true functions can be interpolated by smooth and convex models under commonly-used assumptions. Because those surrogates are cheap to evaluate, it is feasible to develop gradient-based algorithms, which usually requires a large number of function and gradient evaluations. We introduce an adaptive constraint-based method that searches the next evaluation point with some distance from the previously evaluated points and with a low objective value in the current surrogate. Such a strategy is then transferred to a penalty-based optimisation problem, which is proved to be efficient to handle “soft” constraints in practice. By investigating the relationship between the centralised and distributed problems, we allocate each agent a local copy of the decision vector, regarded as the consensus variable. When the consensus of the local copies is achieved, the original problem can be solved in a distributed manner under connected graphs. The distributed projection-based algorithm is then proposed to find the most suitable decision vector for the next costly evaluation by compromising between the local and global searches. It is worth noting that the gradients used in the proposed algorithm are from the surrogates rather than the true black-box functions. Consensus tools and convex analysis are employed to establish the convergence of the proposed algorithm.

The significance of our study is to establish a surrogate-assisted optimisation framework for network-connected multi-agent systems, where the local cost functions and their derivatives are unavailable. Many engineering systems are too complex to be modelled by explicit functions, and evaluations of those systems are often time-consuming and costly. We utilise surrogate-assisted approaches to approximate the black-box functions by cheap surrogates such that efficient gradient-based distributed algorithms can be deployed. The proposed framework can be implemented for complex networks with potentially less communication and computation resources over connected communication graphs. The numerical example with detailed simulation results manifests the effectiveness of the proposed framework. Furthermore, the power charging problem of plug-in electric vehicles (PEVs) is formulated as a black-box optimisation problem in distributed settings, which reveals the practical significance of our work in real-world applications. This application has been very popular in recent studies [28, 29]; however, most of the existing works utilise theoretical models of the charging process, which might be unsuitable due to the complexity of the modelling and uncertainties of the real-time parameters.

The remainder of this paper is organised as follows. In Section 2, preliminaries and mathematical background will be introduced. Section 3 formulates the problem of interest and provides basic assumptions. In Section 4, the structure of our method and theoretical analysis will be presented. Simulation results are demonstrated in Section 5. Finally, Section 6 concludes this paper.

Notation: Let \mathbb{R} , \mathbb{R}^n and $\mathbb{R}^{n \times m}$ denote the real numbers, real vectors of n dimensions and real matrices of dimension $n \times m$. \mathbb{R}_+ and \mathbb{R}_- represent the real non-negative and non-positive numbers, respectively. For a series of vectors, a_1, \dots, a_N , let $\text{col}(a_1, \dots, a_N)$ be a column vector by stacking (a_1, \dots, a_N) on top of each other. The identity matrix of dimension n is represented by I_n . A^T denotes the transpose of a matrix $A \in \mathbb{R}^{n \times m}$, and $\lambda_i(A)$ denotes the i th eigenvalue of a square matrix $A \in \mathbb{R}^{n \times n}$. The Kronecker product is denoted by \otimes . For a vector $x \in \mathbb{R}^n$, $\|x\| = \sqrt{x^T x}$ denotes the Euclidean norm. In this paper, the subscript index k denotes a variable at the k th iteration, e.g., x^k and γ^k . The subscript i represents the variable of the i th agent in the network, e.g., x_i and Ω_i .

2 Preliminaries

This section presents background on graph theory and convex optimisation.

2.1 Graph Theory

The connection among a group of N agents is described by a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, which consists of N vertices, $\mathcal{V} = \{1, \dots, N\}$, denoting the agents in the network, and a set of edges, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, representing the communication links among the agents. The adjacent matrix is defined as $\mathcal{A} = [a_{ij}]_{N \times N}$, where $a_{ij} = 1$ if the edge pair $(i, j) \in \mathcal{E}$, and zero otherwise. The Laplacian matrix is denoted by $\mathcal{L} = [l_{ij}]_{N \times N}$, with $l_{ii} = \sum_{j=1, j \neq i}^N a_{ij}$ and $l_{ij} = -a_{ij}$, for $j \neq i$. The graph is undirected if $a_{ij} = a_{ji}$ for all $i, j \in \mathcal{V}$, and it is directed otherwise. In this paper, all communication graphs are assumed to be undirected. For a connected undirected graph, zero is a simple eigenvalue of the Laplacian matrix, and all other eigenvalues are positive.

2.2 Mathematical Background

In this subsection, preliminaries on convex analysis and projection mapping will be presented.

Given a compact and convex set $\Omega \subset \mathbb{R}^n$, the projection mapping, denoted by $\mathcal{P}_\Omega(x) : \mathbb{R}^n \rightarrow \Omega$, is defined as $\mathcal{P}_\Omega(x) = \arg\min_{y \in \Omega} \|x - y\|$. The normal cone at $x \in \Omega$ is defined by

$$N_\Omega(x) = \{\varphi \in \mathbb{R}^n \mid \varphi^T(y - x) \leq 0, \forall y \in \Omega\}. \quad (1)$$

The following lemma on projection operator will be used in the convergence analysis of the proposed framework. By using projected gradient of the surrogates, the local decision variable can be confined to the design space, as demonstrated later in Section 4.

Lemma 1 ([30]) *For a compact and convex set $\Omega \subset \mathbb{R}^n$, the following properties of the projection mapping hold:*

$$(x - \mathcal{P}_\Omega(x))^T(\mathcal{P}_\Omega(x) - y) \geq 0, \forall x \in \mathbb{R}^n, \text{ and } \forall y \in \Omega \quad (2)$$

$$\|\mathcal{P}_\Omega(x) - \mathcal{P}_\Omega(y)\| \leq \|x - y\|, \forall x, y \in \mathbb{R}^n. \quad (3)$$

Definition 1 (Smoothness and convexity [31]) *A differentiable function $h(x) : \Omega \rightarrow \mathbb{R}$ is α -strongly convex if, for any $a, b \in \Omega$,*

$$(a - b)^T(\nabla h(a) - \nabla h(b)) \geq \alpha \|a - b\|^2 \quad (4)$$

and is β -smooth if

$$\|\nabla h(a) - \nabla h(b)\| \leq \beta \|a - b\| \quad (5)$$

where α and β are nonnegative constants. Moreover, $\mathcal{H}_{\alpha, \beta}$ is defined as a set of α -strongly convex and β -smooth functions. In the case of non-strong convexity, it is denoted by $\mathcal{H}_{0, \beta}$, and the non-smooth set is denoted by $\mathcal{H}_{\alpha, \infty}$. In addition, $\mathcal{H}_{0, \infty}$ is equivalent to the first-order condition for convex functions.

Given a set of evaluation points, the conditions for establishing convex and smooth surrogates will be analysed, and therefore we introduce the definition of interpolation as follows.

Definition 2 ($\mathcal{H}_{\alpha, \beta}$ -interpolation [32]) *Let $\mathcal{K} = \{1, \dots, k\}$ denote a finite index set. For a given set of triples $\mathcal{S} = \{x^j, \nabla h^j(x^j), h^j(x^j)\}, \forall j \in \mathcal{K}$, with $x^j \in \mathbb{R}^n, \nabla h^j(x^j) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $h^j(x^j) : \mathbb{R}^n \rightarrow \mathbb{R}$, the set \mathcal{S} is said to be $\mathcal{H}_{\alpha, \beta}$ -interpolable iff there exists a function $h \in \mathcal{H}_{\alpha, \beta}$ such that $\nabla h^j(x^j) = \nabla h(x^j)$ and $h^j(x^j) = h(x^j)$ for any $j \in \mathcal{K}$.*

3 Problem Formulation

In this section, we formulate the distributed optimisation problems for expensive black-box functions using surrogate-based methods.

We consider a group of N agents, where each of them possesses a private cost function, denoted as $f_i(x_i)$ with $x_i \in \Omega_i$ being the optimisation variable in a compact and convex set. The global set constraint Ω is the intersection of all local ones, i.e., $\Omega = \bigcap_{i=1}^N \Omega_i \subset \mathbb{R}^n$, where Ω_i is privately known to agent i . It should be noted that Ω is used for analytical purpose, which does not have to be available to the distributed agents when running

the proposed algorithms. The goal of the network is to optimise the summation of the local cost functions, in a distributed way,

$$\min_{z \in \Omega} \sum_{i=1}^N f_i(z) \quad (6)$$

where the local cost functions are unknown. Note that we denote $x_i \in \Omega_i$ as the local action of agent i , and denote $z \in \Omega$ as the consensus action of the network.

The problem under consideration is to find the global optimum, in a distributed manner, without knowing the cost functions and their gradients. By exploring the structure of surrogate-assisted optimisation, we divide the problem into three stages: establishing a suitable surrogate, developing a feasible infill strategy and designing a cooperative optimisation algorithm. The following assumptions are made on the communication graph and the cost functions.

