J. Appl. Numer. Optim. 2 (2020), No. 3, pp. 387-399
Available online at http://jano.biemdas.com
https://doi.org/10.23952/jano.2.2020.3.08

A MULTI-VIEW ON THE CQ ALGORITHM FOR SPLIT FEASIBILITY PROBLEMS: FROM OPTIMIZATION LENS

TINGXIA LU, LULU ZHAO, HONGJIN HE*

Department of Mathematics, Hangzhou Dianzi University, Hangzhou, 310018, China

Abstract. The split feasibility problem (SFP) provides a powerful unified model to characterize many real-world inverse problems arising from image reconstruction and intensity-modulated radiation therapy. As we know, the original CQ algorithm, which is essentially a gradient-projection method, is one of the most popular methods in the SFP literature. In this paper, we revisit the CQ algorithm and give a multi-view on such an algorithm from another four different optimization methods. Specifically, we show that the CQ algorithm can be viewed as applications of partially linearized alternating minimization algorithms, fixed-point methods, DC (Difference-of-Convex) algorithms, and majorization-minimization (MM) algorithms to some structured optimization reformulations of the SFP. Our analysis could provide some new insights into the treatment of SFPs and related topics.

Keywords. Alternating minimization method; DC programming; DC algorithm; Fixed-point method; Majorization-minimization algorithm; Split feasibility problem.

1. Introduction

Let $C \subset \mathbb{R}^n$ and $Q \subset \mathbb{R}^m$ be two nonempty closed convex sets, and let A be an $m \times n$ real matrix. The classical split feasibility problem (SFP) originally introduced in [7] is to find a point $x \in \mathbb{R}^n$ such that

$$x \in C \quad \text{and} \quad Ax \in Q. \tag{1.1}$$

In the literature, problem 1.1 was further studied in Hilbert spaces [19, 28] and in the setting of uniformly convex and uniformly smooth Banach spaces [23]. Since many real world problems, e.g., image reconstruction, computer tomography, Dantzig selectors, and therapy treatment planing (see [8, 9, 10, 11, 13, 20]) fall into the Euclidean space, we are concerned with the real SFPs in this paper.

In the earlier work for the computation of SFPs, some algorithms often assumed that A is invertiable due to the requirement of inverse on A (see [7]). However, such an invertibility of A is relatively strong for many real world problems. Moreover, computing the inverse of A is usually expense, even if not impossible, and the algorithms equipped with the inverse of A are comparatively not ideal for large-scale SFPs. To circumvent the computation (or requirement)

*Corresponding author.
E-mail addresses: lutx@163.com (T. Lu), zhllu@163.com (L. Zhao), hehjmath@hdu.edu.cn (H. He).
Received October 13, 2020; Accepted December 4, 2020.

of the inverse of A, Byrne [5, 6] judiciously introduced the so-called CQ algorithm for (1.1), which reads as

$$x_{k+1} = P_C\left(x_k - \gamma A^\top (I - P_Q)Ax_k\right), \quad \gamma \in (0, 2/L), \tag{1.2}$$

where L is the largest eigenvalue of matrix $A^\top A$ and both $P_C(\cdot)$ and $P_Q(\cdot)$ denote the projections onto the convex sets C and Q, respectively. It is clear that CQ algorithm (1.2) is enough implementable as long as the projections onto C and Q have closed-form solutions or are easy to calculate. Certainly, for the case where the projections are onto relatively complex sets C and Q, Yang [30] wisely introduced a relaxed CQ algorithm by using two iteration-varying half-spaces C_k and Q_k instead of C and Q respectively such that the CQ algorithm can maximally alleviate the computations of projections.

Interestingly, the CQ algorithm has been shown as a specific application of the classical gradient-projection method [28] and a special case of the forward-backward method [12], which can be derived based on the following constrained optimization problem:

$$\min_{x \in C} f(x) := \frac{1}{2}\|Ax - P_Q(Ax)\|^2, \tag{1.3}$$

where the objective function $f(x)$ is continuously differentiable and the gradient of $f(x)$ is given by

$$\nabla f(x) = A^\top (I - P_Q)Ax. \tag{1.4}$$

Note that the I here represents an identity operator throughout this paper. From (1.4), we can easily prove that ∇f is Lipschitz continuous with the Lipschitz constant L mentioned in (1.2). Consequently, we can easily obtain the CQ algorithm by the employment of the gradient projection method to (1.3). In the literature, it has been well documented that such an optimization reformulation (1.3) for (1.1) helpfully leads to a large number of efficient algorithms for SFPs; see, e.g., [13, 15, 20, 24, 29, 31, 32], to name just a few. Here, we should emphasize that the gradient projection method indeed exploits the explicit formula of the gradient of $f(x)$ in (1.3). In this case, the CQ algorithm can also be interpreted as a structure-exploiting application of the gradient-projection method to (1.1). Actually, the word ‘split’ appeared in the name of SFP implies that it has promising split nature, which is potentially helpful for algorithmic design. Hence, how to exploit the split nature of SFPs is the key to design efficient algorithms.

In this paper, we shall show that the SFP can be equivalently rewritten as some structured optimization problems. Accordingly, we exploit the structure of these optimization reformulations to show that the classical CQ algorithm can be further interpreted as special cases of some state-of-the-art first-order methods, including the partially linearized alternating minimization algorithm, the fixed point method, the DC (Difference-of-Convex) algorithm, and the Majorization-Minimization (MM) algorithm. Consequently, we believe that some more efficient variants of the CQ algorithm could be developed from these optimization insights.

