

Open Journal of Mathematical Optimization

Aviv Gibali, Shoham Sabach & Sergey Voldman

Non-Convex Split Feasibility Problems: Models, Algorithms and Theory


Volume 1 (2020), article no. 1 (15 pages)

[<http://ojmo.centre-mersenne.org/item/OJMO_2020__1__A1_0>](http://ojmo.centre-mersenne.org/item/OJMO_2020__1__A1_0)

Article submitted on February 6, 2020, revised on June 25, 2020,
accepted on October 7, 2020.

© The journal and the authors, 2020.

Some rights reserved.

 This article is licensed under the
CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL LICENSE.
<http://creativecommons.org/licenses/by/4.0/>



*Open Journal of Mathematical Optimization is member of the
Centre Mersenne for Open Scientific Publishing
www.centre-mersenne.org*

Non-Convex Split Feasibility Problems: Models, Algorithms and Theory

Aviv Gibali

Department of Mathematics, ORT Braude College, Karmiel 2161002, Israel
The Center for Mathematics and Scientific Computation, University of Haifa, Mt. Carmel, Haifa 3498838, Israel
avivg@braude.ac.il

Shoham Sabach

Faculty of Industrial Engineering and Management, The Technion, Haifa, 3200003, Israel
ssabach@ie.technion.ac.il

Sergey Voldman

Faculty of Industrial Engineering and Management, The Technion, Haifa, 3200003, Israel
sergeyv@campus.technion.ac.il

Abstract

In this paper, we propose a catalog of iterative methods for solving the Split Feasibility Problem in the non-convex setting. We study four different optimization formulations of the problem, where each model has advantages in different settings of the problem. For each model, we study relevant iterative algorithms, some of which are well-known in this area and some are new. All the studied methods, including the well-known CQ Algorithm, are proven to have global convergence guarantees in the non-convex setting under mild conditions on the problem's data.

Digital Object Identifier 10.5802/ojmo.1

Keywords Split feasibility problems, non-convex minimization, convergence analysis, constrained minimization, CQ algorithm.

1 Introduction

In 1994 Censor and Elfving [17] introduced the (two-sets) *Split Convex Feasibility* (SCF) Problem, which is formulated as follows. Let $C \subseteq \mathbb{R}^n$ and $Q \subseteq \mathbb{R}^m$ be two non-empty, closed and convex sets, and let $A : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be a linear mapping. The SCF Problem then seeks to:

$$\text{find a point } x^* \in C \text{ such that } y^* = Ax^* \in Q. \quad (1)$$

Here, in this paper, we will focus on the *Split Feasibility* (SF) Problem, where the involved sets are not necessarily convex. For simplicity we start our discussion with only one set per each space and in the last section discuss how the general multiple-sets case can be treated in a similar way.

The SF Problem and its convex variant were employed in solving several real-world challenging problems, such as in Intensity-Modulated Radiation Therapy (IMRT) treatment planning [16], navigation on the Pareto frontier in Multi-Criteria Optimization [20] and others.

The main goal of this paper is to study four main formulations of the SF Problem as an optimization problem, where for each model we present relevant existing algorithms and develop new algorithms that can tackle it. For any mentioned algorithm, we discuss the current state-of-the-art convergence guarantee, which in some cases already exists in the literature and in some cases provided for the first time here. We summarize our study in the following table, which will be made fully clear when reading the following section.

In order to derive the four mentioned formulation, two functions will play a pivotal role: the *indicator function* and the *distance function*. Mathematically speaking, given a non-empty and closed set D in the d -dimensional Euclidean space \mathbb{R}^d , the *indicator function* is defined by

$$\delta_D(u) = \begin{cases} 0, & \text{if } u \in D, \\ +\infty, & \text{otherwise.} \end{cases}$$



© Aviv Gibali & Shoham Sabach & Sergey Voldman;

licensed under Creative Commons License Attribution 4.0 International

2 Non-Convex Split Feasibility Problems: Models, Algorithms and Theory

Setting	Algorithm	Model	Global Convergence	Requirements on the linear transformation A
C and Q non-convex	Proximal ADMM (Alg. 1)	(SF1)	Unknown	Unknown
	PG (Alg. 2)	(SF1 - Penalized)	[3, Th. 5.3, p. 120]	None
	CQ Algorithm (Alg. 4)	(SF1 - Penalized)	Section 3.1	None
C non-convex and Q convex	PG (Alg. 5)	(SF3)	[3, Th. 5.3, p. 120]	None
	AM (Alg. 3)	(SF1 - Penalized)	Section 3.2	None
C convex and Q non-convex	Proximal ADMM (Alg. 6)	(SF4)	Section 3.3	$AA^T \succ 0$
	Linear. Prox. ADMM (Alg. 7)	(SF4)	Section 3.3	$AA^T \succ 0, \kappa(A^T A) < 2$

The *distance function* of the set D , denoted by $d_D : \mathbb{R}^d \rightarrow \mathbb{R}_+$, is given by

$$d_D(u) = \min \{\|v - u\| : v \in D\}.$$

These two functions are obviously lower semi-continuous, non-smooth and, in general, non-convex, unless the set D is convex. Another essential property of these functions, especially when reformulating the SF Problem as an optimization problem, is that both functions are non-negative and achieve their minimal value of zero only on the set D .

Before presenting the models, it is important to note that for the sake of preferable mathematical properties (at least in the convex setting, as we discuss below), the function $d_D^2(\cdot)$ is usually considered, rather than the distance function itself. For example, when the set D is convex, then the function $d_D^2(\cdot)$ is smooth. This property is very important in deriving solution methods and establishing their theoretical guarantees.

Next we present the four optimization models for the SF Problem, which use the above two functions.

$$(SF1) \quad \min_{x \in \mathbb{R}^n} \{\delta_C(x) + \delta_Q(Ax)\}$$

$$(SF2) \quad \min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2}d_C^2(x) + \frac{1}{2}d_Q^2(Ax) \right\}$$

$$(SF3) \quad \min_{x \in \mathbb{R}^n} \left\{ \delta_C(x) + \frac{1}{2}d_Q^2(Ax) \right\}$$

$$(SF4) \quad \min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2}d_C^2(x) + \delta_Q(Ax) \right\}$$

One of the main advantages of formulating the SF Problem as an optimization problem is that there is no need to know/assume in advance consistency of the problem, that is, that the original feasibility problem has a solution. In case that the SF Problem has a solution then the above four models can achieve the minimal value zero. In the convex setting, solutions of the SF Problem coincide with the optimal solution of any of the optimization models. On the other hand, in the non-convex setting, since optimal solutions are very difficult to find, we only know that if the value of the achieved solution is zero then this is a solution of the SF Problem.