Assumption 1 *The communication graph is connected.*

Assumption 2 *The cost functions, $f_i(x_i), \forall i \in \mathcal{V}, x_i \in \Omega_i$ are differentiable and strongly convex. Furthermore, the gradients of the cost functions are β -smooth.*

Remark 1 *The communication graph is assumed to be connected such that the information flow among the distributed agents is complete, as stated in Assumption 1. This condition has been widely-used in the earlier distributed frameworks, e.g., [6, 9, 10]. Assumption 2 ensures that the cost functions are all strongly convex and smooth. Similar conditions are employed in distributed optimisation problems, for example, [9, 11]. In some works, additional assumptions on the smoothness of the gradients are required, e.g., [7, 10].*

4 Main Results

In this section, we first prove that the local samples can be interpolated as smooth and convex functions (see Theorem 1), and then provide an infill principle for function evaluations, which can improve the surrogate model gradually. Then, we propose a distributed algorithm to seek the optimal solution as in (36) further ahead, where the local cost functions are reformulated by taking the infill criteria into account. Finally, the overall structure of the framework will be presented.

4.1 Surrogate Condition and Formulation

Now, to show that the local samples can be interpolated by smooth and convex functions, the following preliminary results are needed.

Lemma 2 ([32, 33]) *For agent i in the network, the finite triple set $\{x_i^j, \nabla f_i(x_i^j), f_i(x_i^j)\}, \forall j \in \mathcal{K}$, where*

$x_i^j \in \Omega_i, \nabla f_i(x_i^j) : \mathbb{R}^n \rightarrow \mathbb{R}^n, f_i(x_i^j) : \mathbb{R}^n \rightarrow \mathbb{R}$, is $\mathcal{H}_{\alpha, \beta}$ -interpolable iff $\{x_i^j, \nabla f_i(x_i^j) - \alpha x_i^j, f_i(x_i^j) - \frac{\alpha}{2} \|x_i^j\|^2\}$ is $\mathcal{H}_{0, \beta - \alpha}$ -interpolable. Furthermore, $\{x_i^j, \nabla f_i(x_i^j), f_i(x_i^j)\}$ is $\mathcal{H}_{0, \beta}$ -interpolable iff $\{\nabla f_i(x_i^j), x_i^j, \nabla^T f_i(x_i^j) x_i^j - f_i(x_i^j)\}$ is $\mathcal{H}_{1/\beta, \infty}$ -interpolable.

We are ready to show the foundation for constructing smooth and convex surrogate.

Theorem 1 *Let Assumption 2 hold. If for any $a, b \in \Omega_i$, the following inequality holds*

$$f_i(a) \geq f_i(b) + \nabla f_i^T(b)(a - b) + \frac{\beta}{2(\beta - \alpha)} \left(\sqrt{\alpha}(a - b) - \frac{1}{\sqrt{\beta}}(\nabla f_i(a) - \nabla f_i(b)) \right)^2 \quad (7)$$

then, for each agent i , given a set of evaluation points $\{x_i^j\}, \forall j \in \mathcal{K}$ randomly generated from Ω_i , the triple set of $\{x_i^j, \nabla f_i(x_i^j), f_i(x_i^j)\}, \forall j \in \mathcal{K}$ is $\mathcal{H}_{\alpha, \beta}$ -interpolable.

Proof: A sufficient and necessary condition for convex interpolation, i.e., $\mathcal{H}_{0, \infty}$ -interpolation [32], is given by

$$f_i(a) \geq f_i(b) + \nabla^T f_i(b)(a - b), \quad \forall a, b \in \{x_i^j\}. \quad (8)$$

It can be obtained that the set

$$\left\{ \frac{\beta x_i^j}{\beta - \alpha} - \frac{\nabla f_i(x_i^j)}{\beta - \alpha}, \nabla f_i(x_i^j) - \alpha x_i^j, f_i(x_i^j) + \frac{\alpha \nabla^T f_i(x_i^j) x_i^j}{\beta - \alpha} - \frac{\alpha \beta \|x_i^j\|^2}{2(\beta - \alpha)} - \frac{\|\nabla f_i(x_i^j)\|^2}{2(\beta - \alpha)} \right\} \quad (9)$$

is $\mathcal{H}_{0, \infty}$ -interpolable, since the following inequality

$$\begin{aligned} f_i(a) &+ \frac{\alpha \nabla^T f_i(a) a}{\beta - \alpha} - \frac{\alpha \beta \|a\|^2}{2(\beta - \alpha)} - \frac{\|\nabla f_i(a)\|^2}{2(\beta - \alpha)} \\ &\geq f_i(b) + \frac{\alpha \nabla^T f_i(b) b}{\beta - \alpha} - \frac{\alpha \beta \|b\|^2}{2(\beta - \alpha)} - \frac{\|\nabla f_i(b)\|^2}{2(\beta - \alpha)} \\ &+ (\nabla f_i(b) - \alpha b)^T \left(\frac{\beta(a - b)}{\beta - \alpha} - \frac{\nabla f_i(a) - \nabla f_i(b)}{\beta - \alpha} \right) \end{aligned} \quad (10)$$

is equivalent to (7). The equivalence can be obtained by

$$\begin{aligned} f_i(a) &\geq f_i(b) + \nabla f_i(b)(a - b) + \frac{\alpha \nabla^T f_i(b)(a - b)}{\beta - \alpha} \\ &- \frac{\nabla f_i(a)^T \nabla f_i(b)}{\beta - \alpha} + \frac{\|\nabla f_i(b)\|^2}{\beta - \alpha} - \frac{\alpha \beta a^T b}{\beta - \alpha} \\ &+ \frac{\alpha \nabla^T f_i(a) b}{\beta - \alpha} + \frac{\alpha \beta \|b\|^2}{\beta - \alpha} - \frac{\alpha \nabla^T f_i(a) a}{\beta - \alpha} + \frac{\alpha \beta \|a\|^2}{2(\beta - \alpha)} \\ &+ \frac{\|\nabla f_i(a)\|^2}{2(\beta - \alpha)} - \frac{\alpha \beta \|b\|^2}{2(\beta - \alpha)} - \frac{\|\nabla f_i(b)\|^2}{2(\beta - \alpha)}. \end{aligned} \quad (11)$$

After some calculations, we have

$$\begin{aligned}
f_i(a) &\geq f_i(b) + \nabla f_i(b)(a-b) + \frac{\|\nabla f_i(a) - \nabla f_i(b)\|^2}{2(\beta - \alpha)} \\
&\quad + \frac{\alpha\beta\|a-b\|^2}{2(\beta - \alpha)} - \frac{\alpha(\nabla f_i(a) - \nabla f_i(b))^T(a-b)}{\beta - \alpha} \\
&\geq f_i(b) + \nabla f_i^T(b)(a-b) \\
&\quad + \frac{\beta}{2(\beta - \alpha)} \left(\alpha\|a-b\|^2 + \frac{1}{\beta} \|\nabla f_i(a) - \nabla f_i(b)\|^2 \right. \\
&\quad \left. - \frac{2\alpha}{\beta} (a-b)^T(\nabla f_i(a) - \nabla f_i(b)) \right). \tag{12}
\end{aligned}$$

Using the second result in Lemma 2, we obtain that

$$\begin{aligned}
&\left\{ \nabla f_i(x_i^j) - \alpha x_i^j, \frac{\beta x_i^j}{\beta - \alpha} - \frac{\nabla f_i(x_i^j)}{\beta - \alpha}, \right. \\
&\left. - f_i(x_i^j) + \frac{\beta \nabla^T f_i(x_i^j) x_i^j}{\beta - \alpha} - \frac{\alpha\beta\|x_i^j\|^2}{2(\beta - \alpha)} - \frac{\|\nabla f_i(x_i^j)\|^2}{2(\beta - \alpha)} \right\} \tag{13}
\end{aligned}$$

is $\mathcal{H}_{0,\infty}$ -interpolable, which is then equivalent to conclude that, by using the first result in Lemma 2,

$$\left\{ \nabla f_i(x_i^j) - \alpha x_i^j, x_i^j, -f_i(x_i^j) + \nabla^T f_i(x_i^j) x_i^j - \frac{\alpha}{2} \|x_i^j\|^2 \right\} \tag{14}$$

is $\mathcal{H}_{1/(\beta-\alpha),\infty}$ -interpolable. By similar arguments used in (13) and (14), it can be derived that $\{x_i^j, \nabla f_i(x_i^j) - \alpha x_i^j, f_i(x_i^j) - \frac{\alpha}{2} \|x_i^j\|^2\}$ is $\mathcal{H}_{0,\beta-\alpha}$ -interpolable, and $\{x_i^j, \nabla f_i(x_i^j), f_i(x_i^j)\}$ is $\mathcal{H}_{\alpha,\beta}$ -interpolable. This completes the proof. ■

Remark 2 Note that the interpolation function, denoted by \hat{f}_i , is not unique. A variety of interpolating methods can be used to generate the surrogate models, for example, radial basis function based neural networks [34], polynomial interpolation [23], and Kriging models [24]. From the proof of Lemma 1, it is straightforward to establish a strongly convex and smooth interpolating function in the form

$$\hat{f}_i^k(x_i) = \text{conv} \left(\bigcup_{j \in \mathcal{K}} \text{epi } h_i^j(x_i) \right) + \frac{\alpha}{2} \|x_i\|^2 \tag{15}$$

where $\text{conv}(\cdot)$ takes the convex hull (also referred as convex envelope) of the argument, epi denotes the epigraph operator, and

$$\begin{aligned}
h_i^j(x_i) &= f_i(x_i^j) + \nabla^T f_i(x_i^j)(x_i - x_i^j) \\
&\quad + \frac{1}{2(\beta - \alpha)} \|x_i - x_i^j\|^2. \tag{16}
\end{aligned}$$

Remark 3 The smoothness of the surrogate enables us to deploy the gradient-based algorithms in the following subsections, which derives the optimal solution at each iteration without solving auxiliary sub-problems. The inequality (7) has been used in some related works for surrogate-based optimisation problems, e.g., [33]. Strong convexity can generate more robust solutions with fast convergence speed, which makes the gradient-based optimisation more efficient [35]. It should be noted that non-smooth convex surrogates can be employed using non-smooth optimisation techniques [16].

