Noname manuscript No.
(will be inserted by the editor)

Linearized Proximal Algorithms with Adaptive Stepsizes for Convex Composite Optimization with Applications

Yaohua Hu · Chong Li · Jinhua Wang · Xiaoqi Yang · Linglingzhi Zhu

Received: date / Accepted: date

Abstract We propose an inexact linearized proximal algorithm with an adaptive stepsize, together with its globalized version based on the backtracking line-search, to solve a convex composite optimization problem. Under the assumptions of local weak sharp minima of order p ($p \geq 1$) for the outer convex function and a quasi-regularity condition for the inclusion problem associated to the inner function, we establish superlinear/quadratic convergence results for proposed algorithms. Compared to the linearized proximal algorithms with a constant stepsize proposed in [19], our algorithms own broader applications and higher convergence rates, and the idea of analysis used in the present paper deviates significantly from that of [19]. Numerical applications to the nonnegative inverse eigenvalue problem and the wireless sensor network localization problem indicate that the proposed algorithms are more efficient and robust, and outperform the algorithms in [19] and some popular algorithms for relevant problems.

Yaohua Hu
College of Mathematics and Statistics, Shenzhen University, Shenzhen 518060, People's Republic of China
E-mail: mayhhu@szu.edu.cn

Chong Li
School of Mathematical Sciences, Zhejiang University, Hangzhou 310027, People's Republic of China
E-mail: cli@zju.edu.cn

Jinhua Wang
Department of Mathematics, Hangzhou Normal University, Hangzhou 311121, People's Republic of China
E-mail: wangjh@hznu.edu.cn

Xiaoqi Yang
Department of Applied Mathematics, The Hong Kong Polytechnic University, Kowloon, Hong Kong
E-mail: mayangxq@polyu.edu.hk

Linglingzhi Zhu
Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong
E-mail: llzzhu@se.cuhk.edu.hk

Keywords Convex composite optimization · Linearized proximal algorithm · Adaptive stepsize · Quadratic convergence · Convex inclusion problem

Mathematics Subject Classification (2010) Primary, 65K05, 49M37 · Secondary, 90C26

1 Introduction

In the present paper, we consider the following convex composite optimization problem

$$\min_{x\in\mathbb{R}^n} f(x) := (h\circ F)(x), \tag{1.1}$$

where the inner function $F:\mathbb{R}^n\to\mathbb{R}^m$ is continuously differentiable, and the outer function $h:\mathbb{R}^m\to\mathbb{R}$ is convex. The minimal value and the set of minima of h are denoted by

$$h_{\min} := \min_{y\in\mathbb{R}^m} h(y) \quad \text{and} \quad C := \{y\in\mathbb{R}^m : h(y) = h_{\min}\}, \tag{1.2}$$

respectively. Problem (1.1) is a typical class of structured (nonconvex) optimization problems, which provides a unified framework for not only a wide variety of important optimization problems including convex inclusions, penalty methods for nonlinear programming and regularized minimization problems (see, e.g., [7,16,22,32,36]), but also the development and analysis of optimization algorithms (see, e.g., [8,11,16,19,32]).

By virtue of the composite structure, many exclusive and efficient methods have been developed to solve the convex composite optimization problem (1.1). One of the most important methods is the famous Gauss-Newton method (GNM); see [10,23,24,37] and references therein. To establish convergence results of the GNM, the following two assumptions are considered:

(A_0) C is the set of weak sharp minima for h.
(B_0) A regularity condition of the inclusion $F(x)\in C$ holds.

In their seminal paper [10], Burke and Ferris, under assumptions (A_0) and (B_0), established a semi-local quadratic convergence result (by providing a convergence criterion based on the data around the initial point) for the GNM. Furthermore, a globalized version of the GNM based on a backtracking line-search was proposed there and its global quadratic convergence result was also established under the same assumptions. Li and Wang [24] improved the corresponding semi-local convergence result for the GNM and the corresponding global convergence one for the globalized version of the GNM, respectively, by removing assumption (A_0) and by relaxing assumption (A_0) to the assumption that $\lim_{\mathrm{d}(y,C)\to 0}\frac{h(y)-h_{\min}}{\mathrm{d}^2(y,C)} = +\infty$, a weaker one than assumption (A) below with $1\le p<2$. This semi-local convergence result for the GNM was further improved in [23] by virtue of a majorizing function technique, under the following assumption (B); this issue was also re-studied in [15] by a similar majorizing function technique.

(A) C is the set of weak sharp minima for h of order p ($p\ge 1$).

(B) A quasi-regularity condition for the inclusion $F(x) \in C$ holds.

Clearly, (A) and (B) are weaker than (A_0) and (B_0), respectively.

However, as explained in [19], the GNM is inefficient to implement for many practical applications. To tackle this obstacle, inspired by the idea of the search direction in an iterative algorithm named ProxDescent proposed in [22] for solving the composite optimization problem (1.1) (with the outer function being prox-regular), Hu et al. [19] proposed a type of *linearized proximal algorithms with "constant-type" stepsize* (CLP algorithms for short) $\{v_k\}$: $0 < \underline{v} \le v_k \le \bar{v} < +\infty$[1] to solve problem (1.1), which include exact, inexact and globalized versions (i.e., [19, Algorithms 10, 19 and 17], respectively). Moreover, Drusvyatskiy and Lewis [14] employed this idea to propose a prox-linear algorithm to minimize the summation of a convex function and the composed function in (1.1). It was revealed in [19] that their proposed algorithms enjoy an attractive computational advantage as each subproblem involved is a strongly convex optimization problem, which is much easier to solve than that of the GNM. For the CLP-type algorithms, the convergence properties rely not only on assumption (B)/(B_0) but also heavily on the order p in assumption (A). Under assumptions (A) and (B)/(B_0), the following convergence properties were established in [19]:

- The exact/inexact CLP algorithm converges locally to a solution of problem (1.1) at a rate of $\frac{2}{p}$ if $1 \le p < 2$, or $p = 2$ and the stepsize $\{v_k\}$ in the algorithm satisfies $\inf v_k > \frac{2}{\eta(\bar{x})[\beta(\bar{x})]^2}$[2].
- The globalized CLP (GCLP for short) algorithm converges globally to a solution of problem (1.1) at a rate of $\frac{2}{p}$ if $1 \le p < 2$ and the generated sequence has a cluster point.

Thus, in the case when $p = 1$, CLP-type algorithms converge quadratically and maintain the same convergence rate as that of the GNM [10]. However, from the theoretical results mentioned above and also as illustrated by numerical results (see Table 1 of this paper and [19, Figure 4]), CLP-type algorithms suffer from the following drawbacks or limitations:

- The exact/inexact CLP algorithm (resp. the globalized CLP algorithm) does not work very efficiently in the case when $p > 2$ (resp. $p \ge 2$).
- In the case when $p = 2$, the convergence performance of the exact/inexact CLP algorithm is sensitive to the choice of the stepsize (that is, the choice of the stepsize depends on the weak sharp minima modulus and the quasi-regular modulus).

Note that the order p in assumption (A) is an intrinsic constant of the outer function h in the problem. The restriction that $p \le 2$ or $p < 2$ is too stringent as any (nontrivial) twice continuously differentiable function does not have a set of weak sharp minima of order $p < 2$. Thus, CLP-type algorithms cannot be applied to efficiently solve the important class of the convex composite optimization problems with the outer function being a convex polynomial (see [25]). Moreover, it is quite difficult or expensive to provide a clear estimation for the weak sharp minima modulus and the quasi-regular modulus for some practical problems, especially

[1] This was missed in the statements of [19, Algorithms 10, 17 and 19].

[2] $\eta(\bar{x})$ and $\beta(\bar{x})$ are the local weak sharp minima modulus of order 2 and the quasi-regular modulus around the involved point $\bar{x}$, respectively.

in the case when the involved problem is large-scale; this would cause some difficulties in the choice of the stepsize when $p = 2$.

Instead of the "constant-type" stepsizes, we consider in the present paper the following type of adaptive stepsizes:

$$u_k := \min\{\sigma, \theta((h \circ F)(x_k) - h_{\min})^{\alpha}\} \quad \text{for each } k \geq 0,$$

where $0 < \theta < 1$, $\alpha \geq 0$ and $\sigma > 0$ are constants, and then propose a type of *linearized proximal algorithms with the adaptive stepsizes* (ALP algorithms for short), which also include exact, inexact and globalized versions; in particular, the residual controls for the inexact ALP algorithm are different from the ones used in the inexact CLP algorithm [19]. Obviously, in the implementation of ALP-type algorithms for the convex composite optimization problem (1.1) with known data of the minimal value $h_{\min}$, the computational cost of ALP-type algorithms at each iteration is comparable to that of CLP-type algorithms, and thus they inherit the same computational advantage mentioned above owned by CLP-type algorithms. Under the same assumptions (A) and (B)/(B$_0$) used for CLP-type algorithms, we establish in the present paper the following convergence results for ALP-type algorithms (assuming $\alpha > p - 2$):

- The ALP algorithm converges locally to a solution of problem (1.1) at a rate of $\min\{2, \frac{2+\alpha}{p}\}$.
- The globalized ALP (GALP for short) algorithm converges globally to a solution of problem (1.1) at a rate of $\min\{2, \frac{2+\alpha}{p}\}$ if $1 \leq p < 2$ and the generated sequence has a cluster point.

In the case when the outer function h in problem (1.1) satisfies that $(h - h_{\min})^{\frac{1}{s}}$ is locally Lipschitz for some positive constant s satisfying $s \geq 1$ if $1 \leq p < 2$ and $s > \frac{1+\sqrt{1+2p(p-2)}}{2}$ if $p \geq 2$, we further establish the following convergence results (assuming $\frac{p-2}{s} < \alpha \leq \frac{2s-2}{p}$):

- The ALP algorithm converges locally to a solution of problem (1.1) at a rate of $\min\{\frac{2s}{p}, \frac{2+\alpha s}{p}\}$.
- The GALP algorithm converges globally to a solution of problem (1.1) at a rate of $\min\{\frac{2s}{p}, \frac{2+\alpha s}{p}\}$ if the generated sequence has a cluster point.

Therefore, for the case when $1 \leq p < 2$, the GALP algorithm converges at a rate of $\min\{2, \frac{2+\alpha}{p}\}$ or $\min\{\frac{2s}{p}, \frac{2+\alpha s}{p}\}$ (with $\alpha \geq 0$ and $s \geq 1$), while the GCLP algorithm converges at a rate of $\frac{2}{p}$. See Theorems 1, 2 and 3 for more details.

These theoretical results together with numerical results (see Table 1 and Figures 1 and 2 of this paper) show that ALP-type algorithms not only eliminate the drawbacks and limitations mentioned above for CLP-type algorithms, but also outperform CLP-type algorithms on both the convergence rate and CPU time (choosing suitable parameter α can further improve the convergence rate of ALP-type algorithms; see Figure 5). It should be remarked that the idea of analysis for most results in this paper deviates significantly from that of [19], except the approach for Theorem 1.

Two applications are provided as special examples of the convex inclusion problem discussed in section 4. One is the nonnegative inverse eigenvalue problem (NIEP). It is revealed from the numerical results that ALP-type algorithms are more efficient than CLP-type algorithms and the popular Riemannian inexact Newton-CG method (RINC) [38] for the sparse NIEP (see Table 3) and enjoy a faster convergence rate than CLP-type algorithms (also for the dense NIEP); see Table 1 and Figure 1. The other is the wireless sensor network (WSN) localization problem. The numerical results show that ALP-type algorithms enjoy a faster convergence rate, and thus achieve more precise WSN localizations and spend less CPU time than CLP-type algorithms and the popular semidefinite relaxation technique (SDR) [4] (see Figure 2 and Table 6). Moreover, a fast initialization strategy based on the multidimensional scaling (MDS) [21] is proposed, and the resulting ALP-type algorithms are more efficient and robust than the corresponding CLP-type algorithms and SDR; see Table 7 and Figures 3 and 4.

The remainder of this paper is organized as follows. In section 2, Algorithm ALP and its globalized version are proposed, where the main convergence theorems of these algorithms are presented, as well as useful notation and preliminary results. The proofs of these convergence theorems are provided in section 3. Applications to the convex inclusion problem and numerical experiments on the NIEP and the WSN localization problem are demonstrated in sections 4 and 5, respectively. A useful proposition and its technical proof are deferred to the appendix.

2 Algorithms and their convergence results

2.1 Notation

Throughout this paper, we use the following notation. Let $\mathbb{R}^n$ be the n-dimensional Euclidean space with inner product $\langle\cdot,\cdot\rangle$ and its induced norm $\|\cdot\|$. As usual, for $x \in \mathbb{R}^n$ and $r > 0$, let $\mathbf{B}(x,r)$ denote the open ball of radius r centered at x. For $g : \mathbb{R}^n \to \mathbb{R}$ and $\epsilon \geq 0$, the set of all ϵ-optimal solutions of g is defined by

$$\epsilon\text{-}\arg\min g := \{x \in \mathbb{R}^n : g(x) \leq \inf_{y\in\mathbb{R}^n} g(y) + \epsilon\},$$

which is reduced to the optimal solution set $\arg\min g$ when $\epsilon = 0$. If g is convex, the subdifferential of g at x is defined by

$$\partial g(x) := \{w \in \mathbb{R}^n : g(y) \geq g(x) + \langle w, y - x\rangle \text{ for each } y \in \mathbb{R}^n\}.$$

Consider a set $Z \subseteq \mathbb{R}^n$. The dual cone (also called negative polar) of Z, the distance function of Z and the projection onto Z are denoted by $Z^\ominus$, $\mathrm{d}(\cdot, Z)$ and $P_Z(\cdot)$; and are defined by

$$Z^\ominus := \{z \in \mathbb{R}^n : \langle z, y\rangle \leq 0 \text{ for each } y \in Z\},$$

$$\mathrm{d}(x,Z) := \inf_{y\in Z} \|x-y\| \text{ and } P_Z(x) := \{\bar{x} \in Z : \|x-\bar{x}\| = \mathrm{d}(x,Z)\} \quad \text{for each } x \in \mathbb{R}^n,$$

respectively, where we adopt the convention that $\mathrm{d}(x,\emptyset) = +\infty$. Let $T : \mathbb{R}^n \to \mathbb{R}^m$. The image $\operatorname{im} T$ is defined by

$$\operatorname{im} T := \{T(x) \in \mathbb{R}^m : x \in \mathbb{R}^n\},$$

and T is said to be Lipschitz on Z with modulus $L > 0$ if

$$\|T(x) - T(y)\| \leq L\|x - y\| \quad \text{for each } x, y \in Z.$$

Particularly, T is said to be locally Lipschitz around $\bar{x} \in \mathbb{R}^n$ if there exist $r > 0$ and $L_r > 0$ such that T is Lipschitz on $\mathbf{B}(\bar{x}, r)$ with Lipschitz modulus L_r.

2.2 Algorithms

Recall that h and F are the functions involved in the convex composite optimization problem (1.1), and that C and $h_{\min}$ are given by (1.2). Throughout the whole paper, we assume that the minimal value $h_{\min}$ is priorly known (but not the set of minima C).