Throughout the paper we will only focus on algorithms which have the following two important properties. First, computational efficiency, by that we mean algorithms which use only orthogonal projections (defined below) onto the individual sets C and Q , the mapping A and its transpose. For example, algorithms which use the inverse mapping of A fail in this aspect, due to their computational deficiencies in solving large-scale instances of the SF Problem (see, for instance, [17] for such an algorithm). Second, algorithms with a provable convergence theory, and since we focus on the non-convex setting, we mean by that algorithms with convergence guarantees to critical points of the used optimization model (exact details and definitions will be given in the sequel).

In the area of SCF Problems there is one central and well-known algorithm, which was designed to tackle the optimization model (SF3), and is called the CQ Algorithm [14, 15]. This algorithm, which exploits the structure of the model (SF3) in the convex setting, is nothing else but the classical Projected Gradient (PG) Method. Since, here we focus on the non-convex setting, we point interested readers to [7] and the recent book [6], where theoretical results on the PG method can be found, including accelerations and various modifications.

While the literature on iterative methods for solving the SCF Problem is focused on the optimization model (SF3) and to the best of our understanding most algorithms are modifications and/or generalizations of the CQ Algorithm for other settings of the problem or other related problems with similar affinities, there is one recent paper [18] (as far as we know) that study the CQ Algorithm in the non-convex setting. We will discuss this work in details, later on, after giving the needed notations and relevant definitions.

Speaking on the different optimization models mentioned above, we have witnessed that most of the works focus on developing algorithms that solve the optimization model (SF3), sometime even without paying attention to this fact. However, another model that got some attention, again only in the convex setting, is (SF2), see for instance [29]. In this case, this model poses a completely smooth optimization problem, that can be solved via classical techniques of optimization for smooth and unconstrained problems, such as the Gradient Descent Method, the Newton Method and others (see for example the books [5, 8]). Therefore, in the non-convex setting, this model is less relevant and we actually do not know of any algorithm that can tackle it. To summarize, the models (SF2) and (SF3) are indeed very natural in the convex setting, due to the nice properties of the squared distance function, something that is much less attractive in the non-convex and preclude using relevant algorithmic ideas. We will see below that, in the non-convex setting, the model (SF1) plays an important role and even allows to easily recover the CQ Algorithm through different algorithmic idea. Model (SF4) also allows for some algorithms developments as will be given below.

We conclude this part with a short description of the rest of the paper. In Section 2 we will discuss each optimization model and present the relevant algorithms (if any) that can tackle it in the non-convex setting. The theoretical guarantees for each algorithm will be presented in Section 3, where in some cases the convergence result is already known and in other cases the result is new and proven here for the first time. It should be noted that in this paper we focus on theoretical guarantees in the sense of global convergence of the whole generated sequence, in comparison to sub-sequences convergence, to critical points of the tackled model. Another type of theoretical guarantees that can be achieved for such algorithms in the non-convex setting is of a local nature and consist of results of linear rate of convergence. This line of research, which usually requires more information of the involved sets, is not studied here but interested readers can find several recent results in [28] and the references therein. We end the paper, with Section 4, where we discuss, in short, the extension of the SF Problem to the multi-set setting.

2 Algorithms for Solving the SF Problem

The indicator and distance functions, as we discussed above, are our central tools in reformulating the SF Problem as a minimization problem. Another object that is very related to these two functions is the *orthogonal projection* operator. Given a non-empty and closed subset D of the Euclidean space \mathbb{R}^d , the *orthogonal projection* onto the set D is the operator $P_D : \mathbb{R}^d \rightarrow D$ which is defined by

$$P_D(u) = \operatorname{argmin} \{ \|v - u\| : v \in D \}.$$

Since the set D is not necessarily convex, the resulting projector P_D may be a set-valued operator. In other words, for a given point $u \in \mathbb{R}^d$, the set $P_D(u)$ consists of all vectors in D that are closest to the point u . This connect us to the distance function, defined above, since we obviously have, for all $u \in \mathbb{R}^d$, that $d_D(u) = \|v - u\|$, for any $v \in P_D(u)$.

Recall that in the convex setting, the squared distance function enjoys some nice properties, two of them are given next, and others can be found, for example, in [4].

► **Proposition 1.** *Let D be a non-empty, closed and convex subset of \mathbb{R}^d . Then, the following hold:*

- (i) *The function $d_D^2(\cdot)$ is continuously differentiable.*
- (ii) *The gradient of $(1/2)d_D^2(\cdot)$ is Lipschitz continuous with constant 1 (nonexpansive) and is given by $I_d - P_D$, where I_d is the identity mapping of \mathbb{R}^d .*

In this section we study the four optimization models mentioned above and relevant iterative methods for solving the SF Problem (1) defined in the Euclidean spaces \mathbb{R}^n and \mathbb{R}^m . Since we are interested in the non-convex setting, we will distinguish in our study between the following three scenarios:

- (i) Both sets C and Q are non-convex.
- (ii) The set C is convex and Q is non-convex.
- (iii) The set C is non-convex and Q is convex.

Although the last two scenarios include one convex set, the associated optimization models still possess a non-convex minimization. However, due to the partial convexity, we will be able to exploit it in order to design iterative algorithms for solving a relevant model. In order to gain from the convexity of one of the sets, we will consider models that use the squared distance function with respect to this set. See more details below.

2.1 The Model (SF1)

We begin with the model (SF1), which tackles both sets C and Q via their associated indicator functions, that is,

$$\min_x \{ \delta_C(x) + \delta_Q(Ax) \}.$$

This model fits to the completely non-convex scenario, and all the algorithms to be presented soon do not require any convexity in their development and analysis (exact details will be given in Section 3). This is a non-smooth and non-convex minimization, which at a first glance seems to be the most complex model between all the four mentioned models, even when both sets are convex. However, the situation is exactly the opposite, and this formulation opens the gate for some interesting algorithms. Moreover, this formulation paves the way to easily obtain the CQ Algorithm in the non-convex setting through a different approach as we will develop below.