To this end, distributed radial basis function (RBF) networks will be deployed to construct the surrogate for the expensive functions. For a set of k samples, $\mathcal{S}_i = \{(x_i^j; f(x_i^j)) | x_i^j \in \Omega_i, f(x_i^j) \in \mathbb{R}, \forall j \in \mathcal{K}\}$, the true functions can be approximated by a set of RBF models, given by

$$\hat{f}_i^k(x_i) = \sum_{j=1}^k \omega_i^j \phi_i(\|x_i - x_i^j\|) \tag{17}$$

where $\hat{f}_i^k(x_i)$ is the RBF interpolant of the i th agent, $\omega_i^j, \forall i \in \mathcal{V}, j \in \mathcal{K}$, are real coefficients, and $\phi_i(\|x_i - x_i^j\|)$ is the j th basis function centred at x_i^j . There are a variety of basis functions that can be employed, for example, linear, cubic and Gaussian basis functions [17]. In this paper, we employ the Gaussian function as the basis, given by

$$\phi_i(r) = \exp(-\gamma_i r^2), \gamma_i > 0 \tag{18}$$

since it possesses some useful properties, as will be examined later. Let $\omega_i = [\omega_i^1, \dots, \omega_i^k]^T$ and $\Phi_i(x_i) = [\phi_i(\|x_i - x_i^1\|), \dots, \phi_i(\|x_i - x_i^k\|)]^T$. Then, the interpolant of the i th agent can be written as

$$\hat{f}_i^k(x_i) = \omega_i^T \Phi_i(x_i). \tag{19}$$

Define the interpolation matrix $\mathfrak{G}_i \in \mathbb{R}^{k \times k}$ by

$$[\mathfrak{G}_i]_{rc} = \phi_i(\|x_i^r - x_i^c\|), \forall r, c \in \mathcal{K} \tag{20}$$

which is also referred as Gram matrix. Note that the RBF surrogates should generate the same output as the real samples, that is,

$$\mathfrak{G}_i^T \omega_i = F_i \tag{21}$$

with $F_i = [f_i(x_i^1), \dots, f_i(x_i^k)]^T$.

Theorem 2 Given a set of different samples \mathcal{S}_i , there exists a unique ω_i such that the RBF surrogate coincides with the true function of the i th agent at all samples, i.e., $\mathfrak{G}_i^T \omega_i = F_i$, using the basis function (18).

Proof: To show the uniqueness of ω_i , it is sufficient to show that the interpolation matrix \mathfrak{G}_i is invertible, which consequently yields $\omega_i = \mathfrak{G}_i^{-1} F_i$. Instead, we further prove that \mathfrak{G}_i is positive definite by using the Gaussian basis function (18), since this guarantees that ω_i can be safely computed by Cholesky decomposition [36]. For any $\xi \in \mathbb{R}^k$ and $\xi \neq \mathbf{0}$,

$$\begin{aligned} \xi^T \mathfrak{G}_i \xi &= \sum_{r \in \mathcal{K}} \sum_{c \in \mathcal{K}} \xi_r \xi_c \phi_i(\|x_i^r - x_i^c\|) \\ &= \sum_{r \in \mathcal{K}} \sum_{c \in \mathcal{K}} \xi_r \xi_c \exp(-\gamma_i(\|x_i^r - x_i^c\|^2)). \end{aligned} \quad (22)$$

Denoting a non-trivial measure $\varsigma(x_i)$, which has the mass ξ_r at $x_i = x_i^r, \forall r \in \mathcal{K}$, and has zero mass elsewhere over compact set Ω_i , then we have

$$\xi^T \mathfrak{G}_i \xi = \iint_{\Omega_i \times \Omega_i} \exp(-\gamma_i \|\mu - \nu\|^2) d\varsigma(\mu) d\varsigma(\nu). \quad (23)$$

Note that the right-hand side of (23) can be expressed as

$$\begin{aligned} &\iint_{\Omega_i \times \Omega_i} \exp(-\gamma_i \|\mu - \nu\|^2) d\varsigma(\mu) d\varsigma(\nu) \\ &= \iint_{\Omega_i \times \Omega_i} \pi^{-\frac{n}{2}} \left[\int_{\mathbb{R}^n} \exp(j2\gamma_i \tau(\mu - \nu)) \exp(-\|\tau\|^2) d\tau \right] \\ &\quad d\varsigma(\mu) d\varsigma(\nu) \end{aligned} \quad (24)$$

where we have used the fact that $\exp(-\|y\|^2) = \pi^{-n/2} \int_{\mathbb{R}^n} \exp(j2\tau y) \exp(-\|\tau\|^2) d\tau$, with j being the unit imaginary number. Using $|\exp(j2y\tau)| = 1, \forall y \in \Omega_i, \tau \in \mathbb{R}^n$, and Fubini's theorem, we obtain

$$\begin{aligned} \xi^T \mathfrak{G}_i \xi &= \pi^{-\frac{n}{2}} \int_{\mathbb{R}^n} \exp(-\|\tau\|^2) \int_{\Omega_i} \exp(j2\gamma_i \tau \mu) d\varsigma(\mu) \\ &\quad \int_{\Omega_i} \exp(-j2\gamma_i \tau \nu) d\varsigma(\nu) d\tau \\ &= \pi^{-\frac{n}{2}} \int_{\mathbb{R}^n} \exp(-\|\tau\|^2) \left| \int_{\Omega_i} \exp(j2\gamma_i \tau \mu) d\varsigma(\mu) \right|^2 d\tau. \end{aligned} \quad (25)$$

Since the integrand in above equation is positive whenever $\varsigma(\mu)$ is of non-zero mass, and zero otherwise, it follows that $\xi^T \mathfrak{G}_i \xi > 0$. Thus, there exists a unique ω_i such that the surrogate coincides with the true objective function. This completes the proof. \blacksquare

From the above result, it can be guaranteed that each local surrogate will generate the same output as the true function, and therefore, the global surrogate can mimic the network objective at the sampling instants. It follows from (17) and (18) that the surrogate generated by Gaussian basis function is smooth. According to [37], the model error at any $x_i \in \Omega_i$ can be estimated by

$$\varepsilon_i^2(x_i) = 1 - \Phi_i(x_i)^T \mathfrak{G}_i^{-1} \Phi_i(x_i). \quad (26)$$

Over a compact set, if the samples are dense in the design domain, the surrogate error will converge to zero [20].

4.2 Distributed Cooperative Algorithm with RBF Surrogate and Balanced Infill Strategy

In the previous subsections, we have proved that the cost functions can be interpolated by smooth and convex surrogate (Theorem 1), and then we have developed RBF networks for local surrogate construction (Theorem 2). In this subsection, we will propose an evaluation strategy that can guarantee the density of the evaluation sequence by reformulating the original problem as a constrained optimisation problem (Theorem 3). Then, we develop an equivalent formulation of the centralised problem, and then present a distributed algorithm that can be implemented by each agent with local interactions. Convergence of the proposed algorithm will be established in Theorem 5 further ahead.

In order to guarantee that the surrogate model converges as the number of iteration increases, an algorithm should be able to generate a sequence of evaluation points that is everywhere dense in Ω_i , as stated in the following lemma. This underlying principle has been established in the earlier work [38], and then further developed in [20, 27].