The rest of this paper is organized as follows. In Section 2, we summarize some notations, definitions, properties on operators that will be used in the subsequent analysis. In Section 3, we first introduce an auxiliary variable to reformulate (1.1) as a coupled optimization problem over two separable convex sets. Then, we employ the classical alternating minimization algorithm to the resulting coupled optimization model. By using the linearization technique, we gainfully derive out the CQ algorithm. In Section 4, we further reformulate the model proposed in Section 3 as an unconstrained coupled optimization problem by the employment of indicator functions.

Consequently, we shall show that the CQ algorithm is an application of the fixed point method. In Section 5, we will reformulate the SFP as a DC minimization problem based on (1.3). Then, we can easily show that the CQ algorithm indeed is a specific DC algorithm for SFPs. In Section 6, by using the Lipschitz continuity of the gradient of $f(x)$ (see (1.4)), we construct a surrogate function to locally approximate the objective function $f(x)$. Hence, a direct application of the well-known MM algorithm immediately leads to the CQ algorithm. Finally, we conclude the relationships between the above methods and the CQ algorithm in Section 7.

2. Preliminaries

Let $\mathbb{R}^n$ be an n-dimensional Euclidean space. For any two vectors $x,y \in \mathbb{R}^n$, we shall use $\langle x,y\rangle = x^\top y$ to represent the standard inner product, where $^\top$ means the transpose of a vector or a matrix. The standard ℓ_2-norm is $\|\cdot\| = \sqrt{\langle \cdot,\cdot\rangle}$. Let K be a nonempty closed convex subset of $\mathbb{R}^n$. The projection from $\mathbb{R}^n$ onto K, denoted by $P_K(\cdot)$, is defined by

$$P_K(\cdot) = \arg\min\{\|\cdot - y\| \mid y \in K\}.$$

With the above definition of the projection, we have the following property on the projection operator [2].

Proposition 2.1. *Given $x \in \mathbb{R}^n$ and $z \in K$. Then $z = P_K(x)$ if and only if there holds the relation:*

$$\langle x-z, y-z\rangle \le 0, \quad \forall y \in K.$$

Definition 2.1. Let C be a nonempty closed convex set. We call $\tau_C(x)$ the indicator function of C, which is given by

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

Definition 2.2. Let $C \subseteq \mathbb{R}^n$ be a nonempty convex set. For each $x \in C$, we say that $\mathscr{N}_C(x)$ given by

$$\mathscr{N}_C(x) := \left\{\zeta \in \mathbb{R}^n \mid \langle \zeta, x' - x\rangle \le 0 \ \text{ for all } \ x' \in C\right\}$$

is the normal cone of C at x.

Letting $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function such that the gradient ∇f is L-Lipschitz continuous, i.e.,

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x-y\|, \quad \forall x,y \in \mathbb{R}^n,$$

we denote $f \in C_L^{1,1}(\mathbb{R}^n)$ throughout this paper. According to [2, Lemma 4.22], we have the following descent lemma.

Lemma 2.1. *Let $f \in C_L^{1,1}(\mathbb{R}^n)$. Then, for any $x,y \in \mathbb{R}^n$,*

$$f(y) \le f(x) + \langle \nabla f(x), y-x\rangle + \frac{L}{2}\|x-y\|^2. \tag{2.1}$$

Definition 2.3. Let $T : \mathbb{R}^n \to \mathbb{R}^n$ be a nonlinear operator. We say that

(i) T is nonexpansive if

$$\|Tx - Ty\| \le \|x-y\|, \quad \forall x,y \in \mathbb{R}^n.$$

(ii) T is firmly nonexpansive if

$$\langle x-y, Tx-Ty\rangle \geq \|Tx-Ty\|^2, \quad \forall x,y\in\mathbb{R}^n.$$

(iii) T is averaged if

$$T=(1-\alpha)I+\alpha N,$$

where $\alpha\in(0,1)$ and N is a nonexpansive operator.

It is well known that projection operator is nonexpansive, and also is firmly nonexpansive. In particular, the projection is a $\frac{1}{2}$-averaged operator.

Definition 2.4. Let $f:\mathbb{R}^n\to(-\infty,\infty]$ be a proper function and let $x\in\text{dom}(f)$. A vector $\xi\in\mathbb{R}^n$ is called a subgradient of f at x if

$$f(y)\geq f(x)+\langle\xi, y-x\rangle, \quad \forall y\in\mathbb{R}^n.$$

Definition 2.5. The set of all subgradients of f at x is called the subdifferential of f at x and is denoted by $\partial f(x)$:

$$\partial f(x)\equiv\{\xi\in\mathbb{R}^n \mid f(y)\geq f(x)+\langle\xi,y-x\rangle,\ \forall y\in\mathbb{R}^n\}.$$

Theorem 2.1. *Let f be a twice continuously differentiable function over an open convex set $C\subseteq\mathbb{R}^n$. Then f is convex if and only if the Hessian matrix of f is positive semidefinite, i.e., $\nabla^2 f(x)\succeq 0$, for any $x\in C$.*

Theorem 2.2 ([4, 28])**.** *Let K be a closed convex set in $\mathbb{R}^n$ and let $T:K\to K$ be an averaged operator with $Fix(T)\neq\emptyset$, where $Fix(T)$ means the set of fixed points associated to T. Then, for any $x\in K$, the sequence $\{T^n x\}_{n=0}^{\infty}$ is weakly convergent to a fixed point of T.*

3. Alternating Minimization Algorithm

In this section, we first reformulate SFP (1.1) as a coupled optimization problem over two separable convex sets. Then, we show that the employment of a partially linearized alternating minimization algorithm to the resulting coupled optimization model can yield the CQ algorithm.