The optimization model (SF1) belongs to the well-studied class of additive composite models, which is a very popular and active area of research in optimization, both in the convex and non-convex settings. Putting aside, for the moment, the issue of convexity, the main difficulty comes from the composition of the linear mapping A in the term $\delta_Q(Ax)$. The most popular approach to handle this situation, is methods which decompose the linear mapping A out of the indicator function $\delta_Q(\cdot)$, and therefore usually called decomposition or splitting methods. More precisely, we reformulate Model (SF1) by introducing a new variable u , which leads to the following linearly constrained non-smooth minimization

$$\min_{x \in \mathbb{R}^n, u \in \mathbb{R}^m} \{ \delta_C(x) + \delta_Q(u) : Ax = u \}. \quad (2)$$

This splitting technique usually leads to Lagrangian-based methods. Therefore, the augmented Lagrangian of Problem (2) is given by

$$L_\rho(x, u, y) = \delta_C(x) + \delta_Q(u) + \langle y, Ax - u \rangle + \frac{\rho}{2} \|Ax - u\|^2,$$

where $y \in \mathbb{R}^m$ is the multiplier which associated with the linear constraint $Ax = u$, and $\rho > 0$ is a penalization parameter.

The well-known Alternating Directions Method of Multipliers (ADMM) [23, 19] is a very popular algorithm for solving large-scale problems with linear constrains (see, for example, a recent review paper on Lagrangian-based methods [33] including the references therein). Here, we will consider the Proximal ADMM (which includes ADMM as a particular case and can be useful in some cases) that applied to the augmented Lagrangian of Problem (2) and recorded now.

Algorithm 1 - Proximal ADMM for (SF1)

Initialization: Choose $x^0 \in \mathbb{R}^n$ and $u^0, y^0 \in \mathbb{R}^m$, and fix $\tau_1, \tau_2 \geq 0$.

General Step: For $k = 0, 1, 2, \dots$ compute

$$\begin{aligned} u^{k+1} &\in \operatorname{argmin}_u \left\{ \delta_Q(u) + \langle y^k, Ax^k - u \rangle + \frac{\rho}{2} \|Ax^k - u\|^2 + \frac{\tau_1}{2} \|u - u^k\|^2 \right\}, \\ x^{k+1} &\in \operatorname{argmin}_x \left\{ \delta_C(x) + \langle y^k, Ax - u^{k+1} \rangle + \frac{\rho}{2} \|Ax - u^{k+1}\|^2 + \frac{\tau_2}{2} \|x - x^k\|^2 \right\}, \\ y^{k+1} &= y^k + \rho (Ax^{k+1} - u^{k+1}). \end{aligned}$$

However, this algorithm is hard to analyze (to the best of our knowledge there is no convergence theory for this algorithm in the non-convex setting). Therefore, we will not pursue this direction for this model, but discuss Lagrangian-based methods for the model (SF4) below.

Now, we would like to study another way to tackle the optimization model (SF1) and thus we return to Problem (2). We consider now a different approach to tackle the linear constraint. A very popular idea in the literature to handle linear constraints is to penalize them. In this case, the following optimization problem is used

$$\min_{x \in \mathbb{R}^n, u \in \mathbb{R}^m} \left\{ \delta_C(x) + \delta_Q(u) + \frac{\lambda}{2} \|Ax - u\|^2 \right\}, \quad (3)$$

where $\lambda > 0$ is a penalizing parameter. This and the previous problems are obviously not equivalent, but it is easy to verify that an obtained solution x^* is a solution of the SF Problem at hand if and only if the pair (x^*, Ax^*) is an optimal solution of Problem (3), with the optimal value of zero.

The optimization problem (3) consists of two blocks, which are coupled through the smooth and convex function $(\lambda/2) \|Ax - u\|^2$. In addition, the problem includes two separable constraints via the corresponding indicator functions. Such problems belong to the class of block additive composite models, which are well-studied in the convex literature and even in the non-convex there are several works on this class of problems, see, for instance, [11]. We present here two approaches – one that exploits the block structure and the other which does not.

First approach tackles the two blocks as a single block via the Projected Gradient Method, which enjoys the separability in the constraint sets C and Q in order to produce a parallel method. Taking a step-size $\tau > 0$, after simple manipulations we obtain the following algorithm.

Algorithm 2 - Projected Gradient for (SF1-Penalized)

Initialization: Choose $x^0 \in \mathbb{R}^n$ and $u^0 \in \mathbb{R}^m$, and fix $\tau > 0$.

General Step: For $k = 0, 1, 2, \dots$ compute

$$\begin{aligned} u^{k+1} &\in P_Q \left(u^k - \frac{\lambda}{\tau} (u^k - Ax^k) \right), \\ x^{k+1} &\in P_C \left(x^k - \frac{\lambda}{\tau} A^T (Ax^k - u^k) \right). \end{aligned}$$

In a similar spirit to the application of the PG Method we can actually apply the inertial variant, which yields better performances in practice and tend to converges to critical points with lower function value (more on that can be found in [30]). To keep the paper short, we do not go into details here, but the convergence guarantees we present below also apply to the inertial variant.

Our second approach to tackle the penalized problem (3) is designed to exploit the two block structure of the problem. To this end, we will use the Alternating Minimization (AM) technique, where the separability in the constraints is crucial. In this case, we obviously have a cyclic algorithm rather than a parallel as in the PG Method.

Algorithm 3 - Alternating Minimization for (SF1-Penalized)

Initialization: Choose $x^0 \in \mathbb{R}^n$.

General Step: For $k = 0, 1, 2, \dots$ compute

$$u^{k+1} \in \operatorname{argmin}_u \left\{ \delta_Q(u) + \frac{\lambda}{2} \|Ax^k - u\|^2 \right\}, \quad (4)$$

$$x^{k+1} \in \operatorname{argmin}_x \left\{ \delta_C(x) + \frac{\lambda}{2} \|Ax - u^{k+1}\|^2 \right\}. \quad (5)$$

A careful inspection of the two updating rules reveals a similar computational situation to the Proximal ADMM mentioned above. More precisely, it is easy to see that Problem (4) simply reduces to the computation of the orthogonal projection of Ax^k onto the set Q . Therefore, as long as the operator P_Q is explicitly given or can be efficiently computed, the update of the block u is easy. However, this is not the case with the updating rule of the block x . Indeed, Problem (5) is not easily solved unless the matrix $A^T A$ is invertible, which is a very demanding assumption that is not assumed in this paper.

We will overcome this obstacle by linearizing the quadratic coupling term. This idea can be seen also as incorporating the AM Algorithm above with a PG step applied on the minimization (5). More precisely, the suggested algorithm is recorded now, where we denote by $\lambda_{max}(A^T A)$ the largest eigenvalue of the symmetric square matrix $A^T A$.

Algorithm 4 - Semi Alternating Projected Gradient for (SF1-Penalized)

Initialization: Choose $x^0 \in \mathbb{R}^n$ and fix $\tau > \lambda_{max}(A^T A)$.