Lemma 3 ([38]) *An algorithm converges to the optimum for the continuous functions $\sum_{i=1}^N f_i(x_i)$ iff the decision sequence $\{x_i^j\}$ generated by the algorithm is dense in the compact set Ω .*

One of the commonly-used strategies to ensure the density of the sequence is to select the next evaluation point that maintains certain distance from the previously evaluated points [19, 25, 27]. Given a set of previously evaluated points, $\{x_i^j\}, \forall j \in \mathcal{K}$, the maximum distance in a compact set is

$$D_{\max,i} = \max_{d \in \Omega_i} \min_{j \in \mathcal{K}} \|d - x_i^j\|. \quad (27)$$

Therefore, the next evaluation point x_i^{k+1} can be chosen according to the distance from previous points, given by

$$\|x_i^{k+1} - x_i^j\| \geq \gamma^k D_{\max,i}, \quad \forall j \in \mathcal{K} \quad (28)$$

where $0 \leq \gamma^k \leq 1$. To save space and avoid notational complexity, x_i will be used to represent x_i^{k+1} in the rest of this paper. By introducing this inequality constraint, the surrogate-based optimisation problem becomes

$$\begin{aligned} &\min_{x_i \in \Omega_i, \forall i \in \mathcal{V}} \sum_{i=1}^N \hat{f}_i^k(x_i) \\ &\text{s.t. } \|x_i - x_i^j\| \geq \gamma^k D_{\max,i}, \quad \forall j \in \mathcal{K} \\ &\quad x_i = x_l, \forall i, l \in \mathcal{V}. \end{aligned} \quad (29)$$

Remark 4 The reformulated problem (29) selects the next evaluation point that minimises $\sum_{i=1}^N \hat{f}_i^k(x_i)$ with a distance at least $\gamma^k D_{\max,i}$ from the previously evaluated points. In the case of $\gamma^k = 0$, the problem (29) reduces to an unconstrained optimisation problem that searches the best optimiser of the current surrogate model $\sum_{i=1}^N \hat{f}_i^k(x_i)$, which can be understood as a pure exploitation strategy. When $\gamma^k = 1$, it becomes a pure exploration strategy that aims to find the next point having the largest distance from the previously evaluated points. It is worth noting that $\gamma^k = 1$ and 0 correspond to the global and local search, respectively. Various search mechanisms can be used according to the evaluation budget and evaluation costs, for example, constant pattern $\gamma^k = c \in [0, 1]$, decaying pattern $\gamma^k = \frac{1}{k+1}$ and cycle search [20, 27]. From the practical point of view, if function evaluation is extremely expensive, γ^k should be designed as fast decaying sequence, e.g., $\frac{1}{k+1}$, while it can be relaxed to slow decaying sequence, e.g., $\frac{1}{0.1k+1}$ or constant patterns, for less expensive cases. As a trade-off, if slow decaying sequence is used, surrogate fitting can be improved, and more function evaluations are required.

Now, we show the sequence of evaluation points, generated by the reformulated problem (29), is dense in Ω .

Theorem 3 The strictly increasing sequence of iterations $\{x_i^j\}$, generated by the optimisation problem (29) with $0 < \gamma^k \leq 1$, is everywhere dense in Ω , and therefore solving (29) yields the global minimum of $\sum_{i=1}^N \hat{f}_i^k(x_i)$ as $k \rightarrow \infty$.

Proof: We prove this theorem by seeking a contradiction. First, we assume that the generated sequence $\{x_i^j\}$ is not dense in Ω . It follows that there exist a positive constant $\epsilon > 0$ and a decision vector $\bar{z} \in \Omega$ such that the open ball $\mathcal{O} = \{o_i \mid \|o_i - \bar{z}\| < \epsilon\}$ does not contain any elements in the sequence $\{x_i^j\}$, that is,

$$\|x_i^j - \bar{z}\| \geq \epsilon, \forall j \in \mathcal{K}. \quad (30)$$

From the constraints in (29) and (27), it can be obtained that

$$\min_{m,n \in \mathcal{K}, m \neq n} \|x_i^m - x_i^n\| \geq \underline{\gamma} \max_{d \in \Omega} \min_{j \in \mathcal{K}} \|d - x_i^j\| \geq \underline{\gamma} \epsilon \quad (31)$$

where $\underline{\gamma} = \min_{j \in \mathcal{K}} \gamma^j$. This indicates that any two elements in the sequence satisfy

$$\|x_i^m - x_i^n\| \geq \underline{\gamma} \epsilon, \forall m, n \in \mathcal{K}, m \neq n. \quad (32)$$

As the number of evaluation k goes to infinity, the elements in the sequence cannot be encompassed into a compact set since any two elements are of certain distance shown in (32), which leads to a contradiction with

the compactness of Ω . Therefore, the iteration sequence $\{x_i^j\}$ is dense in Ω . Then, the rest of the results can be concluded by invoking Lemma 3. This completes the proof. ■

Remark 5 Theorem 3 illustrates that, with an unlimited number of function evaluations, exact optimal solution can be obtained by solving the constrained optimisation problem. This is a common practice and a basic requirement when designing an algorithm for expensive optimisation problems, as shown in many pioneering works [27, 38]. A suitable selection of parameter γ^k can be formed by employing time-varying γ^k that is initially set as some value close to 1 and decreases to 0 when approaching to the limit of function evaluation [17, 25]. Hence, the designer should balance the trade-off between the optimal error and the number of evaluations required.

To facilitate the distributed algorithm design in the following subsections, we introduce a penalty-based method in [39] to handle the inequality constraints in (29). Let $g_i(x_i) = \gamma^k D_{\max,i} - \|x_i - x_i^j\|, \forall j \in \mathcal{K}$, at the k th iteration. The smoothed penalty function can be defined as

$$P(g_i) = \begin{cases} 0, & \text{if } g_i \leq 0 \\ \frac{g_i^2}{2\delta}, & \text{if } 0 \leq g_i \leq \delta \\ g_i - \frac{\delta}{2}, & \text{if } g_i \geq \delta \end{cases} \quad (33)$$

with δ being a positive constant.

Remark 6 It is worth mentioning that the smoothed penalty is proved to be efficient and robust in applications that are with “soft” constraints as discussed in [39]. In this paper, the inequality introduced in (29) does not have to be satisfied at all iterations. Therefore, the penalty-based method can be efficiently implemented in practice.

Then, the local cost function can be formulated as

$$\bar{f}_i^k(x_i) = \hat{f}_i^k(x_i) + \kappa_i P(g_i(x_i)) \quad (34)$$

where κ_i denotes the weight of the penalty function. Under Assumption 1, the problem is equivalent to

$$\begin{aligned} \min_{x_i \in \Omega_i, \forall i \in \mathcal{V}} \quad & \sum_{i=1}^N \bar{f}_i^k(x_i) \\ \text{s.t.} \quad & (\mathcal{L} \otimes I_n)X = \mathbf{0} \end{aligned} \quad (35)$$

where $X = \text{col}(x_1, \dots, x_N)$. Note that the reformulated functions combine both the surrogate model and the in-fill criteria.

Remark 7 In conventional works on surrogate-based optimisation, e.g., [27, 34], it is usually assumed that centralised optimisation toolboxes that can be used to

find the solution of the auxiliary problem (35) are directly available. In this paper, we will further develop a distributed algorithm to solve (35) so as to facilitate a fully distributed operation.

The distributed algorithm is designed as

$$\begin{aligned}\dot{x}_i &= \mathcal{P}_{\Omega_i} \left[x_i - \nabla \bar{f}_i^k(x_i) \right. \\ &\quad \left. - \sum_{j=1}^N a_{ij}(x_i - x_j) - \sum_{j=1}^N a_{ij}(\lambda_i - \lambda_j) \right] - x_i \quad (36) \\ \dot{\lambda}_i &= \tau \sum_{j=1}^N a_{ij}(x_i - x_j)\end{aligned}$$

where $\lambda_i \in \mathbb{R}^n$ is the Lagrangian multiplier of agent i , and $\tau > 0$ is to be designed. To analyse the network behaviour, we concatenate the local variables to form the augmented ones. Let $F^k(X) = \text{col}(\bar{f}_1^k(x_1), \dots, \bar{f}_N^k(x_N))$, $\nabla F^k(X) = \text{col}(\nabla \bar{f}_1^k(x_1), \dots, \nabla \bar{f}_N^k(x_N))$, $\Lambda = \text{col}(\lambda_1, \dots, \lambda_N)$, and $\bar{\Omega} = \Omega_1 \times \dots \times \Omega_N$. The compact form of (36) can be written as

$$\begin{aligned}\dot{X} &= \mathcal{P}_{\bar{\Omega}} [X - \nabla F^k(X) \\ &\quad - (\mathcal{L} \otimes I_n)X - (\mathcal{L} \otimes I_n)\Lambda] - X \quad (37) \\ \dot{\Lambda} &= \tau(\mathcal{L} \otimes I_n)X.\end{aligned}$$

Remark 8 The proposed algorithm in (36) takes advantage of the saddle-point dynamics [9, 10] and the projection operator [11, 16]. The projection operator guarantees the decision variable is always in the compact set Ω , which consequently ensures the density of the sequence as $k \rightarrow \infty$, due to the adoption of the penalty functions. All the local copies of the decision vector will converge to the same optimum that minimises the sum of local cost functions by using the distributed gradient-based algorithm in (36). Some related schemes dealing with different problems have been reported in [7, 9, 10, 35].