Taking a close look at (1.1), we can find that the underlying two convex sets C and Q are coupled directly due to the linear operator A. Indeed, we can easily decouple both of them by introducing an auxiliary variable $y\in\mathbb{R}^m$ such that $y=Ax$. As an immediate consequence, finding a solution to SFP (1.1) amounts to solving

$$Ax=y,\ \ x\in C,\ \ y\in Q, \tag{3.1}$$

which is indeed a system of linear equations with constraints. It is clear that such a system can be recast as the following constrained least square problem:

$$\min_{x\in C, y\in Q}\ \Phi(x,y):=\frac{1}{2}\|Ax-y\|^2. \tag{3.2}$$

It is trivially to prove that (3.2) is a convex optimization problem, which has global minimizers. However, due to the coupled objective function and the appearances of C and Q, such a problem cannot be efficiently solved by the direct utilization of traditional gradient methods. Fortunately, we can minimize x and y one-by-one in an alternating order, i.e., for a given x_k, we respectively update the y_{k+1} and x_{k+1} via

$$\begin{cases} y_{k+1} = \arg\min_{y\in Q} \Phi(x_k,y) := \frac{1}{2}\|Ax_k - y\|^2, & \text{(3.3a)} \\ x_{k+1} = \arg\min_{x\in C} \Phi(x,y_{k+1}) := \frac{1}{2}\|Ax - y_{k+1}\|^2. & \text{(3.3b)} \end{cases}$$

In the literature, the above method is called *alternating minimization algorithm* (AMA), see [1]. Since A often is not an identity matrix, the x-subproblem (3.3b) has no explicit formula for x_{k+1} when $C \neq \mathbb{R}^n$. In this situation, it is computationally expensive to require an accurate solution. By the smoothness of $\Phi(x,y_{k+1})$ with respect to x, we accordingly approximate the objective function by linearizing it at x_k, i.e.,

$$\begin{aligned} \Phi(x,y_{k+1}) &\approx \Phi(x_k,y_{k+1}) + \langle \nabla_x \Phi(x_k,y_{k+1}),\, x - x_k\rangle + \frac{\mu}{2}\|x - x_k\|^2 \\ &= \frac{1}{2}\|Ax_k - y_{k+1}\|^2 + \langle A^\top(Ax_k - y_{k+1}),\, x - x_k\rangle + \frac{\mu}{2}\|x - x_k\|^2, \end{aligned} \tag{3.4}$$

where $\mu > 0$ is an approximation parameter. Accordingly, we use (3.4) instead of $\Phi(x,y_{k+1})$ in the x-subproblem (3.3b), thereby reducing the computational cost of updating x_{k+1} in the sense that it has a closed-form solution as follows:

$$x_{k+1} = P_C\left(x_k - \mu^{-1}A^\top (Ax_k - y_{k+1})\right). \tag{3.5}$$

Notice that the y-subproblem (3.3a) has a closed-form solution with

$$y_{k+1} = P_Q(Ax_k). \tag{3.6}$$

Consequently, plugging (3.6) into (3.5) immediately yields a simplified scheme, i.e.,

$$x_{k+1} = P_C\left(x_k - \mu^{-1}A^\top (I - P_Q)Ax_k\right), \tag{3.7}$$

which is the CQ algorithm. Since only one subproblem of the original AMA is linearized in algorithmic derivation, we call the proposed algorithm *partially linearized alternating minimization algorithm* (PLAMA). Theoretically, we can prove that (3.7) is globally convergent for $\mu \geq \|A^\top A\|$ (see, e.g., [3, 25]), where $\|A^\top A\|$ corresponds to the Lipschitz constant of $\nabla f(x)$ (or $\nabla_x \Phi(x,y)$).

Remark 3.1. Actually, we can call y in (3.3) an intermediate variable in the derivation of (3.7), since such a variable can be removed easily by (3.6). Similarly, plugging (3.6) into (3.5) directly, we can also obtain a simplified scheme as follows:

$$x_{k+1} = \arg\min_{x\in C} \frac{1}{2}\|Ax - P_Q(Ax_k)\|^2. \tag{3.8}$$

It is easy to observe that (3.7) is an one projection approximation to (3.8), and the approximation relies on the choice of μ. However, if we could obtain x_{k+1} via (3.8) directly, we do not assume that $\mu \geq \|A^\top A\|$. To some extent, iterative scheme (3.8) can be regarded as a relaxed method without any parameters. In future, we will pay our attention to investigating its numerical performance.