General Step: For $k = 0, 1, 2, \dots$ compute

$$u^{k+1} \in P_Q (Ax^k), \quad (6)$$

$$x^{k+1} \in \operatorname{argmin}_x \left\{ \delta_C(x) + \lambda \langle A^T (Ax^k - u^{k+1}), x - x^k \rangle + \frac{\tau}{2} \|x - x^k\|^2 \right\}. \quad (7)$$

Now, the updating rule of the x block is explicitly solvable and by substituting the solution to the u block from (6), we obtain

$$x^{k+1} \in P_C \left(x^k - \frac{\lambda}{\tau} A^T (Ax^k - u^{k+1}) \right) = P_C \left(x^k - \frac{\lambda}{\tau} A^T (Ax^k - P_Q (Ax^k)) \right).$$

The obtained algorithm (after substituting the auxiliary variable u) is exactly the CQ Algorithm, which is recovered via application of the Semi Alternating Projected Gradient on the penalized formulation of Problem (SF1). As far as we know this is a new approach to get the CQ Algorithm (rather than applying the PG Method on the Model (SF3), as we discuss in the next subsection), which allow us to derive convergence guarantees under much milder assumptions on the sets C and Q in comparison to [26].

2.2 The Model (SF3)

Now we move to the next optimization model

$$\min_x \left\{ \delta_C(x) + \frac{1}{2} d_Q^2(Ax) \right\} = \min_{x \in C} d_Q^2(Ax),$$

where the last equality follows from the property of the indicator function. Meaning, we have a constrained minimization.

As we already discussed before, this model of the SF Problem makes sense mainly if the set Q is convex, since then the function $(1/2) d_Q^2(\cdot)$ is continuously differentiable, and its gradient is given by $I_m - P_Q$. More importantly, this gradient is Lipschitz continuous with constant 1. Algorithmically speaking, these properties opens the gate for applying the Projected Gradient (PG) Method, even when the set C is non-convex. This leads again to the well-known CQ Algorithm [14, 15] via different optimization model and different algorithmic framework (but limited only to the case where Q is convex).

Algorithm 5 - Projected Gradient for (SF3)

Initialization: Choose $x^0 \in \mathbb{R}^n$ and fix $\tau > 0$.

General Step: For $k = 0, 1, 2, \dots$ compute

$$\begin{aligned} x^{k+1} &\in \operatorname{argmin}_x \left\{ \delta_C(x) + \langle A^T (Ax^k - P_Q(Ax^k)), x - x^k \rangle + \frac{\tau}{2} \|x - x^k\|^2 \right\} \\ &= P_C \left(x^k - \frac{1}{\tau} A^T (Ax^k - P_Q(Ax^k)) \right). \end{aligned}$$

Very recently, the authors of [26] managed to derive the CQ Algorithm in the fully non-convex setting via this optimization model using techniques of Difference-of-Convex (DC) Programming. However, in their paper the authors assume two main assumptions on the involved sets C and Q , which are not needed in our case. First assumption revolves around the regularity of the distance function $d_Q(y^*)$, at limit points of the form $y^* = Ax^*$, which is needed to obtain the sub-differential of the distance function in the non-convex setting. However, this is a demanding assumption that is difficult to check in practice. In addition, the regularity assumption should be satisfied at limit points of the generated sequence, which complicates more the ability to verify it (unless all point in $A(C)$ are regular). The second assumption is a prox-regularity of the set Q at limit points of the generated sequence. Verifying this assumption in practice could be also very difficult, and limit significantly the applicability of the results. Under this assumption we assure that the function $d_Q^2(\cdot)$ is continuously differentiable, which is crucial property in their global convergence analysis.

2.3 The Model (SF4)

In this part we study the Model (SF4), which like the Model (SF3) mixes the indicator and distance functions in the following way

$$\min_x \left\{ \frac{1}{2} d_C^2(x) + \delta_Q(Ax) \right\},$$

and following the same reasons as in the previous part, we assume here that the set C is convex.

Due to the composition of the nonsmooth function $\delta_Q(\cdot)$ with the linear mapping A , we can not apply the PG Method on this problem, and would need to consider again splitting techniques. By applying a simple splitting technique, as above, with the introduction of a new variable u , we derive an equivalent formulation of the problem

$$\min_{x \in \mathbb{R}^n, u \in \mathbb{R}^m} \left\{ \frac{1}{2} d_C^2(x) + \delta_Q(u) : Ax = u \right\}.$$

Using the augmented Lagrangian associated with this problem, we can apply the Proximal ADMM to obtain the following algorithm.

Algorithm 6 - Proximal ADMM for (SF4)

Initialization: Choose $x^0 \in \mathbb{R}^n$ and $u^0, y^0 \in \mathbb{R}^m$, and fix $\tau \geq 0$.

General Step: For $k = 0, 1, 2, \dots$ compute

$$u^{k+1} \in \operatorname{argmin}_u \left\{ \delta_Q(u) + \langle y^k, Ax^k - u \rangle + \frac{\rho}{2} \|Ax^k - u\|^2 \right\}, \quad (8)$$

$$x^{k+1} = \operatorname{argmin}_x \left\{ \frac{1}{2} d_C^2(x) + \langle y^k, Ax - u^{k+1} \rangle + \frac{\rho}{2} \|Ax - u^{k+1}\|^2 + \frac{\tau}{2} \|x - x^k\|^2 \right\}, \quad (9)$$

$$y^{k+1} = y^k + \rho (Ax^{k+1} - u^{k+1}). \quad (10)$$

First, it is easy to notice that if the parameter τ is chosen to be zero then we recover the classical ADMM. More importantly, is regarding the updating step of the auxiliary variable u . In this case, the augmented term $(\rho/2) \|Ax^{k+1} - u\|^2$ serves as a natural proximal term and therefore no need for an additional proximal term. In addition, the sub-problem given in step (8) are explicitly solvable and given via the projection onto the set Q . Indeed, simple manipulations reveal that

$$u^{k+1} = P_Q \left(\frac{1}{\rho} (y^k + \rho Ax^k + \tau_1 u^k) \right).$$

On the other hand, the updating step with respect to the variable x is more involved and requires computing the inverse of the matrix $\rho A^T A + \tau I_n$ (where I_n is the identity matrix), and as mentioned above, could be computationally expensive in large-scale instances.

A classical approach to overcome this obstacle is by linearizing, at iteration k , the augmented term $(\rho/2) \|Ax - u^k\|^2$ around the current iteration x^k (for more details on this approach in the convex setting we refer to [24, 35] and in the non-convex setting to [12]). This leads us to the Linearized Proximal ADMM, which is recorded now.