Remark 9 In the existing literatures, various approaches have been developed to solve distributed optimisation problems, e.g., the alternating direction method of multipliers (ADMM)/Douglas-Rachford splitting method [40, 41] and distributed proximal gradient (DPG) [42]. ADMM-based approaches require more auxiliary dynamics, usually consisting of two primal-variable updates and one dual-variable update that are performed in an alternating fashion. Such alternating methods can lead to slow convergence rate and synchronisation issues [43]. Moreover, ADMM and DPG algorithms introduce additional sub-optimisation problems, as they have to deploy extra tools to solve argmin and proximal operators (see [40, 42, 43]). The proposed algorithm in this paper employs a projected gradient to deal with the constraints of the design domain, which is different from the vanilla gradient used in related works [3, 7, 40, 42, 44]. By using

the projection-based algorithm, it is guaranteed that the next evaluation point is confined to the feasible design space Ω , even though such a global set is unavailable to the local agents. Furthermore, algorithm (36) only requires to exchange two variables (λ_j and x_j) with neighbours, while some algorithms rely on additional information exchange, e.g., local gradient [45], and the proposed method does not need to solve any sub-optimisation problems.

In the following, we show that the equilibrium of the proposed algorithm (36) is the solution to the reformulated problem in (35), and then we prove the convergence of the algorithm.

Theorem 4 Let Assumptions 1 and 2 hold. Under inequality (7), the equilibrium point of the proposed algorithm (36) is an optimal solution to the reformulated problem (35).

Proof: The equilibrium of (37), denoted by $(\tilde{X}, \tilde{\Lambda})$, can be obtained from

$$\begin{aligned}\mathbf{0} &= \mathcal{P}_{\bar{\Omega}} [\tilde{X} - \nabla F^k(\tilde{X}) \\ &\quad - (\mathcal{L} \otimes I_n)\tilde{X} - (\mathcal{L} \otimes I_n)\tilde{\Lambda}] - \tilde{X} \quad (38) \\ \mathbf{0} &= \tau(\mathcal{L} \otimes I_n)\tilde{X}.\end{aligned}$$

For connected graphs, the null-space of the Laplacian matrix \mathcal{L} is $\kappa \mathbf{1}_N$, $\forall \kappa \in \mathbb{R}$, i.e., the consensus space. Thus, the second equation in (38), $\tau(\mathcal{L} \otimes I_n)\tilde{X} = \mathbf{0}$, holds if and only if $\tilde{x}_i = \tilde{x}_j \triangleq x^*$, $\forall i, j \in \mathcal{V}$, $x^* \in \mathbb{R}^n$. Then, by substituting the result into the first equation and decomposing the network variables, we have

$$\begin{aligned}\tilde{x}_i &= x^*, \forall i \in \mathcal{V}, x^* \in \mathbb{R}^n \\ &\quad - \nabla \bar{f}_i^k(x^*) - \sum_{i=1}^N a_{ij}(\tilde{\lambda}_i - \tilde{\lambda}_j) \in N_{\Omega}(x^*).\end{aligned} \quad (39)$$

Comparing (39) with the Karush-Kuhn-Tucker (KKT) conditions in [46], it can be concluded that the equilibrium $(\tilde{X}, \tilde{\Lambda})$ is an optimal solution to (35). This completes the proof. ■

Now, we present the convergence analysis of the proposed algorithm.

Theorem 5 Suppose Assumptions 1 and 2 hold. The distributed algorithm (36), with $\tau < \frac{1}{\lambda_N(\mathcal{L})}$, solves the reformulated problem (35), under inequality (7).

Proof: The Lyapunov candidate is proposed as

$$V = V_1 + V_2 + V_3 \quad (40)$$

where

$$\begin{aligned}
V_1 &= \frac{1}{2}(X - \tilde{X})^T(X - \tilde{X}) \\
&\quad + \frac{1}{2}(\Lambda - \tilde{\Lambda})^T \left[\left(\frac{1}{\tau} I_N - \mathcal{L} \right) \otimes I_n \right] (\Lambda - \tilde{\Lambda}) \\
V_2 &= \mathbf{1}_N^T F^k(X) - \mathbf{1}_N^T F^k(\tilde{X}) - \nabla^T F^k(\tilde{X})(X - \tilde{X}) \\
V_3 &= \frac{1}{2}(X + \Lambda)^T(\mathcal{L} \otimes I_n)(X + \Lambda) \\
&\quad - \frac{1}{2}(\tilde{X} + \tilde{\Lambda})^T(\mathcal{L} \otimes I_n)(\tilde{X} + \tilde{\Lambda}) \\
&\quad - (\tilde{X} + \tilde{\Lambda})^T(\mathcal{L} \otimes I_n)(X - \tilde{X} + \Lambda - \tilde{\Lambda}).
\end{aligned} \tag{41}$$

Due to the condition $\tau < \frac{1}{\lambda_N(\mathcal{L})}$, we have $\frac{1}{\tau} I_N - \mathcal{L}$ is positive definite, and therefore $V_1 \geq 0$. Note that we assume $\mathbf{1}_N^T F^k(X)$ and $\frac{1}{2}(X + \Lambda)^T(\mathcal{L} \otimes I_n)(X + \Lambda)$ are convex. Thus, it is clear that $V_2, V_3 \geq 0$ by using the first-order property of convexity. The time derivative of V_1 along (37) is obtained as

$$\begin{aligned}
\dot{V}_1 &= (X - \tilde{X})^T \dot{X} + (\Lambda - \tilde{\Lambda})^T \left[\left(\frac{1}{\tau} I_N - \mathcal{L} \right) \otimes I_n \right] \dot{\Lambda} \\
&= (X - \tilde{X})^T (\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - X) \\
&\quad + (\Lambda - \tilde{\Lambda})^T \left[\left(\frac{1}{\tau} I_N - \mathcal{L} \right) \otimes I_n \right] \dot{\Lambda}
\end{aligned} \tag{42}$$

with $\Psi = \nabla F^k(X) + (\mathcal{L} \otimes I_n)X + (\mathcal{L} \otimes I_n)\Lambda$. The derivative of V_2 is calculated as

$$\dot{V}_2 = [\nabla F^k(X) - \nabla F^k(\tilde{X})]^T (\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - X). \tag{43}$$

For V_3 , we have

$$\begin{aligned}
\dot{V}_3 &= (X + \Lambda)^T(\mathcal{L} \otimes I_n)(\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - X) \\
&\quad - (\tilde{X} + \tilde{\Lambda})^T(\mathcal{L} \otimes I_n)(\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - X) \\
&\quad + (X + \Lambda)^T(\mathcal{L} \otimes I_n)\dot{\Lambda} - (\tilde{X} + \tilde{\Lambda})^T(\mathcal{L} \otimes I_n)\dot{\Lambda}.
\end{aligned} \tag{44}$$

Denote $\Psi^* = \nabla F^k(\tilde{X}) + (\mathcal{L} \otimes I_n)\tilde{X} + (\mathcal{L} \otimes I_n)\tilde{\Lambda}$. Then, adding \dot{V}_1, \dot{V}_2 and \dot{V}_3 together yields

$$\begin{aligned}
\dot{V} &= \Psi^T(\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - X) - \Psi^{*T}(\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - X) \\
&\quad + (X - \tilde{X})^T(\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - X) \\
&\quad + (\Lambda - \tilde{\Lambda})^T \left[\left(\frac{1}{\tau} I_N - \mathcal{L} \right) \otimes I_n \right] \dot{\Lambda} \\
&\quad + (X - \tilde{X})^T(\mathcal{L} \otimes I_n)\dot{\Lambda} + (\Lambda - \tilde{\Lambda})^T(\mathcal{L} \otimes I_n)\dot{\Lambda}.
\end{aligned} \tag{45}$$

To take the advantage of the properties of projection

operators, we regroup the items in (45) as

$$\begin{aligned}
\dot{V} &= (\Psi - X + \mathcal{P}_{\tilde{\Omega}}[X - \Psi])^T(\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - \tilde{X}) \\
&\quad - (\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - X)^T(\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - \tilde{X}) \\
&\quad - \Psi^T(X - \tilde{X}) - \Psi^{*T}(\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - \tilde{X}) \\
&\quad + \Psi^{*T}(X - \tilde{X}) - (X - \tilde{X})^T(\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - X) \\
&\quad + \frac{1}{\tau}(\Lambda - \tilde{\Lambda})^T I_{nN} \dot{\Lambda} + (X - \tilde{X})^T(\mathcal{L} \otimes I_n) \dot{\Lambda}.
\end{aligned} \tag{46}$$

Since $\dot{\Lambda} = \tau(\mathcal{L} \otimes I_n)X$, we have

$$\begin{aligned}
\dot{V} &= \underbrace{(\Psi - X + \mathcal{P}_{\tilde{\Omega}}[X - \Psi])^T(\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - \tilde{X})}_{W_1} \\
&\quad - (\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - X)^T(\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - X) \\
&\quad - \underbrace{\Psi^{*T}(\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - \tilde{X})}_{W_2} - \underbrace{(\Psi - \Psi^*)^T(X - \tilde{X})}_{W_3} \\
&\quad + \underbrace{(\Lambda - \tilde{\Lambda})^T(\mathcal{L} \otimes I_n)X + \tau(X - \tilde{X})^T(\mathcal{L}^2 \otimes I_n)X}_{W_4}.
\end{aligned} \tag{47}$$