Remark 3.2. Here, we shall emphasize that why we update x and y in the order $y_{k+1} \to x_{k+1}$ as used in (3.3). Theoretically, both x and y share the equal roles in algorithmic design. However, when we update x and y in the order $x_{k+1} \to y_{k+1}$ as follows

$$\begin{cases} x_{k+1} = \arg\min_{x\in C} \Phi(x, y_k) := \frac{1}{2}\|Ax - y_k\|^2, & (3.9a) \\ y_{k+1} = \arg\min_{y\in Q} \Phi(x_{k+1}, y) := \frac{1}{2}\|Ax_{k+1} - y\|^2, & (3.9b) \end{cases}$$

it is obvious that such an iterative scheme requires an initial input y_0. Due to the general matrix A in x-subproblem (3.9a), we cannot remove x_{k+1} in (3.9b) or y_k as the way used in (3.5), which means that there is no intermediate variable for simplifying the algorithm. We do not know, for (3.3) and (3.9), which one is better for SFPs in practice. However, the numerical results in [13] demonstrate that (3.3) performs better than (3.9) empirically when linearizing their x-subproblems simultaneously. One possible reason is that we need only one initial point x_0 when linearizing (3.3b), but two initial points x_0 and y_0 when linearizing (3.9a), where the initial guess y_0 perhaps results in a nonideal approximation for (3.9).

4. Fixed Point Method

It was shown that the gradient projection method for SFPs is indeed a fixed point method [28] based on (1.3). In this section, we employ the indicator functions to reformulate (3.2) as an unconstrained coupled optimization problem. Then, we show that an application of the fixed point method to the resulting coupled minimization problem also yields the CQ algorithm.

First, by Definition 2.1, we reformulate (3.2) as the following unconstrained optimization problem:

$$\min_{x,y} \ \tau_C(x) + \tau_Q(y) + \Phi(x, y), \tag{4.1}$$

where $\Phi(x, y)$ is given in (3.2). Recalling the definition of subdifferential (see Definition 2.5), the first-order optimality condition of (4.1) is

$$\begin{cases} 0 \in \partial\tau_Q(y) + \nabla_y\Phi(x, y) = \mathscr{N}_Q(y) + (y - Ax), & (4.2a) \\ 0 \in \partial\tau_C(x) + \nabla_x\Phi(x, y) = \mathscr{N}_C(x) + A^\top(Ax - y), & (4.2b) \end{cases}$$

where both $\mathscr{N}_C$ and $\mathscr{N}_Q$ are the normal cones to the convex sets C and Q, respectively. Clearly, it follows from (4.2) that, for $\gamma > 0$,

$$\begin{cases} Ax & \in & \mathscr{N}_Q(y) + y, \\ x - \gamma A^\top(Ax - y) & \in & \mathscr{N}_C(x) + x, \end{cases}$$

which in turn leads to

$$\begin{cases} y = (I + \mathscr{N}_Q)^{-1} Ax, & (4.3a) \\ x = (I + \gamma\mathscr{N}_C)^{-1} \left(x - \gamma A^\top(Ax - y)\right). & (4.3b) \end{cases}$$

Since $(I + \mathscr{N}_Q)^{-1} = P_Q$ and $(I + \gamma\mathscr{N}_C)^{-1} = P_C$, it immediately follows from (4.3) that

$$\begin{cases} y = P_Q(Ax), & (4.4a) \\ x = P_C\left(x - \gamma A^\top(Ax - y)\right). & (4.4b) \end{cases}$$

Clearly, plugging (4.4a) into (4.4b) immediately leads to

$$x = P_C\left(x - \gamma A^\top (I - P_Q)Ax\right), \tag{4.5}$$

which is a fixed point equation. Since (4.1) is a convex minimization problem, we know that solving (4.1) amounts to finding a fixed point of (4.5). Below, we also present an elementary proof to show that solutions of the fixed point equation (4.5) are exactly the solutions of (1.1).

Proposition 4.1. *A point $x^* \in \mathbb{R}^n$ solves the SFP (1.1) if and only if x^* is a fixed point of (4.5).*

Proof. Obviously, if x^* is a solution of SFP (1.1), then it also solves fixed point equation (4.5). Conversely, we assume that x^* is a fixed point of (4.5), then it is obvious that $x^* \in C$. According to Proposition 2.1, we have

$$\langle x^* - \gamma A^\top (I - P_Q)Ax^* - x^*,\, z - x^* \rangle \leq 0, \quad \forall z \in C,$$

which is equivalent to

$$\langle A^\top (I - P_Q)Ax^*,\, z - x^* \rangle \geq 0, \quad \forall z \in C,$$

or

$$\langle Ax^* - P_Q(Ax^*),\, Ax^* - Az \rangle \leq 0, \quad \forall z \in C. \tag{4.6}$$

On the other hand, once more application of Proposition 2.1 on the convex set Q yields

$$\langle Ax^* - P_Q(Ax^*), v - P_Q(Ax^*) \rangle \leq 0, \quad \forall v \in Q. \tag{4.7}$$

Adding (4.6) and (4.7) immediately arrives at

$$\langle Ax^* - P_Q(Ax^*), Ax^* - P_Q(Ax^*) - (Az - v) \rangle \leq 0, \quad \forall z \in C, \quad \forall v \in Q. \tag{4.8}$$

By the arbitrariness of z and v, we take z being a solution of (1.1) and $v = P_Q(Az) \in Q$ in (4.8). Consequently, we obtain $Ax^* = P_Q(Ax^*) \in Q$, which, together with $x^* \in C$, means that x^* is a solution of the SFP (1.1). □

As we know, a benchmark method for fixed point equation (4.5) is fixed point method. Hence, for a given x_k, the fixed point method for (4.5) reads as

$$x_{k+1} = P_C\left(x_k - \gamma A^\top (I - P_Q)Ax_k\right),$$

which is exactly the CQ algorithm, and can be further rewritten as

$$x_{k+1} = Tx_k \quad \text{with} \quad T := P_C\left(I - \gamma A^\top (I - P_Q)A\right). \tag{4.9}$$

When taking $\gamma \in (0, 2/L)$, it has been documented in [6] that operator T is an averaged operator. Hence, assuming that the solution set of SFP (1.1) is nonempty, we have $Fix(T) \neq \emptyset$ by Proposition 4.1. So, by Theorem 2.1, the sequence $\{x_k\}$ generated by (4.9) converges weakly to an element of $Fix(T)$ or a solution of SFP (1.1).