Algorithm 7 - Linearized Proximal ADMM for (SF4)

Initialization: Choose $x^0 \in \mathbb{R}^n$ and $u^0, y^0 \in \mathbb{R}^m$, fix $\tau > 0$.

General Step: For $k = 0, 1, 2, \dots$ compute

$$u^{k+1} \in P_Q \left(\frac{1}{\rho} (y^k + \rho Ax^k + \tau_1 u^k) \right). \quad (11)$$

$$x^{k+1} = \operatorname{argmin}_x \left\{ \langle x^k - P_C(x^k) + \rho A^T (Ax^k - u^{k+1}), x \rangle + \langle y^k, Ax - u^{k+1} \rangle + \frac{\tau}{2} \|x - x^k\|^2 \right\}, \quad (12)$$

$$y^{k+1} = y^k + \rho (Ax^{k+1} - u^{k+1}). \quad (13)$$

In this case we obviously see that the updating step of the variable x is also explicitly computable and is given by the following formula

$$x^{k+1} = \left(1 - \frac{1}{\tau} \right) x^k + \frac{1}{\tau} P_C(x^k) - \frac{1}{\tau} A^T (y^k + \rho (Ax^k - u^{k+1})).$$

Therefore, in this case we obtain a completely computable algorithm for solving the SF Problem in the partially non-convex setting. As we already mentioned, these two algorithms come with provable convergence guarantees that will be given below in Section 3.

As the reader probably notice, we did not discuss Model (SF2) so far. The main reason for that is the fact that in the non-convex setting there is no way to exploit properties of this model, which could be handy in designing solution schemes. However, for the SCF Problem, that is, when both sets are convex the situations is better and, as mentioned above, several algorithms can be applied .

3 Convergence Analysis

In this section we first recall the table presented in the Introduction and summarize the convergence information regarding all the 7 algorithms presented in Section 2. The convergence of two algorithm follows directly from known results as stated in table, while one algorithm is lack of convergence theory. Below, we will provide the details about the convergence analysis of all the other four algorithms. Therefore, let us first recall the definition of semi-algebraic sets and functions.

► **Definition 2** (Semi-algebraic sets and functions). (i) *A subset S of \mathbb{R}^d is a real semi-algebraic set if there exists a finite number of real polynomial functions $g_{ij}, h_{ij} : \mathbb{R}^d \rightarrow \mathbb{R}$ such that*

$$S = \bigcup_{j=1}^p \bigcap_{i=1}^q \{u \in \mathbb{R}^d : g_{ij}(u) = 0 \text{ and } h_{ij}(u) < 0\}.$$

(ii) *A function $h : \mathbb{R}^d \rightarrow (-\infty, +\infty]$ is called semi-algebraic if its graph*

$$\{(u, t) \in \mathbb{R}^{d+1} : h(u) = t\}$$

is a semi-algebraic subset of \mathbb{R}^{d+1} .

We are not aiming at reviewing here these notions but few comments could be of help for the reader (more details can be found in [1, 3, 9, 10, 11] and references therein). First, most of the constraint sets that are used in real-world applications are semi-algebraic, for example, balls with respect to the ℓ_1 -norm, ℓ_2 -norm, ℓ_0 -norm, cone of PSD matrices, Stiefel manifold, to mention just a few. In addition, a very important fact to our study, is that the indicator and distance functions are semi-algebraic whenever the associated set is semi-algebraic.

Semi-algebraicity is mentioned here due to its pivotal role in a recent proof methodology of optimization algorithms in the non-convex setting, which is based on the Kurdyka-Łojasiewicz property [27, 25, 9]. The origin of this proof technique was a decade ago in the paper of Attouch and Bolte [1]. Later on, in [11], the technique was unified and extended into a general proof methodology that can be applied to any algorithm in the non-convex setting as long as the optimization problem at hand involves semi-algebraic objects. Very recently, in [13], a concise version of the methodology was presented and the concept of gradient-like descent sequences was coined. Here, we will mention the slight modification of this concept that was studied first in [34].

► **Definition 3** (Gradient-like Descent Sequence). *Let $F : \mathbb{R}^d \times \mathbb{R}^p \rightarrow (-\infty, +\infty]$ be a proper and lower semi-continuous function which is bounded from below, and let $\{(z^k, v^k)\}_{k \in \mathbb{N}}$ be a sequence generated by a certain algorithm for solving the problem*

$$\inf \{F(z, v) : z \in \mathbb{R}^d, v \in \mathbb{R}^p\}.$$

We say that $\{(z^k, v^k)\}_{k \in \mathbb{N}}$ is a gradient-like descent sequence for minimizing F if the following three conditions hold:

(C1) *Sufficient decrease property. There exists a positive scalar ρ_1 such that*

$$\rho_1 \|z^{k+1} - z^k\|^2 \leq F(z^k, v^k) - F(z^{k+1}, v^{k+1}), \quad \forall k \in \mathbb{N}.$$

(C2) *A subgradient lower bound for the iterates gap. There exists a positive scalar ρ_2 such that*

$$\|w^{k+1}\| \leq \rho_2 \|z^{k+1} - z^k\|, \quad w^{k+1} \in \partial F(z^{k+1}, v^{k+1}), \quad \forall k \in \mathbb{N}.$$

(C3) *Let (\bar{z}, \bar{v}) be a limit point of a subsequence $\{(z^k, v^k)\}_{k \in K}$, then $\limsup_{k \in K \subset \mathbb{N}} F(z^k, v^k) \leq F(\bar{z}, \bar{v})$.*

Based on these conditions, global convergence of the generated sequence $\{z^k\}_{k \in \mathbb{N}}$ to a critical point of F can be proved (see [13, Theorem 6.2, p. 2149]). As usual in the non-convex setting, we will use here the concept of

limiting subdifferential for characterizing critical points of F and thanks to Fermat's rule [32, Theorem 10.1, p. 422], the set of critical points of F is given by:

$$\text{crit } F = \{(z, v) \in \mathbb{R}^d \times \mathbb{R}^p : \mathbf{0} \in \partial F(z, v)\}.$$

Now, we can record the convergence result mentioned above.