It follows from the property of projection mapping in Lemma 1 that $W_1 \leq 0$. According to the definition of $\Psi^* = \nabla F^k(\tilde{X}) + (\mathcal{L} \otimes I_n)\tilde{X} + (\mathcal{L} \otimes I_n)\tilde{\Lambda}$ and the optimality condition in (39), we can obtain $\Psi^* = \nabla F^k(\tilde{X}) + (\mathcal{L} \otimes I_n)\tilde{\Lambda} \in -N_{\tilde{\Omega}}(\tilde{X})$ due to $(\mathcal{L} \otimes I_n)\tilde{X} = 0$. Then,

$$\begin{aligned}
W_2 &= -\Psi^{*T}(\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - \tilde{X}) \\
&\in \{\varsigma^T(\mathcal{P}_{\tilde{\Omega}}[X - \Psi] - \tilde{X}) \mid \varsigma \in N_{\tilde{\Omega}}(\tilde{X})\}.
\end{aligned} \tag{48}$$

Invoking the definition of normal cone at \tilde{X} as in (1), we have $W_2 \leq 0$ by using $\mathcal{P}_{\tilde{\Omega}}[X - \Psi] \in \tilde{\Omega}$. Now, decomposing W_3 gives

$$\begin{aligned}
W_3 &= -[\nabla F^k(X) - \nabla F^k(\tilde{X})]^T(X - \tilde{X}) \\
&\quad - (X - \tilde{X})^T(\mathcal{L} \otimes I_n)(X - \tilde{X}) \\
&\quad - (\Lambda - \tilde{\Lambda})^T(\mathcal{L} \otimes I_n)(X - \tilde{X}).
\end{aligned} \tag{49}$$

Notice that, for connected graphs, we have $(\mathcal{L} \otimes I_n)\tilde{X} = \mathbf{0}$. Then, combining W_3 and W_4 results in

$$\begin{aligned}
W_3 + W_4 &= -[\nabla F^k(X) - \nabla F^k(\tilde{X})]^T(X - \tilde{X}) \\
&\quad + \tau(X - \tilde{X})^T(\mathcal{L}^2 \otimes I_n)(X - \tilde{X}) \\
&\quad - (X - \tilde{X})^T(\mathcal{L} \otimes I_n)(X - \tilde{X}) \\
&\leq - (X - \tilde{X})^T[(\mathcal{L} - \tau\mathcal{L}^2) \otimes I_n](X - \tilde{X})
\end{aligned} \tag{50}$$

where $-[\nabla F^k(X) - \nabla F^k(\tilde{X})]^T(X - \tilde{X}) \leq 0$ has been utilised to derive the inequality. Finally, it can be obtained that $W_3 + W_4 \leq 0$, since $\mathcal{L} - \tau\mathcal{L}^2$ is negative semidefinite. Thus, we have $\dot{V} \leq 0$, and $\dot{V} = 0$ iff the optimal solution is achieved. ■

Remark 10 From Theorems 4 and 5, it is proved that algorithm (36) converges to the optimal solution of the auxiliary problem (35). Then, such a solution is selected as the next evaluation point, i.e., $x_i^{k+1} = x^*$, which achieves an appropriate balance between exploration and exploitation.

4.3 Implementation Structure

To this end, we are ready to summarise the implementation structure of our distributed solution to the expensive black-box optimisation with surrogate models. Algorithm 1 illustrates the detailed procedures of the proposed framework.

Algorithm 1 The Implementation Structure.

Initialisation:

for each agent $i \in \mathcal{V}$

1. select an initial set of points, $\mathcal{S}^0 = \{x_i^1, \dots, x_i^m\}$;
2. evaluate the functions at the specified points \mathcal{S}^0 ;
3. construct the initial surrogate model $\hat{f}_i^m(x_i)$.

Iteration:

set $k := k + 1$, for $i \in \mathcal{V}$

4. update the surrogate $\hat{f}_i^k(x_i^k)$ with the penalty function and a time-varying γ^k ;
5. run the distributed algorithm (36) to drive x_i^{k+1} by communicating with its neighbours \mathcal{N}_i ;
6. evaluate the local function at x_i^{k+1} ;
7. update iteration information $\mathcal{S}^{k+1} := \mathcal{S}^k \cup \{x_i^{k+1}\}$.

End if termination condition is satisfied or iteration budget is approached.

In Section 4.1, the conditions for constructing smooth and convex models are analysed, and then distributed RBF networks are utilised to establish cheap surrogate such that gradient-based algorithms can be devised to solve the auxiliary problem. Because we have to optimise the surrogate functions and also have to improve the surrogate fitness, an effective infill strategy is needed to balance the exploration and exploitation. As presented in Section 4.2, we develop a helpful distance-based infill method, which can guarantee the density of the decision sequence, and enable the designer to choose the balancing parameters. Then, to avoid using additional optimisation toolboxes, we propose a fully distributed algorithm solving the auxiliary problem. Such an algorithm takes advantage of the cheap surrogates by using fast gradient descent method, and uses the projection operator to guarantee all local actions are aligned with the global decision space, even if this global information is unknown to the local agents. This is particularly important, as the local surrogate should be feasible in the global decision set. Consequently, the obtained solution can be used for the surrogate update in next iteration.

Combining previous results in Theorems 1-5, the following corollary on the overall convergence of Algorithm 1 can be obtained. From Theorem 2, the proposed RBF surrogate (17) with basis function (18) can generate exact true function, i.e., $\varepsilon_i^2(x_i) \rightarrow 0$, as $k \rightarrow \infty$ since the decision sequence is dense over the design domain, which is guaranteed due to the employment of balanced infill strategy and distributed algorithm in (36) as proved in Theorems 3-5.

Corollary 1 Let the conditions in Theorems 1-5 hold. The decision sequence $\{x_i^j\}$ obtained from Algorithm 1 with the RBF surrogate (17) converges to the optimal solution of the original problem (6) as $k \rightarrow \infty$.

The combined scheme, as shown in Algorithm 1, contains two parts: initialisation and iteration. By selecting a set of initial evaluation points, a rough surrogate model can be generated based on the RBF networks. Then, algorithm (36) can be employed to obtain the next evaluation point, which utilises a balanced infill strategy integrated in the penalty functions. This new evaluation point will then be used to improve the surrogate for better interpolation performance. According to Corollary 1, the combined algorithm will converge to the optimal solution of the original problem when k goes to infinity, since the generated sequence from the algorithm is dense in the design space and the RBF networks converge to the true functions.

Remark 11 Compared with the centralised approaches, the distributed algorithm possesses several significant advantages, for example, potentially less communication, less computation requirement and robustness to network uncertainties. All the procedures can be performed by the distributed individuals using local interactions with their adjacent neighbours, by which private information can be well-protected. In algorithm (36), the local information required by agent i includes the gradient of the augmented surrogate function $\nabla \hat{f}_i^k(x_i)$ and the auxiliary variable λ_i , and the shared information from its neighbours includes the decision vector x_j and the Lagrangian multiplier λ_j , for all neighbours $j \in \mathcal{N}_i$.

Remark 12 The proposed framework is different from the existing studies on distributed optimisation [7, 9–11, 16, 47] which assume the cost functions are explicitly known and can be evaluated efficiently. On the other hand, centralised methods are proposed in the early literature to solve expensive optimisation [20, 27], which does not consider network cooperation and communication. In this paper, we propose a useful framework that integrates network communication and surrogate models together to solve distributed expensive optimisation problems.

Remark 13 Since solving the surrogate-based optimisation problem as in (35) is computationally affordable, multiple runs of the distributed algorithm (36) can be carried out simultaneously to deal with the non-convexity,

from which the optimal solution can be selected [27]. Therefore, convexity has been utilised in the theoretical analysis such that the convergence of the proposed scheme can be established. From practical point of view, local minimum can also be used in the decision update, because it introduces another source of exploration that can be helpful for finding the global optimum.

5 Simulation Study

In this section, we provide two examples with detailed simulation results to demonstrate the effectiveness of our work.

5.1 Numerical Example

We consider a network of 6 agents that aim to solve

$$\begin{aligned} \min_{x_i \in [-10, 10]} \quad & \sum_{i=1}^6 f_i(x_i) \\ \text{s.t.} \quad & x_i = x_j, \forall i, j = 1, \dots, 6 \end{aligned}$$

$$\begin{aligned} f_1(x_1) &= (x_1 - 2)^2 & f_2(x_2) &= 0.5x_2^2 + x_2 \\ f_3(x_3) &= \frac{x_3^2}{\sqrt{x_3^2 + 7}} + 1.3x_3^2 & f_4(x_4) &= \ln(x_4^2 + 1) + 0.7x_4^2 \\ f_5(x_5) &= 0.8(x_5 - 3)^2 & f_6(x_6) &= 0.5x_6^2 - 4. \end{aligned}$$

It is worth emphasising that the cost functions are assumed to be unknown, and function calls should be performed at specified points. In practice, the cost values are generated by black-box functions, which are usually expensive to evaluate. Here, the functions are given for analytical purposes only. The Laplacian matrix of the connected graph is assumed as

$$\mathcal{L} = \begin{bmatrix} 3 & -1 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & -1 & 3 \end{bmatrix}$$

with $\lambda_N(\mathcal{L}) = 5$. With those graph and function settings, it can be easily verified that Assumptions 1 and 2 are satisfied. The initial evaluation points are set as $\mathcal{S}^0 = \{-8.7; -6.1; -4.2; 5.9; 6.6; 9.6\}$, based on which a rough surrogate can be constructed.