Remark 1 (a) There are many important applications in which the minimal value $h_{\min}$ is known; e.g., the outer function h is taken to be a norm, a penalty function, or a distance function to a closed and convex set. In particular, in the case when $h(\cdot) = \frac{1}{2}\|\cdot\|^2$, (1.1) is reduced to the nonlinear least squares problem; in the case when $h(\cdot) := \frac{1}{p}\mathrm{d}^p(\cdot, Q)$ (with $p > 1$ and $Q \subseteq \mathbb{R}^m$ being a closed convex set), (1.1) is reduced to the convex inclusion problem (see Remark 5 in section 3). For these two cases, $h_{\min} = 0$.

(b) For the general case, there are many efficient numerical algorithms for convex optimization to approach the minimal values $h_{\min}$ and $\inf\limits_{d \in \mathbb{R}^n} f_{x,u}(d)$ in (2.1); see, e.g., [27].

In the following ALP-type algorithms for solving problem (1.1), we always assume that

$$\sigma > 0, \quad 0 < \theta < 1, \quad \alpha \geq 0 \quad \text{and} \quad \rho \geq \alpha + 2,$$

and define, for fixed $u \geq 0$ and $x \in \mathbb{R}^n$, the linearized proximal function (LP-function for short) $f_{x,u}(\cdot) : \mathbb{R}^n \to \mathbb{R}$ by

$$f_{x,u}(d) := h(F(x) + F'(x)d) + u\|d\|^2 \quad \text{for each } d \in \mathbb{R}^n. \tag{2.1}$$

Algorithm ALP. Choose an initial point $x_0 \in \mathbb{R}^n$ and set $k := 0$.

Step 1. Calculate $w_k := (h \circ F)(x_k) - h_{\min}$, $u_k := \min\{\sigma, \theta w_k^\alpha\}$, and choose $0 \leq \epsilon_k \leq \theta w_k^\rho$.
Step 2. If $(h \circ F)(x_k) = \inf\limits_{d \in \mathbb{R}^n} f_{x_k,u_k}(d)$, then stop.
Step 3. If $(h \circ F)(x_k) \leq \inf\limits_{d \in \mathbb{R}^n} f_{x_k,u_k}(d) + \epsilon_k$, then set $\epsilon_k := \theta\epsilon_k$ and go back to Step 3.
Step 4. Find d_k satisfying

$$f_{x_k,u_k}(d_k) \leq \inf_{d \in \mathbb{R}^n} f_{x_k,u_k}(d) + \epsilon_k. \tag{2.2}$$

Step 5. Set $x_{k+1} := x_k + d_k$ and update $k := k + 1$. Go back to Step 1.

Below, we propose a globalized version of Algorithm ALP, i.e., Algorithm GALP, based on the backtracking line-search.

Algorithm GALP. Choose an initial point $x_0 \in \mathbb{R}^n$, $\gamma \in (0, 1)$, $\lambda \in (0, 1)$ and set $k := 0$.

Step 1. Generate d_k by Steps 1-4 in Algorithm ALP.
Step 2. Find t_k which is the maximum value of γ^i for $i = 0, 1, \cdots$ such that

$$(h \circ F)(x_k + \gamma^i d_k) - (h \circ F)(x_k) \le \lambda \gamma^i \left(f_{x_k, u_k}(d_k) - (h \circ F)(x_k) \right).$$

Step 3. Set $x_{k+1} := x_k + t_k d_k$ and update $k := k + 1$. Go back to Step 1.

Remark 2 (a) In the case when $\epsilon_k = 0$ for each $k \ge 0$, Algorithms ALP and GALP are reduced to the exact ALP algorithm and its globalized version, which are denoted as Algorithm ALP-E and Algorithm GALP-E, respectively.

(b) Although the function $f_{x,u}(\cdot)$ given by (2.1) is a special case of the function $\phi(\cdot, x, \rho)$ given by [7, (3.2)] with $\rho(\cdot) := u\|\cdot\|$, the direction chosen in Algorithm GALP is different from the corresponding ones in [7, choices 4.2-4.4] in the following senses:

- In [7, choices 4.2-4.4], a regularized function $\rho(\cdot)$ is selected for all iterations, while in Algorithm GALP, the regularized functions are adaptively selected as $\rho_k(\cdot) := u_k\|\cdot\|$ with an adaptive stepsize $u_k := \min\{\sigma, \theta w_k^{\alpha}\}$.
- In Algorithm GALP, $\Delta_k := f_{x_k, u_k}(d_k) - (h \circ F)(x_k)$ is not required to be less than $-\epsilon_k$ and so is different from the corresponding one in [7, choice 4.4].
- In Algorithm GALP, (2.2) is equivalent to $d_k \in D_k := \{\bar{d} \in \mathbb{R}^n : \ 0 \in \partial_{\epsilon_k} f_{x_k, u_k}(\bar{d})\}$, in which D_k is different from the corresponding one in [7, choice 4.4]. Here for a convex function $g : \mathbb{R}^n \to \mathbb{R}$ and $\epsilon \ge 0$,

$$\partial_\epsilon g(x) := \{w \in \mathbb{R}^n : g(y) \ge g(x) + \langle w, y - x\rangle - \epsilon \ \text{ for each } y \in \mathbb{R}^n\}.$$

2.3 Convergence results

Associated to problem (1.1), we consider the following inclusion

$$F(x) \in C, \tag{2.3}$$

where C is defined by (1.2). For $x \in \mathbb{R}^n$, let $D(x)$ be defined by

$$D(x) := \{d \in \mathbb{R}^n : F(x) + F'(x)d \in C\}.$$

To establish the convergence results for Algorithms ALP and GALP, we need two important notions: one is the notion of the quasi-regular point for an inclusion problem (2.3), and the other is that of the local weak sharp minimizer of order $p \ge 1$ for a function.

Definition 1 Let $\bar{x} \in \mathbb{R}^n$ and $p \ge 1$. Then $\bar{x}$ is said to be

(a) a quasi-regular point for (2.3) if there exist $r > 0$ and $\beta_r > 0$ such that

$$\beta_r \mathrm{d}(0, D(x)) \le \mathrm{d}(F(x), C) \quad \text{for each } x \in \mathbf{B}(\bar{x}, r). \tag{2.4}$$

(b) a local weak sharp minimizer of order p for g if there exist $r > 0$ and $\eta_r > 0$ such that

$$\eta_r \mathrm{d}^p(x, \arg\min g) \le g(x) - g(\bar{x}) \quad \text{for each } x \in \mathbf{B}(\bar{x}, r).$$

Remark 3 (a) The notion of the quasi-regular point was originally introduced by Li and Ng [23]. Recall from Burke and Ferris [10] that a point $\bar{x}$ is said to be a regular point for (2.3) if

$$\ker(F'(\bar{x})^\top) \cap (C - F(\bar{x}))^\ominus = \{0\}.$$

By [10, Proposition 3.3], a regular point is also a quasi-regular point for (2.3).

(b) The concepts of weak sharp minimizers, introduced by Burke and Ferris [9], have been widely explored and played a key role for convergence analysis of many algorithms, see [10,24,39,40] and references therein. One natural extension of these concepts is that of weak sharp minimizers of order p ($p \geq 1$); see [6,20,28,34] and references therein. Here the definition of a local weak sharp minimizer of order p was introduced by Studniarski and Ward [34].

For the remainder of this section, we shall use the following blanket assumptions:

$$\begin{cases} (\bar{x}, p) \in \mathbb{R}^n \times [1, +\infty); \\ \bar{x} \text{ is a quasi-regular point for (2.3);} \\ F(\bar{x}) \in C \text{ is a local weak sharp minimizer of order } p \text{ for } h; \\ F' \text{ is locally Lipschitz around } \bar{x}. \end{cases} \tag{2.5}$$

Furthermore, we write for any $\alpha \geq 0$ and $s \geq 1$ that

$$q_\alpha := \min\left\{2, \frac{2+\alpha}{p}\right\} \quad \text{and} \quad q_{\alpha,s} := \min\left\{\frac{2s}{p}, \frac{2+\alpha s}{p}\right\}. \tag{2.6}$$

Below, we present convergence theorems for Algorithm ALP.

Theorem 1 *Assume* (2.5)*, and let $\alpha > p - 2$. Then $q_\alpha > 1$, and for any $\delta > 0$, there exists $r_\delta \in (0, \delta)$ such that any sequence $\{x_k\}$, generated by Algorithm ALP with initial point $x_0 \in \mathbf{B}(\bar{x}, r_\delta)$, stays in $\mathbf{B}(\bar{x}, \delta)$ and converges to some point x^* satisfying $F(x^*) \in C$ at a rate of q_α. In particular, the sequence $\{x_k\}$ converges quadratically if $\alpha \geq 2p - 2$.*

In the case when $(h - h_{\min})^{\frac{1}{s}}$ is locally Lipschitz at $F(\bar{x})$, we have the following convergence theorem.

Theorem 2 *Assume* (2.5)*, and that*

$$(h - h_{\min})^{\frac{1}{s}} \textit{ is locally Lipschitz at } F(\bar{x}) \tag{2.7}$$

for some positive constant s satisfying

$$s \geq 1 \textit{ if } 1 \leq p < 2 \quad \textit{and} \quad s > \tfrac{1+\sqrt{1+2p(p-2)}}{2} \textit{ if } p \geq 2. \tag{2.8}$$

Suppose that $\frac{p-2}{s} < \alpha \leq \frac{2s-2}{p}$. Then $q_{\alpha,s} > 1$, and for any $\delta > 0$, there exists $r_\delta \in (0, \delta)$ such that any sequence $\{x_k\}$, generated by Algorithm ALP with initial point $x_0 \in \mathbf{B}(\bar{x}, r_\delta)$, stays in $\mathbf{B}(\bar{x}, \delta)$ and converges to some point x^* satisfying $F(x^*) \in C$ at a rate of $q_{\alpha,s}$. In particular, the sequence $\{x_k\}$ converges quadratically if $s = p$ and $\alpha = 2 - \frac{2}{p}$.

Remark 4 Recall that $f = h \circ F$ is defined in (1.1).

(a) Theorem 2 remains true if the second and third assumptions in (2.5) are replaced by the one that $\bar{x}$ is a local weak sharp minimizer of order p for f and $F(\bar{x}) \in C$; see the proof of Theorem 2.

(b) Assumption (2.5) implies that $\bar{x}$ is a local weak sharp minimizer of order p for f. Indeed, by (2.5), there exist positive constants L, β and r such that, on $\mathbf{B}(\bar{x}, 2r)$, F and F' are Lipschitz with modulus L and (2.4) holds with $\beta_r := \frac{1}{\beta}$. Set $r_0 := \min\{\frac{1}{L^2\beta^2}, \frac{r}{2L\beta}, r\}$ and fix $x_0 \in \mathbf{B}(\bar{x}, r_0)$. Then [23, (4.23)] holds on $\mathbf{B}(x_0, r) \subseteq \mathbf{B}(\bar{x}, 2r)$ with β. Let $\eta := 1$, $\Delta := +\infty$, and let ξ and R^* be defined as in [23, Theorem 5.1]. That is, $\xi = \beta \mathrm{d}(F(x_0), C)$ and $R^* = \frac{1+L\beta\xi-\sqrt{1-(L\beta\xi)^2}}{L\beta}$. Then [23, (5.7)] holds and F' is Lipschitz on $\mathbf{B}(x_0, R^*)$ with Lipschitz modulus L because

$$\xi = \beta \mathrm{d}(F(x_0), C) \leq \beta \|F(x_0) - F(\bar{x})\| \leq L\beta \|x_0 - \bar{x}\| \leq \min \left\{ \frac{1}{L\beta}, \frac{r}{2} \right\}$$

and $R^* \leq 2\xi \leq r$. Thus, all assumptions in [23, Theorem 5.1] are seen to hold, and so [23, Theorem 5.1] is applicable. Clearly, Algorithm A (η, Δ, x_0) with initial point x_0 in [23] is well-defined, and then by [23, Theorem 5.1] the generated sequence $\{x_n\}$ converges to some $x^* \in \mathbb{R}^n$ with $F(x^*) \in C$ and satisfies [23, (5.8)]; in particular, one has that $\|x_0 - x^*\| \leq R^* \leq 2\xi = 2\beta \mathrm{d}(F(x_0), C)$, and so $\mathrm{d}(x, \arg\min f) \leq 2\beta \mathrm{d}(F(x), C)$ for all $x \in \mathbf{B}(\bar{x}, r_0)$. This, together with the weak sharp minimizer assumption for h in (2.5), implies that $\bar{x}$ is a local weak sharp minimizer of order p for f (using a smaller r_0 if necessary), as desired to show.

The following corollary follows directly from Theorems 1 and 2.

Corollary 1 *Assume* (2.5)*, and that assumptions* (2.7) *and* (2.8) *in Theorem 2 hold. Then, Algorithm ALP converges locally to some point x^* satisfying $F(x^*) \in C$ at a rate of q_α if $\alpha > p - 2$ and at a rate of $q_{\alpha,s}$ if $\frac{p-2}{s} < \alpha \leq \frac{2s-2}{p}$.*

Below, we provide the global convergence results for Algorithm GALP.

Proposition 1 *Let $\{x_k\}$ be a sequence generated by Algorithm GALP-E, and suppose that $\{x_k\}$ has an accumulation point $\bar{x}$ such that F' is Lipschitz continuous around $\bar{x}$. Then $\bar{x}$ is a stationary point: $0 \in F'(\bar{x})^\top \partial h(F(\bar{x}))$, and $F(\bar{x}) \in C$ if $\bar{x}$ is further a regular point for* (2.3)*.*

Theorem 3 *Let $\{x_k\}$ be a sequence generated by Algorithm GALP, and suppose that $\{x_k\}$ has an accumulation point $\bar{x}$ satisfying* (2.5)*. Then $\{x_k\}$ converges to $\bar{x}$ at a rate of q_α if $1 \leq p < 2$, and at a rate of $q_{\alpha,s}$ if assumptions* (2.7) *and* (2.8) *in Theorem 2 hold and $\frac{p-2}{s} < \alpha \leq \frac{2s-2}{p}$.*

3 Proofs of convergence theorems

3.1 Basic facts

Fact 4 below is known (see [3, Proposition A.24] for assertion (i) and [19, Lemma 6] for assertion (ii)); Fact 5 is taken from [19, Lemma 9]. Recall that h and F are the functions involved in the convex composite optimization problem (1.1), and that C and $h_{\min}$ are given by (1.2).