► **Theorem 4 (Global Convergence).** *Let $\{(z^k, v^k)\}_{k \in \mathbb{N}}$ be a bounded gradient-like descent sequence for minimizing F . If F is semi-algebraic (or, in general, satisfies the KL property), then the sequence $\{z^k\}_{k \in \mathbb{N}}$ converges to some z^* . In addition, for any limit point v^* of $\{v^k\}_{k \in \mathbb{N}}$ we have that $(z^*, v^*) \in \text{crit } F$.*

Therefore, below we will show that the following algorithms: AM for (SF1 - Penalized) and CQ Algorithm for (SF1 - Penalized), generate gradient-like descent sequences and thus, under the semi-algebraicity assumption, converge to a critical point of the associated optimization model. It is very easy to notice that the proof methodology mentioned above can not be applied to the Lagrangian-based algorithms (they are obviously not descent algorithms) and therefore require a different proof technique that will be discussed below in Section 3.3.

One last comment before proving the convergence results of the two algorithms is regarding the boundedness assumption, which is made in Theorem 4. The boundedness assumption on the generated sequence by any of the 7 algorithms holds in several scenarios such as when the sets C and Q are bounded. For a few more scenarios see [2] and also [26] which discuss the same setting of SF Problems. Another possibility is to assume that the following condition holds

$$C^\infty \cap A^{-1}D^\infty = \{\mathbf{0}\},$$

which implies the boundedness of the generated sequence (see [26]).

3.1 Convergence of the CQ Algorithm for (SF1 - Penalized)

Recalling the optimization model (3)

$$\min_{x \in \mathbb{R}^n, u \in \mathbb{R}^m} \left\{ F_1(x, u) := \delta_C(x) + \delta_Q(u) + \frac{\lambda}{2} \|Ax - u\|^2 \right\}.$$

Therefore, from [32, Exercise 8.8 and Proposition 10.5], we have that

$$\text{crit } F_1(x, u) = (\partial \delta_C(x) + \lambda A^T(Ax - u), \partial \delta_Q(u) + \lambda(u - Ax)).$$

We will show now that the Semi Alternating Projected Gradient method (which is nothing but the CQ Algorithm) generates a globally convergent sequence to critical points of F_1 under only the semi-algebraicity assumption of the involved sets C and Q .

► **Proposition 5.** *Let $\{(x^k, u^k)\}_{k \in \mathbb{N}}$ be a bounded sequence generated by the Semi Alternating Projected Gradient method. Suppose that C and Q are semi-algebraic sets. Then, the sequence $\{x^k\}_{k \in \mathbb{N}}$ converges globally to some point x^* such that for any limit point u^* of the sequence $\{u^k\}_{k \in \mathbb{N}}$, the pair (x^*, u^*) is a critical point of F_1 .*

Proof. We first prove that the sequence $\{(x^k, u^k)\}_{k \in \mathbb{N}}$ is a gradient-like descent sequence for minimizing F_1 according to Definition 3. We begin with the first condition (C1). From the Pythagoras identity we have that

$$\begin{aligned} \frac{1}{2} \|Ax^{k+1} - u^{k+1}\|^2 &= \frac{1}{2} \|Ax^k - u^{k+1}\|^2 + \langle Ax^k - u^{k+1}, Ax^{k+1} - Ax^k \rangle + \frac{1}{2} \|Ax^{k+1} - Ax^k\|^2 \\ &\leq \frac{1}{2} \|Ax^k - u^{k+1}\|^2 + \langle Ax^k - u^{k+1}, Ax^{k+1} - Ax^k \rangle + \frac{\lambda_{\max}(A^T A)}{2} \|x^{k+1} - x^k\|^2. \end{aligned}$$

Writing the optimality condition of (7) yields that

$$\delta_C(x^{k+1}) + \lambda \langle Ax^k - u^{k+1}, Ax^{k+1} - Ax^k \rangle + \frac{\tau}{2} \|x^{k+1} - x^k\|^2 \leq \delta_C(x^k).$$

By combining the last two inequalities we obtain that

$$\frac{\tau - \lambda \cdot \lambda_{\max}(A^T A)}{2} \|x^{k+1} - x^k\|^2 + F(x^{k+1}, u^{k+1}) \leq F(x^k, u^{k+1}).$$

From the optimality of u^{k+1} in step (6) we obtain

$$F(x^k, u^{k+1}) \leq F(x^k, u^k),$$

and therefore from the last two inequalities the first assertion holds.

For proving condition (C2) we use the optimality conditions associated with the two steps (6) and (7), meaning

$$\lambda(Ax^k - u^{k+1}) \in \partial\delta_Q(u^{k+1}),$$

and

$$\lambda A^T(u^{k+1} - Ax^k) + \tau(x^k - x^{k+1}) \in \partial\delta_C(x^{k+1}).$$

Therefore, by subtracting λAx^{k+1} and adding $\lambda A^T Ax^{k+1}$, respectively, from both sides of each inclusion, we arrive at

$$\lambda(Ax^k - Ax^{k+1}) \in \partial\delta_Q(u^{k+1}) + \lambda(u^{k+1} - Ax^{k+1}),$$

and

$$\lambda A^T(Ax^{k+1} - Ax^k) + \tau(x^k - x^{k+1}) \in \partial\delta_C(x^{k+1}) + \lambda A^T(Ax^{k+1} - u^{k+1}).$$

This proves that $w^{k+1} := ((\lambda A^T A - \tau I)(x^{k+1} - x^k), \lambda A(x^k - x^{k+1})) \in \partial F(x^{k+1}, u^{k+1})$. Hence, the assertion follows.

Lastly, condition (C3) easily follows since on the generated sequence we have that

$$F_1(x^k, u^k) = (\lambda/2) \|Ax^k - u^k\|^2.$$

Therefore F_1 is continuous on the generated sequence, which proves the required assertion.

Now, from Theorem 4 we get the desired result. ◀

3.2 Convergence of AM for (SF1 - Penalized)

Here we again focus on the model (3) and therefore use again the function $F_1(x, u) = \delta_C(x) + \delta_Q(u) + (\lambda/2) \|Ax - u\|^2$, but with the additional assumption that the set Q is convex.

For simplicity of the proof we will analyze the following shifted equivalent version of the algorithm.

Alternating Minimization for (SF1-Penalized)

Initialization: Choose $x^0 \in \mathbb{R}^n$ and set $u^1 = P_Q(Ax^0)$.