We assume the total number of function evaluation is limited to 20 times. Then, the sequence of x_i^k is obtained by implementing Algorithm 1 as shown in Fig. 1. The rest evaluation points of the sequence are obtained as

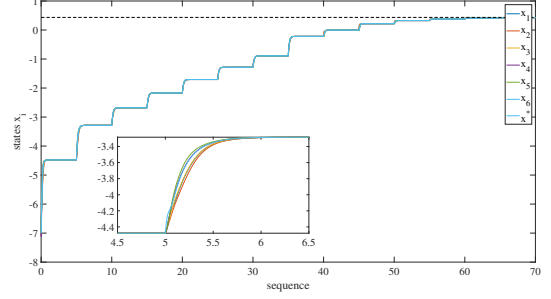


Fig. 1. The sequence of the decision vector.

$\{-4.479; -3.284; -2.680; -2.169; -1.706; -1.283; -0.892; -0.219; -0.003; 0.218; 0.326; 0.380; 0.413; 0.425\}$. To obtain those points, the distributed cooperative algorithm in (36) is utilised. The time interval between two contiguous iterations is set as 5 seconds, which is sufficient for the algorithm to converge as illustrated in Fig. 1. The optimal solution of the true cost functions is calculated as $z^* = 0.434$. The difference between the real optimal solution and the current solution is displayed in Fig. 2. It can be seen that the optimality error decreases as the number of evaluation increases. As the objective functions are expensive black-box, it is desirable to balance the evaluation cost and the design precision. To achieve better performance, a fast-decaying $\gamma^k = \frac{1}{k+1}$ is implemented, which initially explores globally over the decision space, then quickly settles to exploit the current surrogate to search the optimum. This choice can significantly reduce the number of evaluations at the expense of precision. For most of engineering applications, the trade-off is necessary and feasible. For comparison purpose, we have implemented a slower decaying sequence, $\gamma^k = \frac{1}{0.2k+1}$, and the simulation result is presented in Fig. 2. In this case, the agents are given more exploration chances, and consequently the optimal error is relatively larger at the early stage. Because of the initial exploration, more precise surrogates are established, which is helpful for reducing the optimal error after a number of iterations, as demonstrated in the enlarged profile in Fig. 2.

To show the change of the surrogate models, we present the initial and final surrogate models in Fig. 3 by taking agent 5 as an example. Initially, due to the lack of samples, there are some errors between the true function and the surrogate in the domain where few samples have been collected. After a number of iterations, the interpolation performance is significantly improved, as shown in Fig. 4. The error of the surrogate is obtained by integrating $|f_5 - \hat{f}_5^k|$ over the decision space $[-10, 10]$. Because the number of black-box evaluation is quite limited, we have chosen a fast-decaying γ^k that searches over the promising decision space (local search).

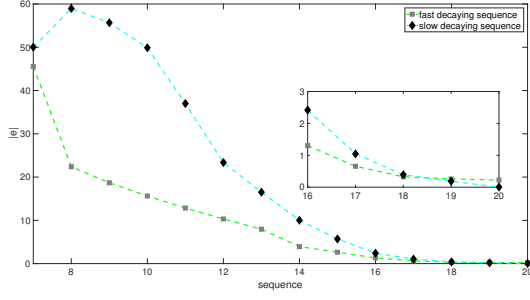


Fig. 2. The optimal error $|e| = \sum_{i=1}^6 |x_i - x^*|$ at the k th iteration using different sequences of γ^k .

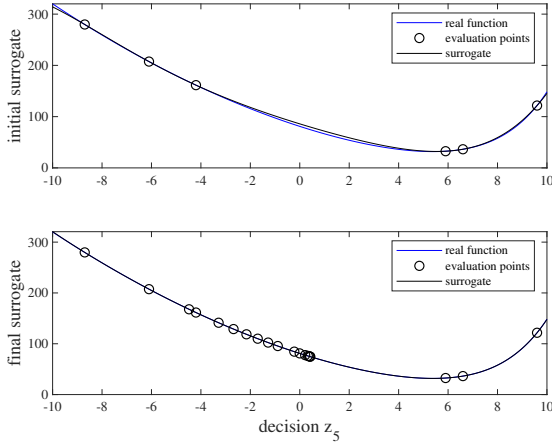


Fig. 3. The initial and final surrogate models of agent 5.

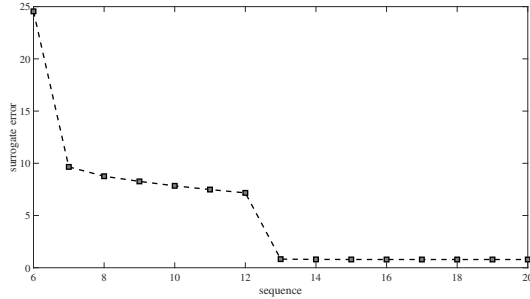


Fig. 4. The surrogate error of agent 5.

5.2 Practical Application

In this subsection, the proposed framework is applied to the optimal charging problem of plug-in electric vehicles. We consider a group of PEVs at a charging station aiming to optimise their charging power via local interactions with their adjacent neighbours. Each vehicle maintains a local cost function consisting of a series of system parameters, including the charging current, internal resistance, state of charge, and the charging price, etc. Modelling of the problem has been studied in recent works, e.g., [28, 29]. Nevertheless, those formula-

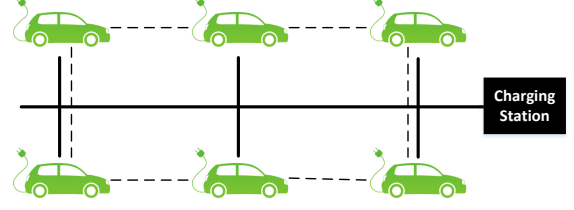


Fig. 5. Connections of the PEVs where the solid lines represent the power network and the dotted lines denote the communication network.

tions are heavily dependent on uncertain or time-varying parameters, which are difficult to be modelled precisely. Therefore, surrogate-assisted methods are applicable to be implemented by performing black-box evaluations. In each vehicle, a local control centre is equipped with a performance indicator, communication and computation units, which can deploy the proposed framework in distributed settings. The objective of this task is to obtain the optimal charging power for multiple parallel-connected vehicles via stimulating the black-box performance functions.

In this simulation, the black-box functions are adopted from [28], formulated as

$$f_i(P_{EV}) = p_i \left(\frac{P_{EV}}{R_i} - \frac{V_{o,i} + 2R_i I_i^{\text{ref}}}{2R_i^2} \sqrt{4R_i P_{EV} + V_{o,i}^2} \right) \\ 0 \leq P_{EV} \leq P_{EV}^{\max}$$

where P_{EV} is the charging power to be optimised; p_i denotes the priority weight of the i th EV; R_i is the equivalent internal resistance; $V_{o,i}$ represents the open-circuit voltage; and I_i^{ref} is the desired charging current reference. According to the formulation of the cost function $f_i(P_{EV})$, the objective of each PEV is to minimise the derivation from the desired charging current subject to a set of system constraints. The cost functions are assumed to be unavailable, and objective values can only be generated by function calls to the performance indicator inside each PEV. The connections among the electric vehicles are illustrated in Fig. 5, and the simulation parameters are adopted from [28], by which Assumptions 1 and 2 are fulfilled.

The optimality error is illustrated in Fig. 6, which shows a decreasing profile as the number of function evaluations increases. It is clear that the optimality error decreases significantly at the first few iterations due to the improvement of the surrogate. In Fig. 7, the convergence of the distributed algorithm (36) is manifested for randomly generated initial values at the 160th function evaluation, where the convergence time is about 5 seconds. The advantages of using the proposed framework include that it does not require the modelling of the charging process; and it is straightforward to be deployed in large-scale networks.

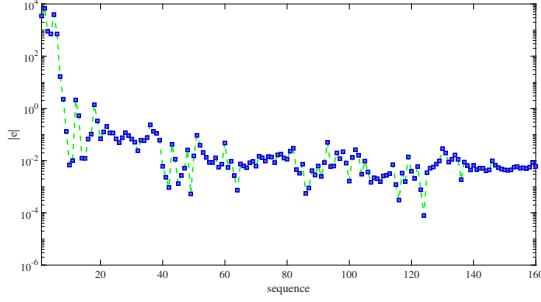


Fig. 6. The optimality error $|e| = \sum_{i=1}^6 |x_i - x^*|$ at the k th iteration.