Fact 4. *Let $L > 0$ and $W \subseteq \mathbb{R}^n$ be such that F' is Lipschitz on W with modulus L. Then, the following assertions hold for any $x, x + d \in W$:*

(i) $\|F(x + d) - F(x) - F'(x)d\| \le \frac{L}{2}\|d\|^2$;
(ii) *if there are $\eta > 0$ and $p \ge 1$ such that*

$$\eta \mathrm{d}^p(F(x) + F'(x)d, C) \le h(F(x) + F'(x)d) - h_{\min}, \tag{3.1}$$

then

$$\mathrm{d}(F(x + d), C) \le \frac{1}{2}L\|d\|^2 + \eta^{-\frac{1}{p}}(h(F(x) + F'(x)d) - h_{\min})^{\frac{1}{p}}. \tag{3.2}$$

Fact 5. *Let $u > 0$, $x \in \mathbb{R}^n$, and let $f_{x,u}(\cdot) : \mathbb{R}^n \to \mathbb{R}$ be the LP-function defined by (2.1). Then the following assertion holds for any $\epsilon \ge 0$ and $d \in \epsilon\text{-}\arg\min f_{x,u}$:*

$$u\|d\|^2 \le u\mathrm{d}^2(0, D(x)) + \epsilon \quad \textit{and} \quad h(F(x) + F'(x)d) \le h_{\min} + u\mathrm{d}^2(0, D(x)) + \epsilon.$$

3.2 Proofs for main theorems

Proof of Theorem 1. The assertion that $q_\alpha = \min\{2, \frac{2+\alpha}{p}\} > 1$ is clear because $\alpha > p - 2$. To proceed, note by assumption (2.5) that there exist positive constants η, $\bar{\delta}$ and $L \ge 1$ such that F' is Lipschitz on $\mathbf{B}(\bar{x}, \bar{\delta})$ with modulus L (so the conclusions in Fact 4 hold with $\mathbf{B}(\bar{x}, \bar{\delta})$ in place of W),

$$\|F'(x)\| \le L,\ \eta\mathrm{d}(0, D(x)) \le \mathrm{d}(F(x), C),\ ((h\circ F)(x) - h_{\min})^\alpha < \min\{1, \sigma\} \quad \text{for each } x \in \mathbf{B}(\bar{x}, \bar{\delta}), \tag{3.3}$$

and

$$\eta\mathrm{d}^p(y, C) \le h(y) - h_{\min} \le L\|y - \bar{y}\| \quad \text{for each } (y, \bar{y}) \in \mathbf{B}(F(\bar{x}), \bar{\delta}) \times (C \cap \mathbf{B}(F(\bar{x}), \bar{\delta})). \tag{3.4}$$

Let $\delta > 0$ be arbitrary. Without loss of generality, one may assume that

$$\delta \le \min\left\{\frac{2\bar{\delta}}{(5c + 2)L}, \frac{\eta}{2c^2L}, \left(\frac{\eta^{1+p}}{4^p\theta L^{2\alpha}(1 + L^4)}\right)^{\frac{1}{2+\alpha-p}}\right\}, \tag{3.5}$$

where $c := 1 + L^2$. Set $r_\delta := \delta \min\left\{1, \frac{\eta}{2L}\right\}$. Below, we show that r_δ is as desired.

To do this, let $x_0 \in \mathbf{B}(\bar{x}, r_\delta)$, and let $\{x_k\}$, together with $\{d_k\}$, be generated by Algorithm ALP with initial point x_0. Now fix $k \in \mathbb{N}$ and assume $x_k \in \mathbf{B}(\bar{x}, (2c+1)\delta)$; then $x_k \in \mathbf{B}(\bar{x}, \bar{\delta})$

(as $2c+1 < \frac{(5c+2)L}{2}$). One checks by Step 1 of Algorithm ALP that $w_k^\alpha = ((h\circ F)(x_k) - h_{\min})^\alpha \le \min\{1,\sigma\}$ (by (3.3)); hence, $w_k \le 1$ and $u_k = \theta w_k^\alpha$. Furthermore, by Fact 5 (applied to x_k, d_k in place of x, d) and Step 1 of Algorithm ALP,

$$\|d_k\|^2 \le \mathrm{d}^2(0, D(x_k)) + \frac{\epsilon_k}{u_k} \le \mathrm{d}^2(0, D(x_k)) + w_k^2, \tag{3.6}$$

(noting $w_k^{\rho-\alpha} \le w_k^2$ as $\rho \ge \alpha+2$ and $w_k \le 1$). To proceed, we first verify the following implications:

$$\|d\| \le \frac{c\delta}{2} \quad \Longrightarrow \quad F(x_k) + F'(x_k)d \in \mathbf{B}(F(\bar{x}), \bar{\delta}), \tag{3.7}$$

(so $F(x_k) \in \mathbf{B}(F(\bar{x}), \bar{\delta})$ as $d = 0$) and

$$\mathrm{d}(0, D(x_k)) \le \frac{\delta}{2} \quad \Longrightarrow \quad [w_k \le L^2 \mathrm{d}(0, D(x_k)) \text{ and } \|d_k\| \le c\mathrm{d}(0, D(x_k))]. \tag{3.8}$$

In fact, note by (3.3) that $\|F'(x_k)\| \le L$ and so $\|F(x_k) - F(\bar{x})\| \le L\|x_k - \bar{x}\|$. Thus, if $\|d\| \le \frac{c\delta}{2}$, then

$$\|F(x_k) + F'(x_k)d - F(\bar{x})\| \le L\|x_k - \bar{x}\| + L\|d\| < L(2c+1)\delta + L\frac{c}{2}\delta \le \bar{\delta}$$

(by (3.5)); hence (3.7) is checked. To show (3.8), we assume $\mathrm{d}(0, D(x_k)) \le \frac{\delta}{2}$. Then, letting $\tilde{d}_k \in P_{D(x_k)}(0)$, one has that $F(x_k) + F'(x_k)\tilde{d}_k \in C$, and that $\|\tilde{d}_k\| \le \frac{\delta}{2} \le \frac{c\delta}{2}$ (noting $c \ge 1$). Therefore, $F(x_k) + F'(x_k)\tilde{d}_k \in C \cap \mathbf{B}(F(\bar{x}), \bar{\delta})$ by (3.7). Thus it follows from (3.4) (with $F(x_k)$, $F(x_k) + F'(x_k)\tilde{d}_k$ in place of y, $\bar{y}$) and (3.3) that

$$w_k = (h \circ F)(x_k) - h_{\min} \le L^2 \|\tilde{d}_k\| = L^2 \mathrm{d}(0, D(x_k)).$$

Moreover, one sees by (3.6) that

$$\|d_k\|^2 \le \mathrm{d}^2(0, D(x_k)) + w_k^2 \le \mathrm{d}^2(0, D(x_k)) + L^4\mathrm{d}^2(0, D(x_k)) \le c^2\mathrm{d}^2(0, D(x_k)).$$

Thus implication (3.8) is shown.

Below, we shall use mathematical induction to show that for each $k \ge 0$

$$x_k \in \mathbf{B}(\bar{x}, (2c+1)\delta), \quad \mathrm{d}(F(x_k), C) \le \eta\delta \left(\frac{1}{2}\right)^{q_\alpha^k + k} \quad \text{and} \quad \|d_k\| \le c\delta\left(\frac{1}{2}\right)^{q_\alpha^k + k}. \tag{3.9}$$

Granting this, one has that $\{x_k\}$ is a Cauchy sequence and converges to a point x^* satisfying $F(x^*) \in C$ (as F is continuous and C is closed); moreover

$$\|x_k - x^*\| \le \sum_{i=k}^{+\infty} \|d_i\| \le 2c\delta\left(\frac{1}{2}\right)^{q_\alpha^k + k},$$

and so $\{x_k\}$ converges to x^* at a rate of q_α.

Note first that (3.9) hold for $k = 0$ because, $\|x_0 - \bar{x}\| < \delta \min\left\{1, \frac{\eta}{2L}\right\} \le \delta$ (and so $x_0 \in \mathbf{B}(\bar{x}, (2c+1)\delta)$),

$$\mathrm{d}(F(x_0), C) \le \|F(x_0) - F(\bar{x})\| \le L\|x_0 - \bar{x}\| \le \frac{1}{2}\eta\delta \tag{3.10}$$

(noting $\|F'(\cdot)\| \leq L$ by (3.3)), and $\|d_0\| \leq c\mathrm{d}(0, D(x_0)) \leq \frac{1}{2}c\delta$ by (3.8) as $\mathrm{d}(0, D(x_0)) \leq \frac{1}{\eta}\mathrm{d}(F(x_0), C) \leq \frac{\delta}{2}$, thanks to (3.3) and (3.10). Now assume that (3.9) holds for each $k = 0, 1, \cdots, i$. Then, $x_{i+1} \in \mathbf{B}(\bar{x}, (2c+1)\delta)$ (i.e., the first assertion in (3.9) is checked for $k = i+1$) as

$$\|x_{i+1} - \bar{x}\| \leq \sum_{j=0}^{i} \|d_j\| + \|x_0 - \bar{x}\| \leq c\delta \sum_{j=0}^{i} \left(\frac{1}{2}\right)^j + \delta < (2c+1)\delta.$$

Since by (3.9) (applied to i in place of k) and (3.3) that

$$x_i \in \mathbf{B}(\bar{x}, (2c+1)\delta) \subseteq \mathbf{B}(\bar{x}, \bar{\delta}) \quad \text{and} \quad \mathrm{d}(0, D(x_i)) \leq \frac{1}{\eta}\mathrm{d}(F(x_i), C) \leq \delta \left(\frac{1}{2}\right)^{q_\alpha^i + i} < \frac{\delta}{2}, \tag{3.11}$$

it follows from (3.3) and implication (3.8) that

$$w_i^\alpha \leq \min\{1, \sigma\}, \quad w_i \leq L^2\mathrm{d}(0, D(x_i)) \text{ and } \|d_i\| \leq c\mathrm{d}(0, D(x_i)) \leq \frac{1}{2}c\delta. \tag{3.12}$$

Thus, $F(x_i) + F'(x_i)d_i \in \mathbf{B}(F(\bar{x}), \bar{\delta})$ by implication (3.7) (applied to x_i, d_i in place of x_k, d). Therefore, with x_i and d_i in place of x and d, (3.1) is satisfied by (3.4) (applied to $F(x_i) + F'(x_i)d_i$ in place of y), and then (3.2) is true by Fact 4(ii). That is,

$$\mathrm{d}(F(x_{i+1}), C) \leq \frac{L}{2}\|d_i\|^2 + \eta^{-\frac{1}{p}}(h(F(x_i) + F'(x_i)d_i) - h_{\min})^{\frac{1}{p}}. \tag{3.13}$$

Note further that $d_i \in \epsilon_i$- $\arg\min f_{x_i, u_i}$. Fact 5 is applicable to $x_i, d_i, u_i, \epsilon_i$ in place of x, d, u, ϵ, and we have that

$$h(F(x_i) + F'(x_i)d_i) - h_{\min} \leq u_i\mathrm{d}^2(0, D(x_i)) + \epsilon_i. \tag{3.14}$$

By (3.12), one sees by definition that

$$u_i = \theta w_i^\alpha \leq \theta L^{2\alpha}\mathrm{d}^\alpha(0, D(x_i)) \quad \text{and} \quad \epsilon_i \leq \theta w_i^\rho \leq \theta w_i^{\alpha+2} \leq \theta L^{2\alpha+4}\mathrm{d}^{\alpha+2}(0, D(x_i))$$

(as $w_i \leq 1$ and $\rho \geq \alpha + 2$); hence $u_i\mathrm{d}^2(0, D(x_i)) + \epsilon_i \leq \theta L^{2\alpha}(1 + L^4)\mathrm{d}^{\alpha+2}(0, D(x_i))$. This, together with (3.14) and the estimate $\mathrm{d}(0, D(x_i)) \leq \delta\left(\frac{1}{2}\right)^{q_\alpha^i + i}$ in (3.11), implies that

$$\eta^{-\frac{1}{p}}(h(F(x_i) + F'(x_i)d_i) - h_{\min})^{\frac{1}{p}} \leq \left(\eta^{-1}\theta L^{2\alpha}(1 + L^4)\right)^{\frac{1}{p}} \delta^{\frac{2+\alpha}{p}} \left(\frac{1}{2}\right)^{\frac{2+\alpha}{p}(q_\alpha^i + i)}.$$

Since $\|d_i\| \leq c\delta\left(\frac{1}{2}\right)^{q_\alpha^i + i}$ by (3.9), it follows from (3.13) that

$$\begin{aligned} \mathrm{d}(F(x_{i+1}), C) &\leq Lc^2\delta \cdot \delta\left(\tfrac{1}{2}\right)^{2(q_\alpha^i + i)+1} + 2\left(\eta^{-1}\theta L^{2\alpha}(1 + L^4)\right)^{\frac{1}{p}} \delta^{\frac{2+\alpha-p}{p}} \cdot \delta\left(\tfrac{1}{2}\right)^{\frac{2+\alpha}{p}(q_\alpha^i + i)+1} \\ &\leq \eta\delta\left(\tfrac{1}{2}\right)^{q_\alpha(q_\alpha^i + i)+1} \end{aligned} \tag{3.15}$$

because $Lc^2\delta + 2\left(\eta^{-1}\theta L^{2\alpha}(1+L^4)\right)^{\frac{1}{p}}\delta^{\frac{2+\alpha-p}{p}} \le \eta$ by (3.5) and $q_\alpha = \min\{2, \frac{2+\alpha}{p}\}$. Since $q_\alpha(q_\alpha^i + i) + 1 \ge q_\alpha^{i+1} + i + 1$ (as $q_\alpha > 1$), it follows from (3.15) that the second assertion in (3.9) holds for $k = i+1$. Consequently, we further have by (3.3) that

$$\mathrm{d}(0, D(x_{i+1})) \le \frac{1}{\eta}\mathrm{d}(F(x_{i+1}), C) \le \delta\left(\frac{1}{2}\right)^{q_\alpha^{i+1}+i+1} \le \frac{\delta}{2},$$

and implication (3.8) is applicable to concluding that $\|d_{i+1}\| \le c\mathrm{d}(0, D(x_{i+1})) \le c\delta\left(\frac{1}{2}\right)^{q_\alpha^{i+1}+i+1}$. Thus the last assertion in (3.9) is also checked for $k = i+1$, completing the proof. □

To prove Theorem 2, we write $X^* := \arg\min f$ and assume, without loss of generality, that $h_{\min} = 0$ for simplicity. We proceed with the following key lemma.

Lemma 1 *Suppose that all assumptions of Theorem 2 hold. Then there exist $\hat{\delta} > 0$ and $c > 0$ such that the following two implications hold for any sequence $\{x_k\}$, together with the associated sequence $\{d_k\}$, generated by Algorithm ALP:*

$$x_k \in \mathbf{B}(\bar{x}, \hat{\delta}) \quad \Longrightarrow \quad \|d_k\| \le c\mathrm{d}(x_k, X^*) \tag{3.16}$$

and

$$x_k, x_{k+1} \in \mathbf{B}(\bar{x}, \hat{\delta}) \quad \Longrightarrow \quad \mathrm{d}(x_{k+1}, X^*) \le c\mathrm{d}^{q_{\alpha,s}}(x_k, X^*). \tag{3.17}$$

Proof. As in the beginning of the proof for Theorem 1, there exist positive constants $\eta, L \ge 1$ and $0 < \bar{\delta} \le 1$ such that F' is Lipschitz on $\mathbf{B}(\bar{x}, \bar{\delta})$ with modulus L (so the conclusions in Fact 4 hold with $\mathbf{B}(\bar{x}, \bar{\delta})$ in place of W) and that (3.3), together with the following inequalities, holds:

$$h^{\frac{1}{s}}(y) - h^{\frac{1}{s}}(y') \le L\|y - y'\| \quad \text{for any } y, y' \in \mathbf{B}(F(\bar{x}), \bar{\delta}); \tag{3.18}$$

$$\eta\mathrm{d}^p(x, X^*) \le (h \circ F)(x) \quad \text{for each } x \in \mathbf{B}(\bar{x}, \bar{\delta}) \tag{3.19}$$

(see Remark 4(b) for (3.19)). Define

$$c := \max\left\{c_1, \left(c_2^{\frac{1}{s}} + \frac{L^2}{2}c_1^2\right)^s, \left(\frac{1}{\eta}\right)^{\frac{1}{p}}\left(c_2^{\frac{1}{s}} + \frac{L^2}{2}c_1^2\right)^{\frac{s}{p}}\right\} \quad \text{and} \quad \hat{\delta} := \frac{\bar{\delta}}{2}\min\left\{1, \frac{1}{2L}\right\}, \tag{3.20}$$

where $c_1 := \sqrt{\frac{L^{2s}}{2^s\theta\eta^\alpha} + L^{4s} + 1}$ and $c_2 := \frac{L^{2s}}{2^s} + \theta L^{2\alpha s} + \theta L^{4s+2\alpha s}$. Below, we show that c and $\hat{\delta}$ are as desired.