General Step: For $k = 1, 2, \dots$ compute

$$x^{k+1} \in \operatorname{argmin}_x \left\{ \delta_C(x) + \frac{\lambda}{2} \|Ax - u^k\|^2 \right\}, \quad (14)$$

$$u^{k+1} \in \operatorname{argmin}_u \left\{ \delta_Q(u) + \frac{\lambda}{2} \|Ax^{k+1} - u\|^2 \right\}. \quad (15)$$

► **Proposition 6.** *Let $\{(x^k, u^k)\}_{k \in \mathbb{N}}$ be a bounded sequence generated by the AM Algorithm. Suppose that C and Q are semi-algebraic sets and Q is convex. Then, the sequence $\{u^k\}_{k \in \mathbb{N}}$ converges globally to some point u^* such that for any limit point x^* of the sequence $\{x^k\}_{k \in \mathbb{N}}$, the pair (x^*, u^*) is a critical point of F_1 .*

Proof. We first prove that the sequence $\{(x^k, u^k)\}_{k \in \mathbb{N}}$ is a gradient-like descent sequence for minimizing F_1 according to Definition 3. In order to prove condition (C1) we use the fact that the function $u \rightarrow F(x^{k+1}, u)$ is strongly convex with parameter λ . Thus, by the definition of strong convexity and the update step (15), we obtain that

$$F(x^{k+1}, u^{k+1}) + \frac{\lambda}{2} \|u^{k+1} - u^k\|^2 \leq F(x^{k+1}, u^k).$$

From the optimality of x^{k+1} in (14) we obtain

$$F(x^{k+1}, u^k) \leq F(x^k, u^k).$$

Combining both inequalities yields the desired result with $\rho_1 = \lambda/2$.

For proving condition (C2) we use the optimality conditions associated with the two steps (14) and (15), meaning

$$\lambda A^T (u^k - Ax^{k+1}) \in \partial \delta_C (x^{k+1}),$$

and

$$\lambda (Ax^{k+1} - u^{k+1}) \in \partial \delta_Q (u^{k+1}).$$

Therefore, by subtracting u^{k+1} from both sides of the first inclusion, we arrive at

$$\lambda A^T (u^k - u^{k+1}) \in \partial \delta_C (x^{k+1}) + \lambda A^T (Ax^{k+1} - u^{k+1}),$$

and

$$\mathbf{0} \in \partial \delta_Q (u^{k+1}) + \lambda (u^{k+1} - Ax^{k+1}).$$

This proves that $w^{k+1} := (\lambda A^T (u^k - u^{k+1}), \mathbf{0}) \in \partial F (x^{k+1}, u^{k+1})$. Hence, the assertion follows with $\rho_2 = \lambda \|A\|$.

Lastly, condition (C3) easily follows since on the generated sequence we have that

$$F_1(x^k, u^k) = (\lambda/2) \|Ax^k - u^k\|^2.$$

Therefore F_1 is continuous on the generated sequence, which proves the required assertion.

Now, from Theorem 4 we get the desired result. ◀

3.3 Convergence Analysis of the Lagrangian-Based Algorithms

As mentioned above, the proof methodology that was used to derive the convergence results of Sections 3.1 and 3.2 can not be used to analyze the Lagrangian-based algorithms that were presented in Section 2. However, for this class of algorithms we can follow other recent proof methodology of [12], which was the first to propose a proof methodology for Lagrangian-based methods in the non-convex setting. To this end, they defined the concept of Lagrangian sequences, which is in some sense an extension of the notion of gradient-like descent sequence to the setting of primal-dual methods. Here we will show that the Proximal ADMM and the Linearized Proximal ADMM presented above indeed generate Lagrangian sequences and therefore converges to critical points of the optimization model (SF4), as was proved in [12, Theorem 2, p. 1218]. It should be noted that the results of [12] are valid also in the case where the mapping A is not necessarily linear.

Recalling that we focus on the formulation (SF4), where the set C is convex and Q non-convex, via its split form

$$\min_{x \in \mathbb{R}^n} \{F_2(x) := d_C^2(x) + \delta_Q(Ax)\} = \min_{x \in \mathbb{R}^n, u \in \mathbb{R}^m} \{d_C^2(x) + \delta_Q(u) : Ax = u\}, \quad (16)$$

and with the following augmented Lagrangian

$$L_\rho(x, u, y) = d_C^2(x) + \delta_Q(u) + \langle y, Ax - u \rangle + \frac{\rho}{2} \|Ax - u\|^2,$$

where $y \in \mathbb{R}^m$ is the multiplier which associated with the linear constraint $Ax = u$, and $\rho > 0$ is a penalization parameter.

We start by presenting a method which unifies both the Proximal ADMM and the Linearized Proximal ADMM presented above in Section 2.3. We define the weighted norm of a vector v with respect to a matrix $M \succ 0$ by $\|v\|_M^2 := \langle v, Mv \rangle$.

Weighted Proximal ADMM for (SF4)

Initialization: Choose $x^0 \in \mathbb{R}^n$ and $u^0, y^0 \in \mathbb{R}^m$, fix a matrix $N \succ 0$.

General Step: For $k = 0, 1, 2, \dots$ compute

$$u^{k+1} \in \operatorname{argmin}_u \left\{ \delta_Q(u) + \langle y^k, Ax^k - u \rangle + \frac{\rho}{2} \|Ax^k - u\|^2 \right\}, \quad (17)$$

$$x^{k+1} \in \operatorname{argmin}_x \left\{ d_C^2(x) + \langle y^k, Ax - u^{k+1} \rangle + \frac{\rho}{2} \|Ax - u^{k+1}\|^2 + \frac{1}{2} \|x - x^k\|_N^2 \right\}, \quad (18)$$

$$y^{k+1} = y^k + \rho(Ax^{k+1} - u^{k+1}). \quad (19)$$

It is easy to verify that the Proximal ADMM is recovered with $N = \tau I$ for some $\tau > 0$, whereas the Linearized Proximal ADMM Algorithm is recovered when $N = \tau I - \rho A^T A$ for some $\tau > \rho \lambda_{\max}(A^T A)$. Therefore, we will analyze only the Weighted Proximal ADMM and prove converges to critical points of the Problem (SF4).

We therefore prove with the following result, which guarantees the global convergence of both algorithms.

► **Proposition 7.** *Let $\{(x^k, u^k, y^k)\}_{k \in \mathbb{N}}$ be a bounded sequence generated by the Weighted Proximal ADMM Algorithm. Suppose that C and Q are semi-algebraic sets and C is convex. Then, the sequence $\{x^k\}_{k \in \mathbb{N}}$ converges globally to some point x^* which is a critical point of F_2 .*

Proof. We first prove that the sequence $\{(x^k, u^k, y^k)\}_{k \in \mathbb{N}}$ is a Lagrangian sequence as defined in [12]. This requires to prove four conditions. In order to prove the first condition we use the optimality of u^{k+1} and x^{k+1} in (17) and (18), respectively, to obtain that

$$L_\rho(x^k, u^{k+1}, y^k) \leq L_\rho(x^k, u^k, y^k),$$

and

$$L_\rho(x^{k+1}, u^{k+1}, y^k) + \frac{1}{2} \|x^{k+1} - x^k\|_N^2 \leq L_\rho(x^k, u^{k+1}, y^k).$$

Combining both inequalities yields that

$$\frac{\lambda_{\min}(N)}{2} \|x^{k+1} - x^k\|^2 \leq L_\rho(x^k, u^k, y^k) - L_\rho(x^{k+1}, u^{k+1}, y^k), \quad \forall k \in \mathbb{N},$$

which proves the first condition.