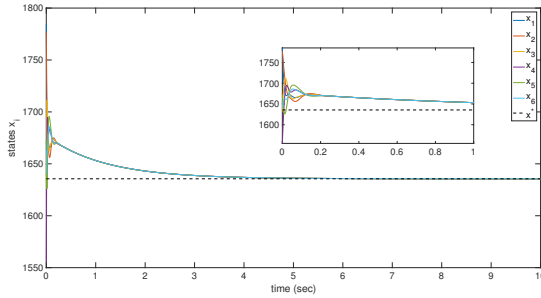


Fig. 7. Convergence of the distributed decision vector with random initial values at the 160th iteration.

6 Conclusion

This paper has studied an important class of distributed optimisation problems with unknown cost functions over connected networks. A novel framework is developed to solve the problems by using surrogate-assisted methods. We have proved that smooth and convex models can be deployed to approximate the true cost functions with commonly-used assumptions. Then, a compromise between the exploration and exploitation has been achieved by using distance-based infill criteria. The density of the decision sequence generated by the infill strategy has been demonstrated. The distributed algorithm has been proposed with a projection operator to solve the auxiliary problems with the compact set constraints. The convergence of the algorithm is investigated via exploring the consensus dynamics and convex properties. The numerical and practical examples have been presented to verify the theoretical development and demonstrate the significance of the our framework in engineering practices.

References

- [1] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.
- [2] Z. Li, Z. Li, and Z. Ding, "Distributed generalized Nash equilibrium seeking and its application to femtocell networks," *IEEE Transactions on Cybernetics*, to be published, doi: 10.1109/TCYB.2020.3004635.
- [3] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.
- [4] B. Liu and Z. Ding, "Distributed heuristic adaptive neural networks with variance reduction in switching graphs," *IEEE Transactions on Cybernetics*, to be published, doi: 10.1109/TCYB.2019.2956291.
- [5] Z. Dong, Z. Li, Y. Dong, S. Jiang, and Z. Ding, "Fully-distributed deloading operation of DFIG-based wind farm for load sharing," *IEEE Transactions on Sustainable Energy*, to be published, doi: 10.1109/TSTE.2020.3002690.
- [6] T.-H. Chang, A. Nedi, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1524–1538, 2014.
- [7] Z. Li and Z. Ding, "Distributed multiobjective optimization for network resource allocation of multiagent systems," *IEEE Transactions on Cybernetics*, to be published, doi: 10.1109/TCYB.2019.2961475.
- [8] S. Liang, L. Y. Wang, and G. Yin, "Exponential convergence of distributed primaldual convex optimization algorithm without strong convexity," *Automatica*, vol. 105, pp. 298–306, 2019.
- [9] B. Ghareisifard and J. Corts, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, 2014.
- [10] Z. Li, Z. Ding, J. Sun, and Z. Li, "Distributed adaptive convex optimization on directed graphs via continuous-time algorithms," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1434–1441, 2018.
- [11] S. Liang, X. Zeng, and Y. Hong, "Distributed nonsmooth optimization with coupled inequality constraints via modified Lagrangian function," *IEEE Transactions on Automatic Control*, vol. 63, no. 6, pp. 1753–1759, 2018.
- [12] Y. Lu and M. Zhu, "Privacy preserving distributed optimization using homomorphic encryption," *Automatica*, vol. 96, pp. 314–325, 2018.
- [13] B. Ning, Q. Han, and Z. Zuo, "Practical fixed-time consensus for integrator-type multi-agent systems: A time base generator approach," *Automatica*, vol. 105, pp. 406–414, 2019.
- [14] B. Ning, Q. Han, and Z. Zuo, "Distributed optimization of multiagent systems with preserved network connectivity," *IEEE Transactions on Cybernetics*, vol. 49, no. 11, pp. 3980–3990, 2019.
- [15] G. Chen and Z. Li, "A fixed-time convergent algorithm for distributed convex optimization in multi-agent systems," *Automatica*, vol. 95, pp. 539–543, 2018.
- [16] X. Li, L. Xie, and Y. Hong, "Distributed continuous-time nonsmooth convex optimization with coupled inequality constraints," *IEEE Transactions on Control of Network Systems*, to be published, doi: 10.1109/TCNS.2019.2915626.
- [17] A. I. J. Forrester and A. J. Keane, "Recent advances in surrogate-based optimization," *Progress in Aerospace Sciences*, vol. 45, no. 1, pp. 50–79, 2009.
- [18] Z. Liang, J. Chen, and Y. Wang, "Equivalent acceleration imitation for single wheel of manned lu-

- nar rover by varying torque on earth,” *IEEE/ASME Transactions on Mechatronics*, to be published, doi: 10.1109/TMECH.2019.2953330.
- [19] Y. Jin, “Surrogate-assisted evolutionary computation: Recent advances and future challenges,” *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.
 - [20] H.-M. Gutmann, “A radial basis function method for global optimization,” *Journal of Global Optimization*, vol. 19, no. 3, pp. 201–227, 2001.
 - [21] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-free Optimization*. SIAM, 2009.
 - [22] A. R. Conn, K. Scheinberg, and P. L. Toint, “Recent progress in unconstrained nonlinear optimization without derivatives,” *Mathematical Programming*, vol. 79, no. 1-3, p. 397, 1997.
 - [23] Q. Chen and I. Babuka, “Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle,” *Computer Methods in Applied Mechanics and Engineering*, vol. 128, no. 3-4, pp. 405–417, 1995.
 - [24] T. Hengl, G. B. M. Heuvelink, and D. G. Rossiter, “About regression-kriging: From equations to case studies,” *Computers & Geosciences*, vol. 33, no. 10, pp. 1301–1315, 2007.
 - [25] A. Bhosekar and M. Ierapetritou, “Advances in surrogate based modeling, feasibility analysis, and optimization: A review,” *Computers & Chemical Engineering*, vol. 108, pp. 250–267, 2018.
 - [26] J. Tian, Y. Tan, J. Zeng, C. Sun, and Y. Jin, “Multiobjective infill criterion driven gaussian process-assisted particle swarm optimization of high-dimensional expensive problems,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 459–472, 2019.
 - [27] R. G. Regis and C. A. Shoemaker, “Constrained global optimization of expensive black box functions using radial basis functions,” *Journal of Global Optimization*, vol. 31, no. 1, pp. 153–171, 2005.
 - [28] Y. Xu, “Optimal distributed charging rate control of plug-in electric vehicles for demand management,” *IEEE Transactions on Power Systems*, vol. 30, no. 3, pp. 1536–1545, 2014.
 - [29] T. Zhao and Z. Ding, “Distributed initialization-free cost-optimal charging control of plug-in electric vehicles for demand management,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 2791–2801, 2017.
 - [30] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*. New York: Springer-Verlag, 2003.
 - [31] R. T. Rockafellar, *Convex Analysis*. Princeton, New Jersey: Princeton University Press, 1970, vol. 28.
 - [32] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*. Berlin Heidelberg: Springer, 2009, vol. 317.
 - [33] A. B. Taylor, J. M. Hendrickx, and F. Glineur, “Smooth strongly convex interpolation and exact worst-case performance of first-order methods,” *Mathematical Programming*, vol. 161, no. 1-2, pp. 307–345, 2017.
 - [34] W. Yao, X. Chen, Y. Huang, and M. van Tooren, “A surrogate-based optimization method with RBF neural network enhanced by linear interpolation and hybrid infill strategy,” *Optimization Methods and Software*, vol. 29, no. 2, pp. 406–429, 2014.
 - [35] P. Lin, W. Ren, and J. A. Farrell, “Distributed continuous-time optimization: Nonuniform gradient gains, finite-time convergence, and convex constraint set,” *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2239–2253, 2017.
 - [36] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge University Press, 2012.
 - [37] M. Gibbs, “Bayesian gaussian processes for regression and classification,” Ph.D. dissertation, University of Cambridge, 1998.
 - [38] A. Torn and A. Ilinskas, *Global Optimization*. Berlin Heidelberg: Springer, 1989, vol. 350.
 - [39] M. C. Pinar and S. A. Zenios, “On smoothing exact penalty functions for convex constrained optimization,” *SIAM Journal on Optimization*, vol. 4, no. 3, pp. 486–511, 1994.
 - [40] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
 - [41] N. Parikh and S. Boyd, “Block splitting for distributed optimization,” *Mathematical Programming Computation*, vol. 6, no. 1, pp. 77–102, 2014.
 - [42] W. Shi, Q. Ling, G. Wu, and W. Yin, “A proximal gradient algorithm for decentralized composite optimization,” *IEEE Transactions on Signal Processing*, vol. 63, no. 22, pp. 6013–6023, 2015.
 - [43] E. Wei and A. Ozdaglar, “On the $O(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers,” in *2013 IEEE Global Conference on Signal and Information Processing*, 2013, pp. 551–554.
 - [44] D. Jakoveti, J. Xavier, and J. M. Moura, “Fast distributed gradient methods,” *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131–1146, 2014.
 - [45] J. Chen and A. H. Sayed, “Diffusion adaptation strategies for distributed optimization and learning over networks,” *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, 2012.
 - [46] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge University Press, 2004.
 - [47] B. Ning, Q. Han, and Z. Zuo, “Distributed optimization for multiagent systems: An edge-based fixed-time consensus approach,” *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 122–132, 2019.