To do this, we have by the similar argument as we did for proving (3.7) the following implication:

$$\left[x \in \mathbf{B}(\bar{x}, 2\hat{\delta}) \text{ and } \|d\| \le 2\hat{\delta}\right] \quad \Longrightarrow \quad F(x) + F'(x)d \in \mathbf{B}(F(\bar{x}), \bar{\delta}). \tag{3.21}$$

Now suppose that $x_k \in \mathbf{B}(\bar{x}, \hat{\delta})$ (and so $x_k \in \mathbf{B}(\bar{x}, \bar{\delta})$). Let $\bar{x}_k \in P_{X^*}(x_k)$, and set $\bar{d}_k := \bar{x}_k - x_k$. Then, $\|\bar{d}_k\| < \hat{\delta}$ and $\bar{x}_k \in \mathbf{B}(\bar{x}, 2\hat{\delta}) \subseteq \mathbf{B}(\bar{x}, \bar{\delta})$ as $\|\bar{d}_k\| \le \|x_k - \bar{x}\| < \hat{\delta}$ and $\|\bar{x}_k - \bar{x}\| \le \|\bar{d}_k\| + \|x_k - \bar{x}\| < 2\hat{\delta}$. Thus, using (3.3), one sees that

$$w_k \le 1 \quad \text{and} \quad \|F(x_k) - F(\bar{x}_k)\| \le L\|x_k - \bar{x}_k\|. \tag{3.22}$$

Furthermore, from implication (3.21), it follows that each of $F(x_k)$, $F(\bar{x}_k)$ and $F(x_k) + F'(x_k)\bar{d}_k$ belongs to $\mathbf{B}(F(\bar{x}), \bar{\delta})$ (by applying (3.21) to 0 and $\bar{d}_k$ in place of d, respectively). Thus one applies (3.18) and (3.19), together with (3.22), to check that

$$w_k = (h \circ F)(x_k) \geq \eta \mathrm{d}^p(x_k, X^*) \quad \text{and} \quad w_k = (h \circ F)(x_k) \leq L^s \|F(x_k) - F(\bar{x}_k)\|^s \leq L^{2s}\|\bar{d}_k\|^s \tag{3.23}$$

(noting that $(h \circ F)(\bar{x}_k) = h_{\min} = 0$). By the choice of u_k and w_k in Step 1 of Algorithm ALP, we have that

$$\epsilon_k \leq \theta w_k^{\rho} = u_k w_k^{\rho - \alpha} \leq u_k w_k^2 \leq u_k L^{4s} \|\bar{d}_k\|^{2s} \tag{3.24}$$

as $\rho - \alpha \geq 2$ and $w_k \leq 1$. Recalling the LP-function f_{x_k,u_k} given in (2.1) and using (3.18) (with $F(x_k) + F'(x_k)\bar{d}_k$, $F(x_k + \bar{d}_k)$ in place of y, y') and Fact 4(i), one concludes that

$$f_{x_k,u_k}(\bar{d}_k) = h(F(x_k) + F'(x_k)\bar{d}_k) + u_k\|\bar{d}_k\|^2 \leq \frac{L^{2s}}{2^s}\|\bar{d}_k\|^{2s} + u_k\|\bar{d}_k\|^2$$

(noting that $(h \circ F)(x_k + \bar{d}_k) = 0$). Since by (2.2) $f_{x_k,u_k}(d_k) \leq f_{x_k,u_k}(\bar{d}_k) + \epsilon_k$, it follows from (3.24) that

$$f_{x_k,u_k}(d_k) \leq \frac{L^{2s}}{2^s}\|\bar{d}_k\|^{2s} + u_k\left(\|\bar{d}_k\|^2 + L^{4s}\|\bar{d}_k\|^{2s}\right). \tag{3.25}$$

Noting by (3.23) that $u_k = \theta w_k^{\alpha} \geq \theta\eta^{\alpha}\mathrm{d}^{\alpha p}(x_k, X^*) = \theta\eta^{\alpha}\|\bar{d}_k\|^{\alpha p}$, we have from (2.1) and (3.25) that

$$\|d_k\|^2 \leq \frac{f_{x_k,u_k}(d_k)}{u_k} \leq \frac{L^{2s}}{2^s\theta\eta^{\alpha}}\|\bar{d}_k\|^{2s-\alpha p} + \|\bar{d}_k\|^2 + L^{4s}\|\bar{d}_k\|^{2s} \leq c_1^2\|\bar{d}_k\|^{\min\{2s-\alpha p,2,2s\}} \tag{3.26}$$

(noting that $\|\bar{d}_k\| < \bar{\delta} \leq 1$). This implies that $\|d_k\| \leq c_1\|\bar{d}_k\| = c_1\mathrm{d}(x_k, X^*)$ because $\min\{2s - \alpha p, 2, 2s\} = 2$ by assumptions $\alpha \leq \frac{2s-2}{p}$ and $s \geq 1$. Hence, implication (3.16) is shown.

To verify implication (3.17), suppose that $x_k, x_{k+1} \in \mathbf{B}(\bar{x}, \hat{\delta})$. Then $d_k \in \mathbf{B}(\bar{x}, 2\hat{\delta})$ (noting that $\|d_k\| \leq \|x_{k+1} - \bar{x}\| + \|x_k - \bar{x}\|$), and so implication (3.21) is applicable to getting that both $F(x_k) + F'(x_k)d_k$ and $F(x_{k+1})$ belong to $\mathbf{B}(F(\bar{x}), \bar{\delta})$. Note by $h_{\min} = 0$ and (2.1) that $h(F(x_k) + F'(x_k)d_k) \leq f_{x_k,u_k}(d_k)$. We check by (3.18) and Fact 4(i) that

$$(h \circ F)^{\frac{1}{s}}(x_{k+1}) \leq f_{x_k,u_k}^{\frac{1}{s}}(d_k) + L\|F(x_{k+1}) - F(x_k) - F'(x_k)d_k\| \leq f_{x_k,u_k}^{\frac{1}{s}}(d_k) + \frac{L^2}{2}\|d_k\|^2. \tag{3.27}$$

By (3.23), one has that $u_k = \theta w_k^{\alpha} \leq \theta L^{2\alpha s}\|\bar{d}_k\|^{\alpha s}$. This and (3.25) imply that

$$f_{x_k,u_k}(d_k) \leq \frac{L^{2s}}{2^s}\|\bar{d}_k\|^{2s} + \theta L^{2\alpha s}\left(\|\bar{d}_k\|^{2+\alpha s} + L^{4s}\|\bar{d}_k\|^{2s+\alpha s}\right) \leq c_2\|\bar{d}_k\|^{\min\{2s,2+\alpha s,2s+\alpha s\}}$$

(recalling that $\|\bar{d}_k\| < \bar{\delta} \leq 1$). Combining this, (3.26) and (3.27), we conclude that

$$(h \circ F)^{\frac{1}{p}}(x_{k+1}) \leq \left(f_{x_k,u_k}^{\frac{1}{s}}(d_k) + \frac{L^2}{2}\|d_k\|^2\right)^{\frac{s}{p}} \leq \left(c_2^{\frac{1}{s}} + \frac{L^2}{2}c_1^2\right)^{\frac{s}{p}}\|\bar{d}_k\|^{\frac{s}{p}\min\{2s-\alpha p,2,\frac{2+\alpha s}{s},2+\alpha,2s\}}.$$

Since $\alpha \geq 0$ and $\min\{2s - \alpha p, 2, 2s\} = 2$ as noted earlier, it follows that

$$\min\left\{2s - \alpha p, 2, \frac{2+\alpha s}{s}, 2+\alpha, 2s\right\} = \min\left\{2, \frac{2+\alpha s}{s}\right\}.$$

Hence we have by the definition of $q_{\alpha,s}$ in (2.6) that

$$(h \circ F)(x_{k+1}) \leq \left(c_2^{\frac{1}{s}} + \frac{L^2}{2}c_1^2\right)^s \mathrm{d}^{pq_{\alpha,s}}(x_k, X^*) \tag{3.28}$$

(noting that $\|\bar{d}_k\| = \mathrm{d}(x_k, X^*)$). This, together with (3.19), implies that

$$\mathrm{d}(x_{k+1}, X^*) \leq \left(\frac{1}{\eta}\right)^{\frac{1}{p}} (h \circ F)^{\frac{1}{p}}(x_{k+1}) \leq \left(\frac{1}{\eta}\right)^{\frac{1}{p}} \left(c_2^{\frac{1}{s}} + \frac{L^2}{2}c_1^2\right)^{\frac{s}{p}} \mathrm{d}^{q_{\alpha,s}}(x_k, X^*),$$

and thus, implication (3.17) is checked. The proof is complete. □

Proof of Theorem 2. One can verify the assertion that $q_{\alpha,s} = \min\{\frac{2s}{p}, \frac{2+\alpha s}{p}\} > 1$ by the assumption that $\frac{p-2}{s} < \alpha$ and the assumption of s in (2.8). Let $\hat{\delta}$ and c be positive constants given by Lemma 1 such that implications (3.16) and (3.17) hold. Let $\delta > 0$ be arbitrary, and without loss of generality, we assume that $\delta \in (0, \hat{\delta})$. Let $r_\delta := \frac{1}{1+2c} \min\left\{\delta, \left(\frac{1}{2c}\right)^{\frac{1}{q_{\alpha,s}-1}}\right\}$, where c is given by (3.20). Below, we shall show that r_δ is as desired.

To do this, let $x_0 \in \mathbf{B}(\bar{x}, r_\delta) \subseteq \mathbf{B}(\bar{x}, \hat{\delta})$, and let $\{x_k\}$ and $\{d_k\}$ be sequences generated by Algorithm ALP with initial point x_0. Then $\mathrm{d}(x_0, X^*) < r_\delta \leq \frac{1}{1+2c}\left(\frac{1}{2c}\right)^{\frac{1}{q_{\alpha,s}-1}} < \left(\frac{1}{2c}\right)^{\frac{1}{q_{\alpha,s}-1}}$. Below we will verify that $\{x_k\} \subseteq \mathbf{B}(\bar{x}, \delta)$. Granting this, one has by (3.16) and (3.17) that (4.4) holds with $q_{\alpha,s}$ in place of q (recalling that $d_k = x_{k+1} - x_k$), and then Proposition A in Appendix is applicable to completing the proof.

Since $x_0 \in \mathbf{B}(\bar{x}, r_\delta)$, it follows that $x_0 \in \mathbf{B}(\bar{x}, \delta)$ and that $\|d_0\| \leq c\mathrm{d}(x_0, X^*) \leq cr_\delta$ by (3.16). Hence $\|x_1 - \bar{x}\| \leq \|x_0 - \bar{x}\| + \|d_0\| < (1+c)r_\delta < \delta$, and so $x_1 \in \mathbf{B}(\bar{x}, \delta)$. Thus, we check by (3.17) that $\mathrm{d}(x_1, X^*) \leq \frac{1}{2}\mathrm{d}(x_0, X^*)$ as $c\mathrm{d}^{q_{\alpha,s}-1}(x_0, X^*) \leq \frac{1}{2}$ as noted earlier. Now fix $i \in \mathbb{N}$ and assume that

$$x_k \in \mathbf{B}(\bar{x}, \delta) \quad \text{and} \quad \mathrm{d}(x_k, X^*) \leq \frac{1}{2}\mathrm{d}(x_{k-1}, X^*) \tag{3.29}$$

hold for all $k \leq i$. Then, $\mathrm{d}(x_k, X^*) \leq (\frac{1}{2})^k \mathrm{d}(x_0, X^*)$ for each $k \leq i$; hence one has by (3.16) that

$$\sum_{k=0}^{i} \|d_k\| \leq c\sum_{k=0}^{i} \mathrm{d}(x_k, X^*) < cr_\delta \sum_{k=0}^{i} \left(\frac{1}{2}\right)^k < 2cr_\delta.$$

Recalling that $x_0 \in \mathbf{B}(\bar{x}, r_\delta)$, it follows that

$$\|x_{i+1} - \bar{x}\| \leq \|x_0 - \bar{x}\| + \sum_{k=0}^{i} \|d_k\| < (1+2c)r_\delta \leq \delta,$$

and so $x_{i+1} \in \mathbf{B}(\bar{x}, \delta)$. This, together with (3.17), implies that (3.29) is satisfied for $k = i+1$ because $\mathrm{d}(x_{i+1}, X^*) \leq c\mathrm{d}^{q_{\alpha,s}-1}(x_i, X^*)\mathrm{d}(x_i, X^*)$ by (3.17) and $c\mathrm{d}^{q_{\alpha,s}-1}(x_i, X^*) \leq c\mathrm{d}^{q_{\alpha,s}-1}(x_0, X^*) \leq \frac{1}{2}$. Therefore, (3.29) holds for all $k \geq 1$ by mathematical induction, and the proof is complete. □

Proof of Proposition 1. By [19, Remark 16], it suffices to verify that $0 \in F'(\bar{x})^\top \partial h(F(\bar{x}))$. To do this, let $\{d_k\}$ be the associated sequence generated by Algorithm GALP-E, and note that each LP-function f_{x_k,u_k} given in (2.1) is strongly convex. Then, for each $k \geq 0$, one has that

$$\Delta_k := f_{x_k,u_k}(d_k) - (h \circ F)(x_k) \leq 0 \quad \text{and} \quad D_k := \arg\min f_{x_k,u_k} = \{d_k\},$$

and that

$$0 \in D_k \quad \Longleftrightarrow \quad \Delta_k = 0 \quad \Longleftrightarrow \quad 0 \in F'(x_k)^\top \partial h(F(x_k))$$

(noting the chain rule for Clarke subgradient: $\partial f(\cdot) = F'(\cdot)^\top \partial h(F(\cdot))$; see [7, Lemma 2.3(a)]). Thus, conditions (a)-(c) in [7, (2.2)] are satisfied and then $\{x_k\}$ can be regarded as a sequence generated by the algorithm [7, (2.1)]. Therefore, [7, Theorem 2.4] is applicable (following the line of analysis in the proof of [7, Theorem 2.4], one can check that the conclusion in [7, Theorem 2.4] remains true under our assumptions).