5. DC ALGORITHM

In this section, we first recall some preliminaries on DC (Difference-of-Convex) programming and DC algorithm. Then, we reformulate the SFP problem (1.1) as a DC programming and show that an application of DC algorithm can derive out the CQ algorithm.

Generally speaking, a so-called standard DC programming takes the form

$$\min_{x\in\mathbb{R}^n} \theta(x) := g(x) - h(x) \tag{5.1}$$

where $g, h : \mathbb{R}^n \to \mathbb{R}$ are convex functions. Since $\theta(x)$ is a difference of two convex functions, we usually call $\theta(x)$ is a DC function, and both $g(x)$ and $h(x)$ are called DC components of $\theta(x)$. Here, we refer the reader to [17] for a recent survey on DC programming.

Proposition 5.1 ([26]). *A point $x^* \in \mathbb{R}^n$ is a locally optimal solution of problem (5.1) if and only if*

$$\partial h(x^*) \subset \partial g(x^*).$$

As we know, DC programming is a nonconvex optimization problem which is intractable in many cases. However, such a problem has special structure that could be exploited in algorithmic design. In the literature, one of the most popular method for DC programming is the so-named DC algorithm (DCA), which uses an affine function of the second DC component, i.e., $h(x)$, at x_k instead of the original one. Specifically, the DCA for (5.1) reads as

$$x_{k+1} \in \arg\min_{x\in\mathbb{R}^n} \{g(x) - h(x_k) - \langle x - x_k,\, \xi_k\rangle\} \quad \text{with} \quad \xi_k \in \partial h(x_k). \tag{5.2}$$

Clearly, subproblem (5.2) in DCA is a convex minimization problem, which is much easier than the original problem (5.1). Below, we summarize some convergence results of the DCA.

Theorem 5.1 ([14, 18, 27]). *DCA is a descent method without line-search which enjoys the following primal properties:*

- (i) *The sequence $\{g(x_k) - h(x_k)\}$ is decreasing.*
- (ii) *$g(x_{k+1}) - h(x_{k+1}) = g(x_k) - h(x_k)$ if and only if $\xi_k \in \partial g(x_k) \cap \partial h(x_k)$, $\xi_k \in \partial g(x_{k+1}) \cap \partial h(x_{k+1})$ and $\rho\|x_{k+1} - x_k\| = 0$, where ρ is a constant associated with g and h. Moreover, if g or h are strictly convex on C, then $x_{k+1} = x_k$. In such a case, DCA terminates at the k-th iteration (finite convergence).*
- (iii) *If the optimal value of problem (5.1) is finite and the infinite sequence $\{x_k\}$ are bounded, then every limit point $\bar{x}$ of the sequence is a critical point of $(g - h)$.*
- (iv) *DCA has a linear convergence for general DC programs.*

Now, we first reformulate (1.3) as an unconstrained optimization problem as follows

$$\min_x \Psi(x) := \tau_C(x) + \frac{1}{2}\|Ax - P_Q(Ax)\|^2,$$

where $\tau_C(x)$ is an indicator function of the set C. Consequently, by introducing a positive constant α, we set $g(x)$ and $h(x)$ in (5.1) as

$$\begin{cases} g(x) := \tau_C(x) + \dfrac{\alpha}{2}\|x\|^2, & (5.3a) \\[2mm] h(x) := \dfrac{\alpha}{2}\|x\|^2 - \dfrac{1}{2}\|Ax - P_Q(Ax)\|^2. & (5.3b) \end{cases}$$

Clearly, $g(x)$ is a convex due to the convexity of $\tau_C(x)$ and $\frac{\alpha}{2}\|x\|^2$. Moreover, $h(x)$ is also a convex function for all $\alpha \geq L$, where L is the Lipschitz constant of $\nabla f(x)$ in (1.4). Therefore, we obtain a DC decomposition of $\Psi(x)$ and a DC programming formulation for SFP (1.1). Since $h(x)$ is continuously differentiable, we know that

$$\partial h(x) = \nabla h(x) = \alpha x - A^\top (Ax - P_Q(Ax)).$$

Applying the DCA on (5.1) with the setting (5.3) immediately leads to

$$\begin{cases} \xi_k = \alpha x_k - A^\top (Ax_k - P_Q(Ax_k)), \\ x_{k+1} = \arg\min_x \ \tau_C(x) + \dfrac{\alpha}{2}\|x\|^2 - (h(x_k) + \langle x - x_k,\ \xi_k\rangle), \end{cases}$$

which can be simplified as

$$x_{k+1} = P_C\left(\alpha^{-1}\xi_k\right) = P_C\left(x_k - \alpha^{-1}A^\top (Ax_k - P_Q(Ax_k))\right). \tag{5.5}$$

Hence, we also derive out the CQ algorithm through the DCA algorithm. As stated in Proposition 5.1, the DCA is based on local optimality condition in DC programming. Thus, we have the following result.