Now let $\{x_{k_i}\} \subseteq \{x_k\}$ be a subsequence converging to $\bar{x}$. Since $f = h \circ F$ is continuous at $\bar{x}$, it follows that

$$\lim_{i\to\infty} w_{k_i} = \bar{w} := (h \circ F)(\bar{x}) - h_{\min} \quad \text{and} \quad \lim_{i\to\infty} u_{k_i} = \bar{u} := \min\{\sigma, \theta\bar{w}^\alpha\}. \tag{3.30}$$

Note that, if $\bar{u} = 0$, then $\bar{w} = 0$ (so $F(\bar{x}) \in C$), and the conclusion follows trivially. Therefore, we may assume that $\bar{u} > 0$. Set $\bar{d} := \arg\min_{d\in\mathbb{R}^n} f_{\bar{x},\bar{u}}(d)$, which is well-defined as $f_{\bar{x},\bar{u}}$ is strongly convex (recalling $\bar{u} > 0$). Since, for each $i \geq 0$, by (2.1) that

$$u_{k_i}\|d_{k_i}\|^2 \leq f_{x_{k_i},u_{k_i}}(d_{k_i}) - h_{\min} \leq f_{x_{k_i},u_{k_i}}(0) - h_{\min} = w_{k_i},$$

it follows that $\{d_{k_i}\}$ is bounded. Then, applying [7, Theorem 2.4], we have that $\lim_{i\to+\infty} \Delta_{k_i} = 0$, and so

$$(h \circ F)(\bar{x}) = \lim_{i\to\infty} (h \circ F)(x_{k_i}) = \lim_{i\to\infty} f_{x_{k_i},u_{k_i}}(d_{k_i}). \tag{3.31}$$

Furthermore, by the definition of $f_{\bar{x},\bar{u}}$, one checks that

$$f_{x_{k_i},u_{k_i}}(d_{k_i}) - f_{\bar{x},\bar{u}}(\bar{d}) \leq h(F(x_{k_i}) + F'(x_{k_i})\bar{d}) - h(F(\bar{x}) + F'(\bar{x})\bar{d}) + (u_{k_i} - \bar{u})\|\bar{d}\|^2.$$

This, together with (3.30), implies that $\lim_{i\to\infty} f_{x_{k_i},u_{k_i}}(d_{k_i}) \leq f_{\bar{x},\bar{u}}(\bar{d})$ as h, F, and F' are respectively continuous at the reference points. Hence, combining this with (3.31), we see that $f_{\bar{x},\bar{u}}(0) = (h \circ F)(\bar{x}) \leq \min_{d\in\mathbb{R}^n} f_{\bar{x},\bar{u}}(d)$ (noting the choice of $\bar{d}$), and so $0 \in F'(\bar{x})^\top \partial h(F(\bar{x}))$ by the optimality condition (applied to the LP-function $f_{\bar{x},\bar{u}}$). The proof is complete. □

Proof of Theorem 3. As explained in the proof for [19, Theorem 18] (see lines 1-12 on page 1220 in [19]), it suffices to verify that there exists $\delta > 0$ such that

$$x_k \in \mathbf{B}(\bar{x}, \delta) \quad \Longrightarrow \quad t_k = 1$$

for each $k \geq 0$. Suppose on the contrary that, there exist a sequence $\{\delta_i\} \subseteq (0,1)$ with $\delta_i \downarrow 0$ and a subsequence $\{k_i\} \subseteq \mathbb{N}$ such that $x_{k_i} \in \mathbf{B}(\bar{x}, \delta_i)$ and $t_{k_i} \neq 1$. Then, by the definition of t_{k_i},

$$(h \circ F)(x_{k_i} + d_{k_i}) - (h \circ F)(x_{k_i}) > \lambda \left(h(F(x_{k_i}) + F'(x_{k_i}) d_{k_i}) + u_{k_i} \|d_{k_i}\|^2 - (h \circ F)(x_{k_i}) \right) \tag{3.32}$$

holds for each k_i, and by (2.5),

$$x_{k_i} \to \bar{x}, \quad F(x_{k_i}) \to F(\bar{x}), \quad \mathrm{d}(F(x_{k_i}), C) \to 0 \quad \text{and} \quad \mathrm{d}(x_{k_i}, X^*) \to 0. \tag{3.33}$$

To complete the proof, we only need to show that $\lambda > 1$ to reach a contradiction because $\lambda \in (0,1)$ by the choice of λ in Algorithm GALP.

Below, we will only verify the case when $s \geq 1$ such that assumption (2.7) in Theorem 2 holds and $\frac{p-2}{s} < \alpha \leq \frac{2s-2}{p}$ as the proof for the other case is similar to that presented in the proof for [19, Theorem 18] (see pages 1218-1219). To do this, let $\eta, c, \bar{\delta}$ and $\hat{\delta}$ be given at the beginning of the proof for Lemma 1. Then, regarding each x_{k_i} as an initial point of Algorithm ALP, one has that (3.16) and (3.19) hold, and that, if both x_{k_i} and $x_{k_i} + d_{k_i}$ belong to $\mathbf{B}(\bar{x}, \hat{\delta})$, then (3.28) holds with x_{k_i} and $x_{k_i} + d_{k_i}$ in place of x_k and x_{k+1}, and so the following implication holds for each k_i:

$$x_{k_i}, x_{k_i} + d_{k_i} \in \mathbf{B}(\bar{x}, \hat{\delta}) \quad \Longrightarrow \quad (h \circ F)(x_{k_i} + d_{k_i}) - h_{\min} \leq c\mathrm{d}^{pq_{\alpha,s}}(x_{k_i}, X^*), \tag{3.34}$$

as $\left(c_2^{\frac{1}{s}} + \frac{L^2}{2} c_1^2 \right)^s \leq c$ (see (3.20)). Then, from (3.16) and (3.33), it follows that $\|d_{k_i}\| \to 0$. Thus, without loss of generality, we may assume that $x_{k_i}, x_{k_i} + d_{k_i} \in \mathbf{B}(\bar{x}, \hat{\delta})$ for each $i \geq 0$, and so $h_{\min} + c\mathrm{d}^{pq_{\alpha,s}}(x_{k_i}, X^*) \geq (h \circ F)(x_{k_i} + d_{k_i})$ by (3.34). This, together with (3.32), implies that

$$\begin{aligned} h_{\min} - (h \circ F)(x_{k_i}) + c\mathrm{d}^{pq_{\alpha,s}}(x_{k_i}, X^*) &> \lambda \left(h(F(x_{k_i}) + F'(x_{k_i}) d_{k_i}) + u_{k_i} \|d_{k_i}\|^2 - (h \circ F)(x_{k_i}) \right) \\ &\geq \lambda \left(h_{\min} + u_{k_i} \|d_{k_i}\|^2 - (h \circ F)(x_{k_i}) \right). \end{aligned}$$

Hence,

$$(1 - \lambda)(h_{\min} - (h \circ F)(x_{k_i})) + c\mathrm{d}^{pq_{\alpha,s}}(x_{k_i}, X^*) \geq \lambda u_{k_i} \|d_{k_i}\|^2 > 0. \tag{3.35}$$

On the other hand, it follows from (3.19) that

$$(1 - \lambda)(h_{\min} - (h \circ F)(x_{k_i})) + c\mathrm{d}^{pq_{\alpha,s}}(x_{k_i}, X^*) \leq (\lambda - 1)\eta \mathrm{d}^p(x_{k_i}, X^*) + c\mathrm{d}^{pq_{\alpha,s}}(x_{k_i}, X^*).$$

Combining this with (3.35) yields that $0 < (\lambda - 1)\eta + c\mathrm{d}^{p(q_{\alpha,s}-1)}(x_{k_i}, X^*)$. Passing to the limit and noting that $\mathrm{d}(x_{k_i}, X^*) \to 0$ by (3.33), we have that $(\lambda - 1)\eta > 0$ and so $\lambda > 1$, completing the proof. □

We end this section with the following remark about the applications of results obtained in the previous sections to the convex inclusion problem.

Remark 5 We consider the following convex inclusion problem:

$$F(x) \in Q, \tag{3.36}$$

where $Q \subseteq \mathbb{R}^m$ is a closed convex set, and as assumed in the preceding sections, $F : \mathbb{R}^n \to \mathbb{R}^m$ is continuously differentiable. It can be cast into framework (1.1) with the outer convex function $h : \mathbb{R}^m \to \mathbb{R}$ defined by

$$h(y) := \frac{1}{p}\mathrm{d}^p(y, Q) \quad \text{for each } y \in \mathbb{R}^m, \tag{3.37}$$

where $p \geq 1$. Clearly, $h_{\min} = 0$; $(h - h_{\min})^{\frac{1}{p}}$ is (globally) Lipschitz on $\mathbb{R}^m$, and each $y \in Q$ is a (global) weak sharp minimizer of order p for h. Moreover, the corresponding LP-function $f_{x,u}$ for fixed $x \in \mathbb{R}^n$ and $u \geq 0$ in (2.1) is reduced to

$$f_{x,u}(d) := \frac{1}{p}\mathrm{d}^p(F(x) + F'(x)d, Q) + u\|d\|^2 \quad \text{for each } d \in \mathbb{R}^n. \tag{3.38}$$

For the remainder of this remark, we assume that $p > 1$. Then, thanks to the first-order optimality condition, solving the subproblem $\min\limits_{d\in\mathbb{R}^n} f_{x,u}(d)$ in ALP-type algorithms is equivalent to solve the following nonlinear equations:

$$G_{x,u}(d) := f'_{x,u}(d) = \mathrm{d}^{p-2}(F(x) + F'(x)d, Q)F'(x)^\top(\mathbb{I} - P_Q)(F(x) + F'(x)d) + 2ud = 0,$$

where $\mathbb{I}$ denotes the identity operator. This motivates us to propose the following algorithm for solving the convex inclusion problem (3.36).

Algorithm 1. Choose initial point $x_0 \in \mathbb{R}^n$ and set $k := 0$.

Step 1. Calculate $w_k := \frac{1}{p}\mathrm{d}^p(F(x_k), Q)$, $u_k := \min\{\sigma, \theta w_k^\alpha\}$, and choose $0 \leq \epsilon_k \leq \theta w_k^\rho$.
Step 2. If $G_{x_k,u_k}(0) = 0$, then stop.
Step 3. If $\|G_{x_k,u_k}(0)\| \leq \sqrt{2u_k\epsilon_k}$, then set $\epsilon_k := \theta\epsilon_k$ and go back to Step 3.
Step 4. Calculate d_k by approximately solving $G_{u_k,x_k}(d) = 0$ such that $\|G_{x_k,u_k}(d_k)\| \leq \sqrt{2u_k\epsilon_k}$.
Step 5. Set $x_{k+1} := x_k + d_k$ and update $k := k + 1$. Go back to Step 1.

The corresponding globalized version of Algorithm 1 is as follows.

Algorithm 2. Choose initial point $x_0 \in \mathbb{R}^n$, $\gamma \in (0, 1)$, $\lambda \in (0, 1)$ and set $k := 0$.

Step 1. Generate d_k by Steps 1-4 in Algorithm 1.
Step 2. Find t_k which is the maximum value of γ^i for $i = 0, 1, \cdots$, such that

$$\tfrac{1}{p}\mathrm{d}^p(F(x_k + \gamma^i d_k), Q) - \tfrac{1}{p}\mathrm{d}^p(F(x_k), Q) \leq \lambda\gamma^i \left(f_{x_k,u_k}(d_k) - \tfrac{1}{p}\mathrm{d}^p(F(x_k), Q)\right).$$

Step 3. Set $x_{k+1} := x_k + t_k d_k$ and update $k := k + 1$. Go back to Step 1.

Below we present the convergence properties of Algorithms 1 and 2 by virtue of the established convergence theorems for ALP-type algorithms.

(a) One can check that the sequence generated by Algorithm 1 can also be regarded as a sequence generated by Algorithm ALP with h given by (3.37).

(b) Suppose that $\bar{x} \in \mathbb{R}^n$ satisfies $F(\bar{x}) \in Q$ and is a quasi-regular point for inclusion (3.36). Suppose further that F' is locally Lipschitz around $\bar{x}$. Then, any sequence $\{x_k\}$ generated by Algorithm 1 converges locally to some point x^* satisfying $F(x^*) \in Q$ at a rate of q_α if $\alpha > \max\{p-2, 2-\frac{2}{p}\}$ and at a rate of $q_{\alpha,p}$ if $1-\frac{2}{p} < \alpha \le 2-\frac{2}{p}$. In fact, this result can be verified directly by assertion (a) of this remark and Corollary 1 (applied to the function h defined in (3.37); noting that the third assumption in (2.5) and assumptions (2.7) and (2.8) in Theorem 2 hold as noted in the beginning of this remark).

(c) Let $\{x_k\}$ be a sequence generated by Algorithm 2. Suppose that $\{x_k\}$ has an accumulation point $\bar{x} \in \mathbb{R}^n$ satisfying $F(\bar{x}) \in Q$ such that $\bar{x}$ is a quasi-regular point for inclusion (3.36) and F' is locally Lipschitz around $\bar{x}$. Then, $\{x_k\}$ converges to $\bar{x}$ at a rate of q_α if $1 < p < 2$ and $\alpha > 2-\frac{2}{p}$ and at a rate of $q_{\alpha,p}$ if $1-\frac{2}{p} < \alpha \le 2-\frac{2}{p}$. In fact, this result follows directly from assertion (a) of this remark and Theorem 3.

Note that under the quasi-regularity condition, the convex composite optimization problem (1.1) can be represented as the inclusion problem $F(x) \in C := \{y \in \mathbb{R}^m : h(y) = h_{\min}\}$ (i.e., (3.36) with $Q := C$). However, it is still valuable to study the efficient algorithms under the framework of the convex composite optimization problem (1.1). Indeed, from a practical point of view, it might be inefficient to implement Algorithms 1 and 2 because the "complete" data of C and the projection to C are not easy to calculate in general (although it is implementable to find an element in C by convex optimization algorithms). Moreover, based on the convergence theorems in section 3, the convergence rates of Algorithms ALP and GALP heavily depends on the selection of the outer function h.

4 Applications and numerical experiments

In this section, we discuss applications to the nonnegative inverse eigenvalue problem (NIEP) and the wireless sensor network (WSN) localization problem by virtue of the convex inclusion framework, and compare the numerical performance of Algorithm GALP (Algorithm 2) with [19, Algorithm 17] (rewritten as Algorithm GCLP for convenience), as well as effective algorithms in applications. Numerical experiments are implemented in MATLAB R2015b and the hardware environment is Intel Core i7 3740QM, @2.70 GHz (8 CPUs), 16.00 GB of RAM.

In the numerical experiments, we set the parameters $\sigma = 0.005$, $v_k \equiv \frac{1}{2\sigma} = 100$, $\gamma = 0.9$, $\theta = 0.5$, $\lambda = c = 0.9$, $\alpha = 1$, $\rho = 2$ for Algorithms GALP and GCLP (if parameters are needed), with which Algorithm GCLP performs well as in [19]. The semismooth Newton method [31] and Newton method are applied to solve the subproblems in Algorithm GALP/GCLP when $p = 2$ and $p > 2$, respectively. The stopping criteria of Algorithms GCLP and GALP are listed as follows.

- Outer iteration: the number of iterations is greater than 100, or the residual (RES)/root mean square distance (RMSD) meets accuracy requirements.
- Inner iteration: the number of iterations is greater than 50 or
  - Algorithm GCLP: $G_{x_k,\frac{1}{2v}}(d) < \max\{\theta\|d_{k-1}\|^\rho, 10^{-(2p+4)}\}$;
  - Algorithm GALP: $G_{x_k,u_k}(d) < \max\{\sqrt{2\theta u_k w_k^\rho}, 10^{-(2p+4)}\}$.

4.1 Problem I: NIEP

The research about the NIEP owns a long history since 1940s [35,29,30] (one can also refer to the review articles [12,13] for more details) and nonnegative matrices play a significant role in various areas such as control design, linear complementarity problems, Markov chains and graph theory; see [1,2,17] and references therein. In this subsection, we investigate the NIEP formulated as follows: Given a n-tuple $\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ as a realizable spectrum of nonnegative matrices, find a nonnegative matrix $X \in \mathbb{R}^{n\times n}$ whose eigenvalues are $\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$.

The following notation is used in this subsection. Let $\mathbb{R}^{n\times n}$ be the set of all $n \times n$ real matrices and $\mathbb{R}_+^{n\times n}$ (resp. $\mathbb{R}_-^{n\times n}$) be the nonnegative (resp. nonpositive) orthant of $\mathbb{R}^{n\times n}$. Let $\mathbb{I}_{n\times n}$ be the $n \times n$ identity matrix and $\mathcal{O}(n)$ denote the set of all $n \times n$ orthogonal matrices, i.e., $\mathcal{O}(n) := \{U \in \mathbb{R}^{n\times n} : U^\top U = \mathbb{I}_{n\times n}\}$, where $U^\top$ denotes the transpose of the matrix U.

For the set of prescribed $\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$, we assume

$$\lambda_{2i-1} := a_i + b_i\sqrt{-1},\ \lambda_{2i} := a_i - b_i\sqrt{-1} \quad \text{for each } 1 \le i \le s; \text{ and } \lambda_i \in \mathbb{R} \quad \text{for each } i \ge 2s+1,$$

where $a_i, b_i \in \mathbb{R}$ with $b_i \neq 0$ for each $1 \le i \le s$. Define a block diagonal matrix $\Lambda \in \mathbb{R}^{n\times n}$ by

$$\Lambda := \text{blkdiag}(\lambda_1^{[2]}, \cdots, \lambda_s^{[2]}, \lambda_{2s+1}, \cdots, \lambda_n) \quad \text{with each } \lambda_i^{[2]} := \begin{bmatrix} a_i & b_i \\ -b_i & a_i \end{bmatrix},$$

and set $\mathcal{I} := \{(i,j) : i \ge j \text{ or } \Lambda_{ij} \neq 0\}$ and $\mathcal{V} := \{V \in \mathbb{R}^{n\times n} : V_{ij} = 0 \text{ for each } (i,j) \in \mathcal{I}\}$. Define further the set $Q \subseteq \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$ and the mapping $F : \mathbb{R}^{n\times n} \times \mathcal{V} \to \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$ by

$$Q := \mathbb{R}_+^{n\times n} \times \{0\} \text{ and } F(U,V) := (U(\Lambda+V)U^\top, UU^\top - \mathbb{I}_{n\times n}) \quad \text{for each } U \in \mathbb{R}^{n\times n}, V \in \mathcal{V}, \tag{4.1}$$

respectively. Then, with the help of real Schur decomposition (see [18, Theorem 7.4.1]), one checks as done in [38, section 2.1] that $X := U(\Lambda + V)U^\top$ is a solution of the NIEP if and only if $(U,V) \in \mathbb{R}^{n\times n} \times \mathcal{V}$ is a solution of problem (3.36) on $\mathbb{R}^{n\times n} \times \mathcal{V}$ with F and Q given by (4.1).

In the numerical experiments, the nonnegative matrix is randomly generated via a uniform distribution in $[0,1]$ and then the target eigenvalues are calculated. One initial point U_0 is randomly generated and the other is given by $V_0 := W \odot V$, where V is the Schur decomposition of a real random matrix and $W \in \mathbb{R}^{n\times n}$ is given by

$$W_{ij} = \begin{cases} 0, & \text{if } (i,j) \in \mathcal{I}, \\ 1, & \text{otherwise.} \end{cases}$$

We use the conjugate gradient (CG) method to solve nonlinear equations associated to the semismooth/smooth Newton method for solving the subproblem of Algorithm GCLP/GALP (Algorithm 2). The stopping criterion is the residual of CG iteration is less than $10^{-(2p+4)}$ or the number of iterations is greater than 1000. The accuracy of each algorithm is evaluated by the residual (RES) of the associated convex inclusion problem:

$$\text{RES} := \sqrt{\|[U_*(\Lambda + V_*)U_*^\top]_-\|_F^2 + \|U_*U_*^\top - \mathbb{I}_{n\times n}\|_F^2},$$

where U_* and V_* form the Schur decomposition of the solution estimated by the algorithm. The algorithms are set to stop whenever RES<1e-4.

The first experiment aims to compare the numerical performance of Algorithms GCLP and GALP on solving the NIEP. Table 1 shows the numerical results (averaged by 10 random trials), in terms of CPU time (in seconds) and RES, of Algorithms GCLP and GALP when $p = 2$ and $p = 4$, respectively. Figure 1(a) plots the variation of RES of Algorithms GCLP and GALP respectively along with the number of outer iterations in a random trial when $p = 2$ and $n = 100$. It is observed from Table 1 that Algorithm GALP performs better on CPU time than Algorithm GCLP when $p = 2$; particularly when $n \geq 100$, Algorithm GALP only costs about half CPU time of Algorithm GCLP to approach a precise solution of the NIEP. Algorithm GALP can obtain a solution of the NIEP when $p = 4$ while Algorithm GCLP fails. This is consistent with the advantage of Algorithm GALP on convergence theory that Algorithm GALP converges at a superlinear rate when $p \geq 1$ (see Remark 5(c)) while Algorithm GCLP is guaranteed to converge only when $1 \leq p \leq 2$ (see [19, Theorem 12]). It is also observed from Figure 1(a) that Algorithm GALP converges faster than Algorithm GCLP. This is consistent with the convergence theory that Algorithm GALP owns a quadratic convergence rate (see Remark 5(c)), while Algorithm GCLP only converges linearly (see [19, Theorem 12]) when $p = 2$.

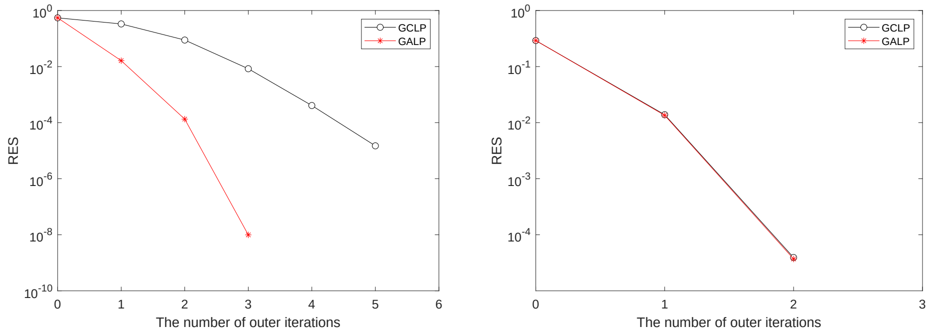
Table 1 The numerical results of Algorithms GCLP and GALP for the NIEP ($p = 2$ or 4).

Algorithm	GCLP				GALP			
	$p = 2$		$p = 4$		$p = 2$		$p = 4$	
n	CPU time	RES	CPU time	RES	CPU time	RES	CPU time	RES
10	0.0849 s	6.5e-5	N/A[3]		0.0639 s	3.8e-5	0.1601 s	4.2e-5
50	1.7763 s	2.9e-5	N/A		1.3076 s	4.7e-6	27.741 s	2.5e-5
100	17.422 s	1.1e-5	N/A		8.6562 s	8.0e-5	148.77 s	8.5e-5
150	70.795 s	7.8e-5	N/A		43.408 s	1.1e-6	459.96 s	6.3e-5
200	163.46 s	4.3e-5	N/A		88.427 s	5.9e-7	1260.5 s	1.1e-5

On the other hand, Table 2 and Figure 1(b) show the corresponding comparison results of Algorithms GCLP and GALP when $p = 1$, for which we use the CVX[4] instead of the

[3] N/A means that the algorithm cannot approach the solution within the tenfold CPU time of that cost by Algorithm GALP.

[4] CVX, designed by Michael Grant and Stephen Boyd, is a MATLAB-based modeling system for convex optimization. Detailed information is available from the website http://cvxr.com/cvx/.



(a) $p = 2$ and $n = 100$ (b) $p = 1$ and $n = 20$

Fig. 1 The numerical results of Algorithms GCLP and GALP for the NIEP.

Table 2 The numerical results of Algorithms GCLP and GALP for the NIEP ($p = 1$).

Algorithm	GCLP		GALP	
n	CPU time	RES	CPU time	RES
10	4.3162 s	2.7e-6	4.2342 s	2.7e-6
20	29.632 s	8.6e-6	27.872 s	8.5e-6
30	197.07 s	1.9e-6	191.65 s	1.9e-6

semismooth Newton method to solve the subproblem since the semismooth Newton method only performs well when $p \geq 2$ (see also the discussion in [19, Remark 31(a)]). While the CVX is not efficient for solving large-scale problems, here we only consider the results when $n \leq 30$. From Table 2 and Figure 1(b), we know that Algorithms GCLP and GALP have the similar performance on CPU time and RES, which is consistent with [19, Theorem 18] and Theorem 3 that both Algorithms GCLP and GALP converge quadratically when $p = 1$. (Much CPU time are consumed by the CVX in solving the subproblems.) Similar results can be observed for all $p \in [1, 2)$ since both Algorithms GCLP and GALP have superlinear convergence rate in this case.

Table 3 The numerical results of the RINC and Algorithm GALP ($p = 2$) for the NIEP.

Eigenvalues	dense matrices				1% sparse matrices			
Algorithm	RINC		GALP		RINC		GALP	
n	CPU time	RES	CPU time	RES	CPU time	RES	CPU time	RES
10	0.01 s	9.5e-5	0.05 s	8.6e-6	1.22 s	9.9e-5	0.21 s	9.5e-7
20	0.03 s	5.9e-6	0.12 s	6.0e-6	21.7 s	9.7e-5	2.73 s	3.7e-7
50	0.21 s	4.5e-5	1.05 s	1.2e-7	N/A		5.76 s	7.7e-6
80	0.52 s	1.4e-6	4.71 s	5.3e-7	N/A		11.7 s	3.1e-5
100	1.02 s	2.4e-5	8.15 s	2.1e-7	N/A		18.4 s	1.6e-7

The Riemannian inexact Newton-CG method (RINC) [38] is one of the most popular and efficient algorithms for solving the NIEP. The second experiment is to compare the numerical performance of the RINC and Algorithm GALP. Table 3 lists the numerical results (averaged by 10 random trials) of the RINC and Algorithm GALP for the dense random matrices and 1% sparse random matrices, respectively. It is revealed from the first part of Table 3 that the RINC is faster than Algorithm GALP in the dense matrix case; while it is illustrated from the second part of Table 3 that Algorithm GALP is much more effective than the RINC in the sparse matrix case. This is because the convergence theory of the RINC requires the surjective assumption on the differential of a smooth mapping (associated to the NIEP) at the accumulation point generated by the RINC (see [38, Assumption 1]), which is likely to be satisfied in the dense matrix case but would fail in the sparse matrix case (see [38, Remark 3.9]); while the assumption for the convergence theorem of Algorithm GALP is less restrictive than the RINC. In a word, it is revealed from the numerical results that Algorithm GALP is an efficient and robust algorithm for the NIEP.

4.2 Problem II: WSN localization

There is an increasing use of ad hoc wireless sensor networks for monitoring the environmental information across an entire physical space. Typical networks of this type consist of quantities of wireless sensors deployed in a geographical area which are able to communicate neighbors within a limited radio range, and the WSN localization problem is to estimate the positions of unknown sensors in a network by using the given incomplete pairwise distance measurements; see [4,19,26] and references therein.

Let $\{x_1,\ldots,x_n\}\subseteq\mathbb{R}^2$ and $\{a_1,\ldots,a_m\}\subseteq\mathbb{R}^2$ denote the sets of sensors and anchors (a small quantity of sensors whose locations are known), respectively. Let N_e (resp. M_e) consist of all indices of pairs (i,j) (resp. (k,j)) where $j=i+1,\ldots,n$ (resp. $j=1,\ldots,n$) such that the Euclidean distance measure d_{ij} between x_i and x_j (resp. $\bar{d}_{kj}$ between a_k and x_j) is within the radio range (denoted by R); otherwise, the distances between sensors or anchors are larger than R. Then the WSN localization problem is to find $\{x_1,\ldots,x_n\}$ satisfying:

$$\begin{aligned} &\|x_i-x_j\|^2=d_{ij}^2, \quad \|a_k-x_j\|^2=\bar{d}_{kj}^2, \quad (i,j)\in N_e,\,(k,j)\in M_e,\\ &\|x_i-x_j\|^2>R^2, \quad \|a_k-x_j\|^2>R^2, \quad (i,j)\notin N_e,\,(k,j)\notin M_e. \end{aligned} \tag{4.2}$$

In general, the WSN localization problem (4.2) is NP-hard [33]. By neglecting all inequality constraints in (4.2), many works focus on the following relaxation model

$$\|x_i-x_j\|^2=d_{ij}^2, \quad \|a_k-x_j\|^2=\bar{d}_{kj}^2, \quad (i,j)\in N_e,\,(k,j)\in M_e; \tag{4.3}$$

see [4,26] and references therein. Let $x:=(x_1,...,x_n)\in\mathbb{R}^{2\times n}$ and define

$$\begin{aligned} g_{i,j,1}(x)&:=R^2-\|x_i-x_j\|^2, \quad (i,j)\notin N_e,\\ g_{i,j,2}(x)&:=R^2-\|a_i-x_j\|^2, \quad (i,j)\notin M_e,\\ \bar{g}_{i,j,1}(x)&:=\|x_i-x_j\|^2-d_{ij}^2, \quad (i,j)\in N_e,\\ \bar{g}_{i,j,2}(x)&:=\|a_i-x_j\|^2-\bar{d}_{ij}^2, \quad (i,j)\in M_e. \end{aligned}$$

Write

$$g(x) := ((g_{i,j,1}(x))_{(i,j)\notin N_e}, (g_{i,j,2}(x))_{(i,j)\notin M_e}) \quad \text{and} \quad \bar{g}(x) := ((\bar{g}_{i,j,1}(x))_{(i,j)\in N_e}, (\bar{g}_{i,j,2}(x))_{(i,j)\in M_e}).$$

Thus, the problems (4.2) and (4.3) are viewed as a convex inclusion problem (3.36) with F and Q respectively given by

$$Q := \mathbb{R}_-^{\frac{1}{2}n(n-1)+mn-|N_e|-|M_e|} \times \{0\} \subseteq \mathbb{R}^{\frac{1}{2}n(n-1)+mn}, \quad F(x) := (g(x), \bar{g}(x)) \quad \text{for each } x \in \mathbb{R}^{2\times n},$$

and

$$Q := \{0\} \subseteq \mathbb{R}^{|N_e|+|M_e|}, \quad F(x) := \bar{g}(x) \quad \text{for each } x \in \mathbb{R}^{2\times n},$$

where $|\cdot|$ denotes the cardinality of a set. We apply Algorithm 2 to solve the convex inclusion problems (4.2) and (4.3), which are abbreviated by Algorithm GALP and Algorithm GALP-R, respectively.