Proposition 5.2. *A point $x^* \in \mathbb{R}^n$ solves the SFP* (1.1) *if and only if x^* is a locally optimal solution of* (5.1) *with the setting* (5.3)*.*

Proof. Obviously, if x^* solves the SFP (1.1), then $x^* \in C$ and $Ax^* \in Q$. Since

$$\begin{cases} \partial h(x^*) = \alpha x^* - A^\top (Ax^* - P_Q(Ax^*)) = \alpha x^* \\ \partial g(x^*) = \partial \tau_C(x^*) + \alpha x^* = \mathcal{N}_C(x^*) + \alpha x^*, \end{cases}$$

we immediately obtain that $\partial h(x^*) \subset \partial g(x^*)$. According to Proposition 5.1, it is clear that $x^* \in C$ satisfies the local optimality condition, which means that x^* is a locally optimal solution of (5.1) with (5.3).

Conversely, we assume that x^* is a locally optimal solution of (5.1) with (5.3). It is evident that $x^* \in C$ satisfies $\partial h(x^*) \subset \partial g(x^*)$. Due to the above DC decomposition of $\Psi(x)$, i.e., $\Psi(x) = g(x) - h(x)$, we have

$$\partial g(x^*) = \partial h(x^*) + \partial \Psi(x^*),$$

which, together with $\partial h(x^*) \subset \partial g(x^*)$, implies that

$$\partial h(x^*) \subset \partial h(x^*) + \partial \tau_C(x^*) + \nabla f(x^*).$$

Since $\partial h(x^*)$ is nonempty and bounded, we have

$$0 \in \partial \tau_C(x^*) + \nabla f(x^*) = \mathcal{N}_C(x^*) + \nabla f(x^*).$$

According to Definition 2.2, we can further obtain

$$0 = \nabla f(x^*) = A^\top (I - P_Q)Ax^*.$$

Consequently, we have $Ax^* \in Q$, which means that x^* is a solution of SFP (1.1). ☐

According to the properties of the convergence of DCA, i.e., Theorem 5.1, the iterative sequence $\{x_k\}$ generated by (5.5) converges to a solution of SFP (1.1) for all $\alpha \geq L$. Of course, we could follow the DC reformulation idea to reformulate (1.3) as other DC programming problems, thereby deriving out more efficient algorithms for SFPs. Moreover, we can employ or modify efficient DC solvers developed in recent years (see, e.g., [17]) to find solutions of SFPs.

6. MM Algorithm

In this section, we first give a briefly description on the idea of the classical Majorization-Minimization (MM) algorithm for minimization problems. Then, we show that the CQ algorithm is indeed a special case of the MM algorithm when we choose a proper surrogate function for the objective.

Consider the following minimization problem

$$\min_{x} \{f(x) \mid x \in C\}, \tag{6.1}$$

where C is a nonempty closed convex subset of $\mathbb{R}^n$ and $f(x) : C \to \mathbb{R}$ is a continuous function. In many real-world problems arising from signal/image processing and statistical learning, directly minimizing the objective function $f(x)$ over C is usually not an easy task due to the nonconvexity or nonsmoothness of $f(x)$. To reduce the difficulty caused by intractable objectives, one natural idea is to find a series of surrogate functions so that all of them are tractable or have more beneficial properties for minimization than the original objective function. Consequently, we can easily obtain minimizers of these surrogate functions such that the sequence of minimizers approaches to a local or global solution of (6.1). All methods followed this idea are usually called MM algorithms, where MM stands for *majorization-minimization* and *minorization-maximization* for minimization and maximization problems, respectively. Here, we refer the reader to [16] for a more recent monograph on MM algorithms.

For minimization problems, the MM algorithm can be divided into two steps. The first step is called the majorization step, where we find a surrogate function being an upper bound of the objective function. The second step corresponds to minimizing the surrogate function, which is called the minimization step. Specifically, when applying the MM algorithm to (6.1), we first construct a surrogate function $\varphi(x|x_k)$ anchored at the current iteration x_k satisfying the MM principle (see [16, Ch. 1]), i.e.,

$$\begin{cases} f(x) \le \varphi(x|x_k), \quad \forall x \in C, & (6.2a) \\ f(x_k) = \varphi(x_k|x_k). & (6.2b) \end{cases}$$

Then, we minimize the surrogate function $\varphi(x|x_k)$ at the k-iteration to obtain x_{k+1}, i.e.,

$$x_{k+1} \in \arg\min_{x\in C} \varphi(x|x_k). \tag{6.3}$$

Promisingly, $\{f(x_k)\}$ is nonincreasing since

$$f(x_{k+1}) \le \varphi(x_{k+1}|x_k) \le \varphi(x_k|x_k) = f(x_k),$$

which is a descent property making the MM algorithm very stable. When $\varphi(x|x_k)$ is minimized, the objection function $f(x)$ is driven downhill as needed. Moreover, we have the following convergence results.

Theorem 6.1 ([21, 22])**.** *If both $f(x)$ and $\varphi(x|x_k)$ are continuously differentiable with respect to x, and $\varphi(x|x_k)$ is continuous in x and x_k. Then, the following statements hold:*

(i) *Any limit point of the sequence $\{x_k\}$ generated by* (6.3) *is a stationary point of* (6.1)*;*
(ii) *$f(x_k)$ is monotonically decreasing to $f(x^*)$, where x^* is a stationary point of* (6.1).