In the numerical experiments, we set $p = 2$ (unless otherwise specified) and the sensors and anchors are randomly placed via a uniform distribution in the unit square $[-0.5, 0.5]^2$. The performance of algorithms is evaluated by:

- (Accuracy) The root mean square distance (RMSD):

$$\text{RMSD} := \frac{1}{\sqrt{n}} \left(\sum_{i=1}^{n} \|s_i - x_i\|^2 \right)^{\frac{1}{2}},$$

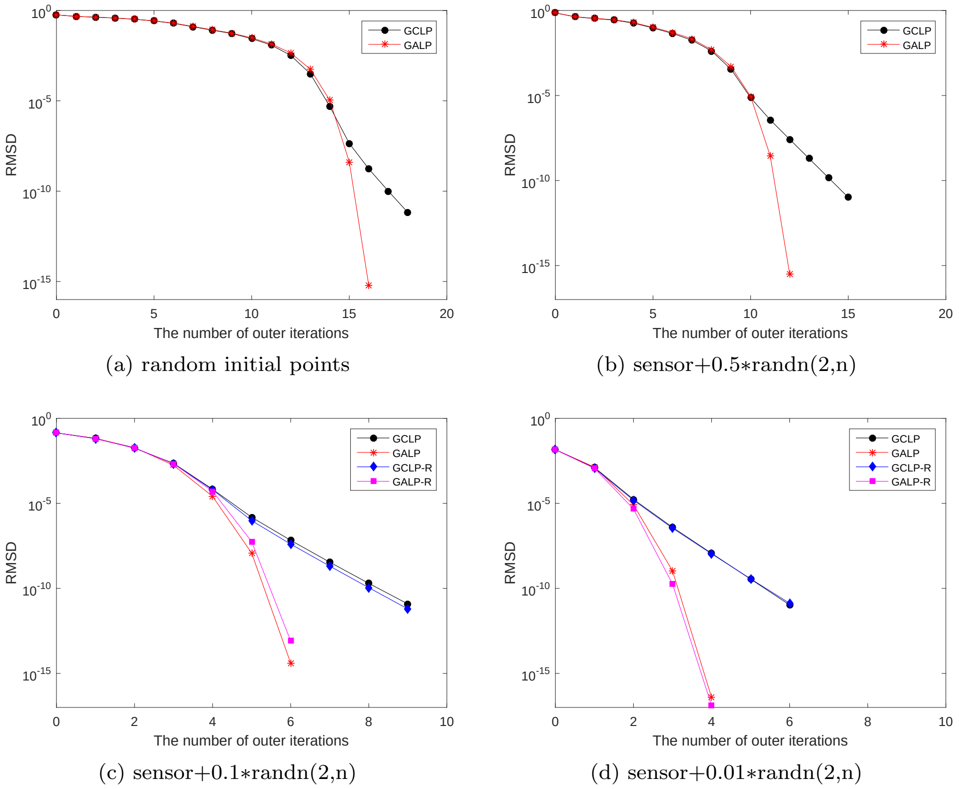
  where $\{x_i\}$ and $\{s_i\}$ denote the locations of the estimated sensors and experiment data, respectively.
- (Speed) The CPU time cost by the algorithm, which is set to stop whenever RMSD<1e-10.
- (Stability) The successful rate: the ratio of successful estimating, in which RMSD<1e-5.

In the following subsections, we first show the convergence rate of Algorithm GALP compared with Algorithm GCLP and introduce an initialization strategy of multidimensional scaling. Next, we compare the performance of Algorithm GALP with other approaches by discussing the results when the scale of the network, the radio range and the number of anchors change respectively.

4.2.1 The performance of Algorithm GALP

This subsection aims to show the numerical performance of Algorithm GALP on solving the WSN localization problem and by comparing with Algorithm GCLP. In our numerical results, Algorithm GALP can successfully localize the positions of the sensors in a WSN (of 200 sensors, 20 anchors, and the radio range being 0.3) within 1.5 seconds. For this WSN, Algorithm GCLP can also achieve an accurate solution within 1.7 seconds, which is a little slower than Algorithm GALP.

We conduct further experiments to compare the capability of Algorithms GCLP and GALP to solve the WSN localization problem (of 200 sensors, 20 anchors, and the radio range being 0.3). Figure 2 plots the variation of RMSD of the estimation along with the number of the outer iterations, and Table 4 lists the CPU time (in seconds) cost by Algorithms GCLP and GALP with different initial points. Three observations are taken from Figure 2 and Table 4: (i) Algorithms GCLP and GALP converge at a linear rate and a quadratic rate, respectively, which is consistent with the convergence theory ([19, Theorem 12] and Remark 5(c)). (ii) Although Algorithm GALP owns a faster convergence rate than Algorithm GCLP, their advantage on CPU time and the number of iterations is limited when starting from a random initial point because the major cost of Algorithm GCLP/GALP is at the initial stage of iterations. (iii) If a good initial point or a good (and fast) initialization strategy is provided, Algorithm GALP will be much faster than Algorithm GCLP.



(a) random initial points

(b) sensor+0.5∗randn(2,n)

(c) sensor+0.1∗randn(2,n)

(d) sensor+0.01∗randn(2,n)

Fig. 2 The numerical results of Algorithms GCLP and GALP with different initial points.

The multidimensional scaling (MDS) [21] is a popular and fast approach for the WSN localization problem. When applying the MDS to solve problem (4.3) (of 200 sensors, 20 anchors, radio range = 0.3 and initial point: random), the result is:

$$\text{RMSD} = \text{1e-3}, \qquad \text{CPU time} = 0.3 \text{ second}.$$

Table 4 The CPU time of Algorithm GCLP/GALP with different initial points for the WSN localization problem (200 sensors, 20 anchors, radio range=0.3).

Algorithm	GCLP	GALP	GCLP-R	GALP-R
random initial points	1.7 s	1.5 s	0.6 s	0.5 s
sensor+0.5∗randn(2,n)	1.5 s	1.2 s	0.6 s	0.5 s
sensor+0.1∗randn(2,n)	1.0 s	0.6 s	0.4 s	0.2 s
sensor+0.01∗randn(2,n)	0.6 s	0.3 s	0.3 s	0.1 s

This shows that the MDS is a fast algorithm to estimate the locations of WSN but of low-precision, which meets the demand for the fast initialization strategy for Algorithm GALP. Hence, we introduce an MDS-GALP solver for the WSN localization problem, in which the MDS is employed to provide an initial point and then Algorithm GALP is used to solve problem (4.2) or (4.3) starting from this initial point.

4.2.2 Comparison among various algorithms

The purpose of this subsection is to compare the numerical performance of the MDS-GALP solver with several effective algorithms for the WSN localization problem. The semidefinite programming (SDP) is an effective approach to formulate the WSN localization problem [5]. One of the most popular and efficient tools for solving the WSN localization problem is the semidefinite relaxation (SDR) technique that relaxes (4.3) into an SDP; see, e.g., [4,26]. In order to facilitate the reading of the numerical results, the abbreviations of algorithms are listed in Table 5.

Table 5 List of the algorithms for solving the WSN localization problem.

Abbreviations	Algorithms
MDS	**M**ulti**D**imensional **S**caling method (for solving (4.3)).
SDR[5]	**S**emi**D**efinite **R**elaxation method (for solving (4.3)).
GCLP	[19, Algorithm 17] (for solving (4.2)).
GALP	Algorithm 2 (for solving (4.2)).
GCLP-R	Algorithm GCLP (for solving (4.3)).
GALP-R	Algorithm GALP (for solving (4.3)).
MDS-GCLP	Algorithm GCLP with initial points given by MDS (for solving (4.2)).
MDS-GALP	Algorithm GALP with initial points given by MDS (for solving (4.2)).
MDS-GCLP-R	Algorithm GCLP-R with initial points given by MDS (for solving (4.3)).
MDS-GALP-R	Algorithm GALP-R with initial points given by MDS (for solving (4.3)).

In this subsection, we compare the MDS-GALP/GALP-R with several effective algorithms, including the MDS and SDR, for the WSN localization problem. For iterative algorithms, the stopping criteria are set as follows: the SDR's are directly determined by the

[5] It is solved by SNLSDP– a MATLAB software for the WSN localization problem whose code and description are available in http://www.math.nus.edu.sg/∼mattohkc/SNLSDP.html.

MATLAB software SNLSDP, and the (MDS based) GCLP and GALP's are the same to those set at the beginning of section 5 with RMSD<1e-10.

The first experiment is to compare the numerical performance of these algorithms on a random WSN localization problem of 200 sensors, 20 anchors, and the radio range being 0.3. The numerical results of these algorithms, in terms of RMSD, CPU time (in seconds) and successful rate, with random initial points and initial points given by the MDS are listed in Tables 6 and 7, respectively. In Table 7, the rows of "CPU time" and "CPU time (+MDS)" record the CPU time cost by the GCLP/GALP and that plus the MDS initialization process, respectively. The following observations are indicated from Tables 6 and 7:

(i) The GCLP and GALP achieve a more precise solution and consume less CPU time than the SDR.

(ii) The GCLP and GALP consume more CPU time than the GCLP-R and GALP-R, because the GCLP and GALP are designed to solve the full version of problem (4.2), whose number of constraints is more than double that of the relaxation problem (4.3) solved by the GCLP-R and GALP-R.

(iii) When random initial points are used, all these algorithms cannot obtain the robust successful estimation within 2 seconds; while, the GCLP and GALP own more robust 3s-successful[6] rate than the GCLP-R and GALP-R, as well as the MDS and SDR. This is benefited from that more constraints information is involved in the full version of problem (4.2) than the relaxation problem (4.3).

(iv) When good initial points are given by the MDS, GCLP/GCLP-R and GALP/GALP-R are faster than the ones with random initial points; particularly, the GALP/GALP-R are much faster than the GCLP/GCLP-R.

(v) When good initial points are given by the MDS, GCLP/GCLP-R and GALP/GALP-R all have robust 1s-successful rate. In a word, the MDS-GALP-R is the best one on accuracy, speed and robustness.

Table 6 The numerical results of several algorithms for a WSN localization problem (200 sensors, 20 anchors, radio range=0.3 and initial points: random).

Algorithm	MDS	SDR	GCLP	GALP	GCLP-R	GALP-R
RMSD	1.0e-3	2.8e-8	1.8e-11	2.2e-13	1.6e-11	7.8e-13
CPU time	0.3 s	38.6 s	1.7 s	1.5 s	0.6 s	0.5 s
3s-S rate[6]	0%	0%	99%	99%	67%	68%
2s-S rate	0%	0%	64%	64%	67%	68%

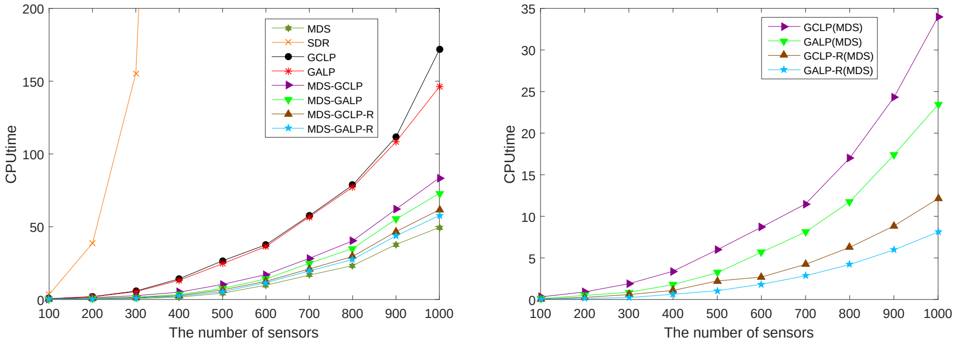
The second experiment aims to show the capability of these algorithms to solve large-scale WSN localization problems, in which the number of sensors is varied from 100 to 1000 and the number of anchors is set as 10% of sensors. Figure 3 shows the variation of CPU time (averaged by 10 random trials) consumed by these algorithms along with the

[6] The estimation is regarded as "ts-successful" if the estimated RMSD is less than 1e-5 within t seconds. "ts-S rate" denotes the ratio of "ts-successful" estimating in 100 random trials.

Table 7 The numerical results of several algorithms for a WSN localization problem (200 sensors, 20 anchors, radio range=0.3 and initial points: given by the MDS).

Algorithm	GCLP	GALP	GCLP-R	GALP-R
RMSD	1.7e-11	2.2e-13	1.4e-11	5.6e-13
CPU time	0.6 s	0.3 s	0.2 s	0.1 s
CPU time (+MDS)	0.9 s	0.6 s	0.5 s	0.4 s
1s-S rate	99%	99%	99%	99%

increasing number of sensors when random initial points and MDS initial points are used, respectively. It is demonstrated by Figure 3(a) that the CPU time of the SDR grows rapidly when the number of sensors increases and it is not available for the large-scale WSN localization problem. The GCLP/GALP is faster than the SDR, but slower than the MDS, MDS-GCLP/GCLP-R and MDS-GALP/GALP-R along with the increasing number of sensors. It is illustrated by Figure 3(b) that the MDS-GALP/GALP-R are much more efficient than the MDS-GCLP/GCLP-R, respectively. This experiment reveals that the MDS-GALP and MDS-GALP-R are suitable and efficient for solving large-scale WSN localization problems.

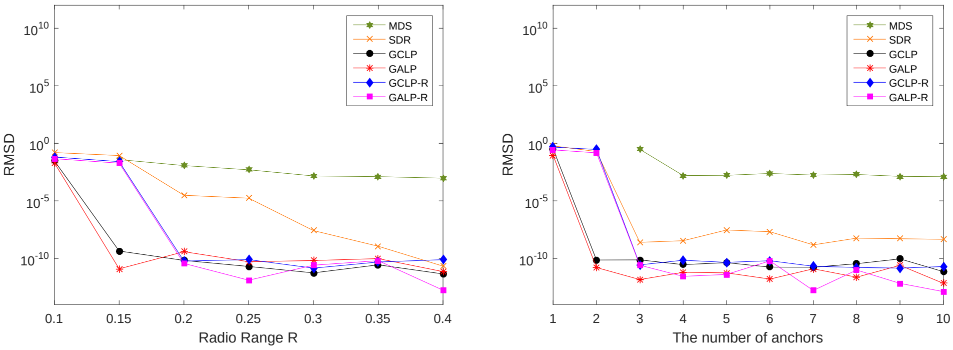


(a) Random initial points

(b) Initial points provided by the MDS

Fig. 3 The CPU time of several algorithms along with the number of sensors (radio range=0.3).

The third experiment is to exhibit the numerical performance of these algorithms when varying the circumstances (the radio range and the number of anchors) of the WSN of 200 sensors. Figure 4 shows the variation of RMSD (averaged by 10 random trials) of these algorithms when increasing the radio range from 0.1 to 0.4 and varying the number of anchors from 1 to 10, respectively. When the radio range or the number of anchors is too small, there is no enough distance information in the WSN to make the estimation effective. It is exhibited by Figure 4 that the GCLP and GALP can obtain an accurate estimation with the smallest radio range (only $R \geq 0.15$ is required) and the least number of anchors requirement (only 2 anchors are needed).



(a) RMSD - Radio range R

(b) RMSD - the number of anchors

Fig. 4 The RMSD of several algorithms along with the radio range and the number of anchors (200 sensors and initial points: random).

Finally, the conclusions of the numerical experiments on the WSN localization problem can be summarized as follows:

(i) The GALP achieves a more precise solution and takes less CPU time than the SDR and GCLP; particularly, when $p = 2$, the GALP owns a quadratic convergence rate that is faster than the linear convergence rate of the GCLP.

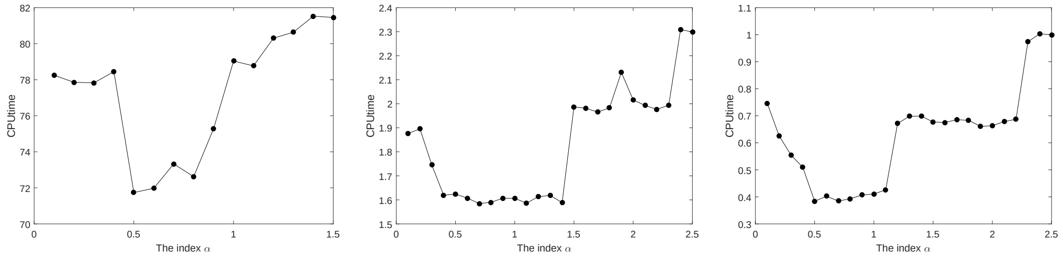
(ii) The MDS provides a good (and fast) initialization strategy for the GALP. The resulting the MDS-GALP outperforms the SDR and GCLP (based on MDS) on accuracy, speed and robustness.

(iii) The MDS-GALP and MDS-GALP-R are suitable and efficient for solving large-scale WSN localization problems with the least measurements.

4.3 The parameter setting of α in ALP-type algorithms

The major difference between CLP- and ALP-type algorithms is the adaptive stepsize, in which α is a key parameter. It is clear from the theoretical results that the convergence rate will be faster along with the increment of α, while the computational cost of subproblems will be more expensive. Hence, the parameter α in ALP-type algorithms provides a tradeoff between the convergence rate and computational cost of subproblems, and plays an important role in the numerical performance of the ALP-type algorithms.

The above experiment is to illustrate the effect of α on the numerical performance of Algorithm GALP in terms of CPU time to the NIEP when $p = 2$ and $n = 200$, and the WSN localization problems of 200 sensors, 20 anchors and the radio range being 0.3 when random initial points and MDS initial points are used, respectively. It is demonstrated by Figure 5 that the CPU time of Algorithm GALP decreases when α is small and increases when α is large due to the tradeoff between the convergence rate and the solution of subproblem. Thus, for the NIEP and the WSN localization problem with the given circumstance, we



(a) NIEP (b) WSNLP: random initial points (c) WSNLP: initialization by MDS

Fig. 5 The CPU time of Algorithm GALP along with α for the NIEP ($p = 2$ and $n = 200$)/WSN localization problem (WSNLP) (200 sensors, 20 anchors and radio range=0.3).

recommend to choose $0.5 \le \alpha \le 0.8$ and $0.5 \le \alpha \le 1$, respectively, for which the CPU time is short.

Acknowledgements The authors are grateful to the editor and the anonymous reviewers for their valuable comments and suggestions toward the improvement of this paper. Yaohua Hu's work was supported in part by the National Natural Science Foundation of China (12222112, 12071306, 11871347, 32170655), Project of Educational Commission of Guangdong Province of China (2021KTSCX103, 2019KZDZX1007), and Natural Science Foundation of Shenzhen (JCYJ20190808173603590). Chong Li's work was supported in part by the National Natural Science Foundation of China (11971429). Jinhua Wang's work was supported in part by the National Natural Science Foundation of China (grant 12171131). Xiaoqi Yang's work was supported in part by the Research Grants Council of Hong Kong (PolyU 15216518).

Appendix

Proposition A. *Let $c > 0$, $q > 1$, and let $X^* \subseteq \mathbb{R}^n$ be closed. Let $\{x_k\} \subseteq \mathbb{R}^n$ be a sequence satisfying*

$$\|x_{k+1} - x_k\| \le c\mathrm{d}(x_k, X^*) \quad \text{and} \quad \mathrm{d}(x_{k+1}, X^*) \le c\mathrm{d}^q(x_k, X^*) \quad \text{for each } k \ge 0. \tag{4.4}$$

If $\mathrm{d}(x_0, X^*) < \left(\frac{1}{c}\right)^{\frac{1}{q-1}}$, then $\{x_k\}$ converges to a point $x^* \in X^*$ at a rate of q.

Proof. Assume that $\mathrm{d}(x_0, X^*) < \left(\frac{1}{c}\right)^{\frac{1}{q-1}}$, and set $\tau := c\mathrm{d}^{q-1}(x_0, X^*)$. Then $\tau < 1$, and

$$c\mathrm{d}^{q-1}(x_k, X^*) \le \tau \quad \text{and} \quad \mathrm{d}(x_{k+1}, X^*) \le \tau\mathrm{d}(x_k, X^*) \quad \text{for each } k \ge 0 \tag{4.5}$$

because, by the second inequality of (4.4), $\mathrm{d}(x_k, X^*) \le c^{\frac{q^k-1}{q-1}}\mathrm{d}^{q^k}(x_0, X^*) = c^{\frac{q^k-1}{q-1}}\left(\frac{\tau}{c}\right)^{\frac{q^k}{q-1}} \le \left(\frac{\tau}{c}\right)^{\frac{1}{q-1}}$ for each k. In particular, we have that $\mathrm{d}(x_k, X^*) \to 0$.

Now fix $k \ge 1$. We have from (4.5) that $\mathrm{d}(x_k, X^*) \le \mathrm{d}(x_{k+1}, X^*) + \|x_{k+1} - x_k\| \le \tau\mathrm{d}(x_k, X^*) + \|x_{k+1} - x_k\|$. It follows that $\mathrm{d}(x_k, X^*) \le \frac{1}{1-\tau}\|x_{k+1} - x_k\|$. Thus, using (4.4), we check that

$$\|x_{k+1} - x_k\| \le c\mathrm{d}(x_k, X^*) \le c^2\mathrm{d}^q(x_{k-1}, X^*) \le \frac{c^2}{(1-\tau)^q}\|x_k - x_{k-1}\|^q.$$

Set $c_k := \frac{c^2}{(1-\tau)^q}\|x_k - x_{k-1}\|^{q-1}$. Then

$$c_{k+1} \le \left(\frac{c^2}{(1-\tau)^q}\right)^{q-1} \|x_k - x_{k-1}\|^{(q-1)^2} c_k \quad \text{and} \quad \|x_{k+1} - x_k\| \le c_k \|x_k - x_{k-1}\|. \tag{4.6}$$

Since $\mathrm{d}(x_k, X^*) \to 0$, it follows from (4.4) that $\|x_k - x_{k-1}\| \to 0$, and so $c_k \to 0$. This, together with (4.6) implies that $\{x_k\}$ is a Cauchy sequence and so converges to a point $x^* \in X^*$ (as X^* is closed). Furthermore, without loss of generality, we may assume that $c_{k+1} \le c_k \le \frac{1}{2}$ (see the first inequality in (4.6)). Write $d_k := x_{k+1} - x_k$ for simplicity. Then (4.6) implies that $\|d_{k+j}\| \le c_k^j \|d_k\|$ for each $j \ge 1$. Therefore, $\frac{\sum_{j=1}^{\infty}\|d_{k+j}\|}{\|d_k\|} \le \frac{c_k}{1-c_k} \to 0$, and so $\lim_{k\to\infty} \frac{\|\sum_{j=0}^{\infty} d_{k+j}\|}{\|d_k\|} = 1$, because

$$1 - \frac{\sum_{j=1}^{\infty}\|d_{k+j}\|}{\|d_k\|} \le \frac{\|\sum_{j=0}^{\infty} d_{k+j}\|}{\|d_k\|} \le 1 + \frac{\sum_{j=1}^{\infty}\|d_{k+j}\|}{\|d_k\|}.$$

Consequently, we conclude that

$$\limsup_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^q} = \limsup_{k\to\infty} \frac{\|\sum_{j=1}^{\infty} d_{k+j}\|}{\|\sum_{j=0}^{\infty} d_{k+j}\|^q} = \limsup_{k\to\infty} \frac{\|d_{k+1}\|}{\|d_k\|^q} \le \left(\frac{1}{1-\tau}\right)^q c^2,$$

which means that $\{x_k\}$ converges to x^* at a rate of q. □

References

1. R. B. Bapat and T. E. S. Raghavan. Nonnegative Matrices and Applications. In *Encyclopedia of Mathematics and its Applications*, volume 15. Cambridge University Press, Cambridge, 1997.
2. A. Berman and R. J. Plemmons. Nonnegative Matrices in the Mathematical Sciences. In *Classics in Applied Mathematics*, volume 9. SIAM, Philadelphia, 1994.
3. D. P. Bertsekas. *Nonlinear Programming.* Athena scientific, Cambridge, UK, 1999.
4. P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Trans. Automat. Sci. Engrg.*, 3(4):360–371, 2006.
5. P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on information processing in sensor networks*, pages 46–54, 2004.
6. J. F. Bonnans and A. Ioffe. Second-order sufficiency and quadratic growth for nonisolated minima. *Math. Oper. Res.*, 20:801–817, 1995.
7. J. V. Burke. Descent methods for composite nondifferentiable optimization problems. *Math. Program.*, 33(3):260–279, 1985.
8. J. V. Burke. An exact penalization viewpoint of constrained optimization. *SIAM J. Control Optim.*, 29:968–998, 1991.
9. J. V. Burke and M. C. Ferris. Weak sharp minima in mathematical programming. *SIAM J. Control Optim.*, 31(5):1340–1359, 1993.
10. J. V. Burke and M. C. Ferris. A Gauss-Newton method for convex composite optimization. *Math. Program.*, 71:179–194, 1995.
11. J. V. Burke and R. A. Poliquin. Optimality conditions for non-finite valued convex composite functions. *Math. Program.*, 57:103–120, 1992.
12. M. T. Chu. Inverse eigenvalue problems. *SIAM Rev.*, 40:1–39, 1998.

13. M. T. Chu and G. H. Golub. Structured inverse eigenvalue problems. *Acta Numer.*, 11:1–71, 2002.
14. D. Drusvyatskiy and A. S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Math. Oper. Res.*, 43:919–948, 2018.
15. O. P. Ferreira, M. L. N. Gonçalves, and P. R. Oliveira. Convergence of the Gauss-Newton method for convex composite optimization under a majorant condition. *SIAM J. Optim.*, 23:1757–1783, 2013.
16. R. Fletcher. *Practical Methods of Optimization.* Wiley, New York, 1987.
17. P. A. Fuhrmann and U. Helmke. Nonnegative matrices and graph theory. In *The Mathematics of Networks of Linear Systems*, pages 411–466. Springer International Publishing, 2015.
18. G. H. Golub and C. F. Van Loan. *Matrix Computations.* John Hopkins University Press, Baltimore, 2013.
19. Y. H. Hu, C. Li, and X. Q. Yang. On convergence rates of linearized proximal algorithms for convex composite optimization with applications. *SIAM J. Optim.*, 26(2):1207–1235, 2016.
20. Y. H. Hu, X. Q. Yang, and C.-K. Sim. Inexact subgradient methods for quasi-convex optimization problems. *European J. Oper. Res.*, 240:315–327, 2015.
21. X. Ji and H. Y. Zha. Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. In *IEEE INFOCOM*, volume 4, pages 2652–2661, 2004.
22. A. S. Lewis and S. J. Wright. A proximal method for composite minimization. *Math. Program.*, 158:501–546, 2016.
23. C. Li and K. F. Ng. Majorizing functions and convergence of the Gauss-Newton method for convex composite optimization. *SIAM J. Optim.*, 18:613–642, 2007.
24. C. Li and X. H. Wang. On convergence of the Gauss-Newton method for convex composite optimization. *Math. Program.*, 91(2):349–356, 2002.
25. G. Li. Global error bounds for piecewise convex polynomials. *Math. Program.*, 137(1-2):37–64, 2013.
26. Z. Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Z. Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Proc. Mag.*, 27(3):20–34, 2010.
27. J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer Science+Business Media, New York, 2006.
28. J.-S. Pang. Error bounds in mathematical programming. *Math. Program.*, 79:299–332, 1997.
29. H. Perfect. Methods of constructing certain stochastic matrices. *Duke Math. J.*, 20:395–404, 1953.
30. H. Perfect. Methods of constructing certain stochastic matrices. ii. *Duke Math. J.*, 22:305–311, 1955.
31. L. Qi and J. Sun. A nonsmooth version of Newton's method. *Math. Program.*, 58(1-3):353–367, 1993.
32. R. T. Rockafellar. First- and second-order epi-differentiability in nonlinear programming. *Trans. Amer. Math. Soc.*, 307(1):75–108, 1988.
33. J. B. Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard. In *Proceedings of the 17th Allerton Conference on Communications, Control, and Computing*, pages 480–489, 1979.
34. M. Studniarski and D. E. Ward. Weak sharp minima: Characterizations and sufficient conditions. *SIAM J. Control Optim.*, 38(1):219–236, 1999.
35. H. R. Suleĭmanova. Stochastic matrices with real characteristic numbers. *Doklady Akad. Nauk SSSR (NS)*, 66:343–345, 1949.
36. W. Y. Sun and Y. X. Yuan. *Optimization Theory and Methods.* Springer Science+Business Media, New York, 2006.
37. R. S. Womersley. Local properties of algorithms for minimizing nonsmooth composite functions. *Math. Program.*, 32(1):69–89, 1985.
38. Z. Zhao, Z. J. Bai, and X. Q. Jin. A Riemannian inexact Newton-CG method for constructing a nonnegative matrix with prescribed realizable spectrum. *Numer. Math.*, 140:827–855, 2018.
39. X. Y. Zheng and K. F. Ng. Strong KKT conditions and weak sharp solutions in convex-composite optimization. *Math. Program.*, 126:259–279, 2011.
40. X. Y. Zheng and X. Q. Yang. Weak sharp minima for semi-infinite optimization problems with applications. *SIAM J. Optim.*, 18:573–588, 2007.