With the above preparations, we are now applying the MM algorithm to SFP (1.1). First, we take $f(x)$ in (6.1) as the objective function given in (1.3). Then, it follows from Lemma 2.1 and the Lipschitz continuity of $\nabla f(x)$ that

$$f(x) \leq f(x_k) + \langle \nabla f(x_k),\, x - x_k \rangle + \frac{\beta}{2}\|x - x_k\|^2, \quad \forall \beta \geq L, \tag{6.4}$$

where L is the Lipschitz constant of $\nabla f(x)$. Let

$$\varphi(x|x_k) := f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{\beta}{2}\|x - x_k\|^2. \tag{6.5}$$

It is trivial to verify that $\varphi(x|x_k)$ satisfies the MM principle (6.2). Therefore, we can take $\varphi(x|x_k)$ defined by (6.5) as the surrogate function of $f(x)$ anchored at the iterate x_k. Invoking the minimization step of MM algorithm to (6.5) immediately yields

$$\begin{aligned} x_{k+1} &= \arg\min_{x \in C}\ \varphi(x|x_k) \\ &= \arg\min_{x \in C}\ f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{\beta}{2}\|x - x_k\|^2 \\ &= P_C\left(x_k - \beta^{-1} A^\top (I - P_Q) A x_k\right), \end{aligned} \tag{6.6}$$

which is the CQ algorithm. According to Theorem 6.1, we have the following theorem, whose proof is elementary and is skipped here for simplicity.

Theorem 6.2. *The sequence $\{x_k\}$ generated by* (6.6) *converges to a solution of SFP* (1.1).

As we know, there many surrogate functions for (1.3). Hence, the MM algorithmic framework provides a wide way to design efficient algorithms for SFPs.

7. Conclusions

The split feasibility problem (SFP) is a deeply studied topic from theories to solution methods and applications. As one of the most popular solvers in the SFPs literature, the CQ algorithm has been shown a special case of the gradient projection method and the forward-backward method. In this paper, we gave a multi-view on the CQ algorithm from optimization lens. Specifically, we first reformulated SFP (1.1) as some structured optimization models. Then, we employed some state-of-the-art first order methods to the resulting reformulations, thereby showing that the CQ algorithm can also be a special case of the partially linearized alternating minimization algorithm, fixed point method, DC algorithm, and MM algorithm. The relationship between these reformulations and algorithms for SFP (1.1) can be shown graphically in Figure 1. Actually, many existing efficient algorithms for SFPs were developed based on optimization model (1.3). In this paper, we provided some more optimization reformulations for SFPs. Hence, designing more efficient numerical algorithms based on these optimization models is one of our future concerns.

Acknowledgments

H. He was supported in part by National Natural Science Foundation of China (No. 11771113) and Natural Science Foundation of Zhejiang Province (No. LY20A010018).

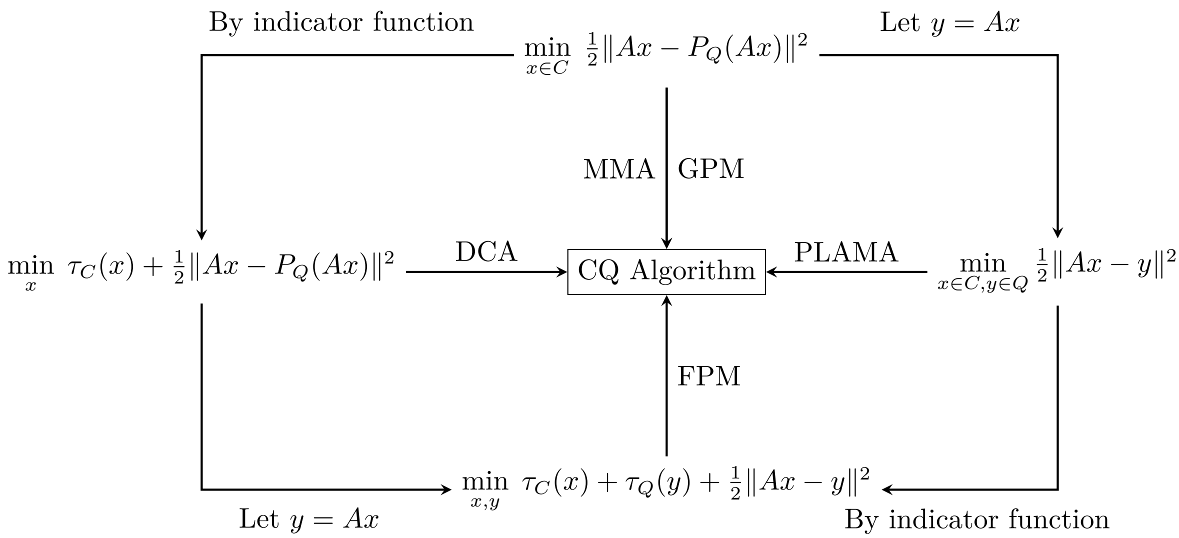

FIGURE 1. The relationship between these reformulations and algorithms for SFP (1.1), where MMA, GPM, DCA, PLAMA, FPM correspond to Majorization-Minimization Algorithm, Gradient Projection Method, DC Algorithm, Partially Linearized Alternating Minimization Algorithm, Fixed Point Method, respectively.

REFERENCES

[1] A. Beck, First-Order Methods in Optimization, SIAM, Philadelphia, 2018.

[2] A. Beck, Introduction to Nonlinear Optimization Theory, Algorithms, and Applications with MATLAB, SIAM, Philadelphia, 2014.

[3] J. Bolte, S. Sabach, M. Teboulle, Proximal alternating linearized minimization for nonconvex and nonsmooth problems, Math. Program. 146 (2014), 459-494.

[4] C.L. Byrne, A. Moudafi, Extensions of the CQ algorithm for the split feasibility and split equality problems, Documents De Travail, 18 (2013), 1485-1496

[5] C.L. Byrne, Iterative oblique projection onto convex sets and the split feasibility problem, Inverse probl. 18 (2002), 441-453.

[6] C.L. Byrne, A unified treatment of some iterative algorithms in signal processing and image reconstruction, Inverse probl. 20 (2004), 103-120.

[7] Y. Censor, T. Elfving, A multiprojection algorithm using Bregman projections in a product space, Numer. Algo. 8 (2004), 221-239.

[8] Y. Censor, T. Bortfeld, B. Martin, A. Trofimov, A unified approach for inversion problems in intensity-modulated radiation therapy, Phys. Med. Biolo. 51 (2006), 2353-2365.

[9] Y. Censor, T. Elfving, N. Kopf, T. Bortfeld, The multiple-sets split feasibility problem and its applications for inverse problems, Inverse Probl. 21 (2005), 2071-2084.

[10] Y. Censor, A. Motova, A. Segal, Perturbed projections and subgradient projections for the multiple-sets split feasibility problem, J. Math. Anal. Appl. 327 (2007), 1244-1256.

[11] P. L. Combettes, Hilbertian convex feasibility problem: Convergence of projection methods, Appl. Math. Optim. 35 (1997), 311-330.

[12] P.L. Combettes, V.R. Wajs, Signal recovery by proximal forward-backward splitting, Multiscale Model. Simul. 4 (2005), 1168-1200.

[13] H.J. He, H.-K. Xu, Splitting methods for split feasibility problems with application to Dantzig selectors, Inverse Probl. 33 (2017), 055003.

[14] L.T. Hoai An, P.D. Tao, On solving linear complementarity problems by DC programming and DCA, Comput. Optim. Appl. 50 (2011), 507-524.

[15] H.J. He, C. Ling, H.-K. Xu, An implementable splitting algorithm for the ℓ_1-norm regularized split feasibility problem, J. Sci. Comput. 67 (2016), 281-298.

[16] K. Lange, MM Optimization Algorithms, SIAM, Philadelphia, 2016.

[17] H.A. Le Thi, T. Pham Dinh, DC programming and DCA: Thirty years of developments, Math. Program. 169 (2018), 5-68.

[18] Y.-S. Niu, P.D. Tao, DC programming approaches for BMI and QMI feasibility problems. In: Advanced Computational Methods for Knowledge Engineering, pp. 37-63, Springer, 2014.

[19] X. Qin, A. Petrusel, J.-C. Yao, CQ iterative algorithms for fixed points of nonexpansive mappings and split feasibility problems in Hilbert spaces, J. Nonlinear Convex Anal. 19 (2018), 157-165.

[20] B. Qu, N.-H. Xiu, A note on the CQ algorithm for the split feasibility problem, Inverse Probl. 21 (2005), 1655-1665.

[21] M. Razaviyayn, M.Y Hong, Z.-Q. Luo, A unified convergence analysis of block successive minimization methods for nonsmooth optimization, SIAM J. Optim. 23 (2013), 1126-1153.

[22] Y. Sun, P. Babu, D.P. Palomar, Majorization-minimization algorithms in signal processing, communications, and machine learning, IEEE Trans. Signal Process. 65 (2016), 794-816.

[23] F. Schöpfer, T. Schuster, A.K. Louis, An iterative regularization method for the solution of the split feasibility problem in Banach spaces, Inverse Probl. 24 (2008), 055008.

[24] Y. Shehu, O.S. Iyiola, C.D. Enyi, An iterative algorithm for solving split feasibility problems and fixed point problems in Banach spaces, Numerical Algo. 72 (2016), 835-864.

[25] R. Shefi, M. Teboulle, On the rate of convergence of the proximal alternating linearized minimization algorithm for convex problems, EURO J. Comput. Optim. 4 (2016), 27-46.

[26] P.D. Tao, The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems, Ann. Oper. Res. 133 (2005), 23-46.

[27] P.D. Tao, L.T. Hoai An, DC programming. Theory, algorithms, applications: the state of the art, First International Workshop on Global Constrained Optimization and Constraint Satisfaction, Nice, vol. 24, 2002.

[28] H.-K. Xu, Iterative methods for the split feasibility problem in infinite-dimensional Hilbert spaces, Inverse Probl. 26 (2010), 105018.

[29] Y. Yao, J.G. Wu, Y.C. Liou, Regularized methods for the split feasibility problem, Abst. Appl. Anal. 2012 (2012), Article ID 140679.

[30] Q.Z. Yang, The relaxed CQ algorithm solving the split feasibility problem, Inverse Probl. 20 (2004), 1261-1266.

[31] H.Y. Zhang, Y.J. Wang, A new CQ method for solving split feasibility problem, Frontiers Math. China 5 (2010), 37-46.

[32] W.X. Zhang, D.R. Han, Z.B. Li, A self-adaptive projection method for solving the multiple-sets split feasibility problem, Inverse Probl. 25 (2009), 115001.