For proving the second and third conditions we use the optimality conditions associated with the two steps (17) and (18), meaning

$$y^k + \rho(Ax^k - u^{k+1}) \in \partial \delta_Q(u^{k+1}),$$

and

$$-A^T(y^k + \rho(Ax^{k+1} - u^{k+1})) + N(x^k - x^{k+1}) \in \nabla d_C^2(x^{k+1}).$$

Therefore, by subtracting λAx^{k+1} from both sides of the first inclusion, we arrive at

$$\rho A(x^k - x^{k+1}) \in \partial \delta_Q(u^{k+1}) - y^k - \rho(Ax^{k+1} - u^{k+1}),$$

and

$$N(x^k - x^{k+1}) \in \nabla d_C^2(x^{k+1}) + A^T(y^k + \rho(Ax^{k+1} - u^{k+1})).$$

Thus $\|\nabla_x L_\rho(x^{k+1}, u^{k+1}, y^k)\| \leq \lambda_{\max}(N) \|x^{k+1} - x^k\|$, which proves the second condition with $b = \lambda_{\max}(N)$ and $v^{k+1} := \rho A(x^k - x^{k+1}) \in \partial_u L_\rho(x^{k+1}, u^{k+1}, y^k)$. Hence, the third condition follows.

The fourth condition easily follows since the set Q is closed and therefore $\bar{u} \in Q$, which yields the desired result.

Now, the result follows directly from [12, Theorem 2, p. 1218]. ◀

4 Multiple-Sets Split Feasibility Problems

In this section we study Multiple-Sets Split Feasibility (MSSF) Problems and discuss how the above mentioned methods can be adjusted to solve this extension. Let r be a natural number. Let C and Q_j , $1 \leq j \leq r$, be non-empty and closed subsets of \mathbb{R}^n and \mathbb{R}^{m_j} , respectively. Given also linear mappings $A_j : \mathbb{R}^n \rightarrow \mathbb{R}^{m_j}$, then the MSSF Problem seeks to find a point x^* for which

$$x^* \in C \text{ such that } A_j x^* \in Q_j, \forall 1 \leq j \leq r, \quad (20)$$

we refer the reader for some papers that discuss this extension in the convex setting [29, 21, 31, 22] (which study a more general version of the problem which also allows for multiple sets C). Note that, when $A_j = A$, $1 \leq j \leq r$, for some linear mapping A , then Problem (20) reduced to the following version:

$$x^* \in C \text{ such that } Ax^* \in \bigcap_{j=1}^r Q_j.$$

This problem can be also formulated using similar optimization models (like (SF1)–(SF4)), where each set can be translated using the indicator function or the squared distance function. Therefore, this can yield, in general, many optimization models, but as we discussed above depends on the properties of the set (like convexity) one should decide if to use the indicator function or the distance function. Let's take an example.

As mentioned above one of the optimization models that is very popular in this area is (SF3), and in case of a multiple sets it translates to.

$$\min_{x \in \mathbb{R}^n} \left\{ \sum_{j=1}^r d_{Q_j}^2(A_j x) + \delta_C(x) \right\} = \min_{x \in C} \left\{ \sum_{j=1}^r d_{Q_j}^2(A_j x) \right\}. \quad (21)$$

This model, makes sense when the set C is not necessarily convex, while the sets Q_j , $1 \leq j \leq r$, are convex. The Projected Gradient can be applied for solving this problem, which yields the following extension of the CQ Algorithm

$$x^{k+1} = P_C \left(x^k - \frac{1}{\tau} \sum_{j=1}^r A_j^T (A_j x^k - P_{Q_j}(A_j x^k)) \right),$$

for some step-size $\tau > 0$. This algorithm, as can be seen, tackles the sets Q_j , $1 \leq j \leq r$, in a simultaneous manner, which could result in a slow convergence. A cyclic based algorithm for solving this problem in the convex setting is presented in [36].

- **Remark 8.** (i) It should be noted that all the algorithms presented in Section 2 can be easily generalized to this version of MSSF problems in a simultaneous way similar to the above. The case where there exists multiple sets C is much more complicated and deserves a special study especially in the non-convex setting.
- (ii) Since our analysis is based on non-linear optimization techniques, some of the results in this paper could also be applied for solving non-linear split feasibility problems. The proposed algorithms are different than the ones presented in Non-linear convex feasibility problems were studied for example in [20, 37] and are less restrictive.

To conclude, in this paper, we have studied the split feasibility problem in the non-convex setting through the presentation of four different models that formulate the feasibility problem as an optimization problem using the indicator and/or squared distance functions. We discussed seven algorithms for tackling these formulations, where each one of them tackles the formulation which fits best in sense of the information on the involved sets C and Q (in this abstract paper we have only exploited the feature of convexity). This paper has two main goals: in the theoretical front, we have provided an up-to-date list of algorithms with the current known theoretical results as summarized in the table on page 2. From this table, one can find a comparison between the discussed algorithms in the sense of global convergence, all algorithms but Algorithm 1 have this property (the global convergence of Algorithms 1 seems to be very a challenging task for future research!). In the applied front, this paper provides a source for users that are interested in solving split feasibility problems in the non-convex world. By exploiting the special structure and data information of the particular sets C and Q (like convexity but others could be useful) one can find the algorithm that best fits the applications at hand.

References

- 1 H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Math. Program.*, 116(1-2):5–16, 2009.
- 2 H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for non-convex problems: an approach based on the Kurdyka-Łojasiewicz inequality. *Math. Oper. Res.*, 35(2):438–457, 2010.
- 3 H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Math. Program.*, 137(1-2):91–129, 2013.
- 4 H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, volume 408. Springer, 2011.
- 5 M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. Wiley, 2013.
- 6 A. Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, 2017.
- 7 A. Beck and M. Teboulle. Gradient-based algorithms with applications to signal-recovery problems. *Convex optimization in signal processing and communications*, pages 42–88, 2010.
- 8 D. P. Bertsekas. *Nonlinear Programming: Theory and Algorithms*. Athena Scientific, Belmont, MA, USA, 1995.
- 9 J. Bolte, A. Daniilidis, and A. Lewis. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM J. Optimiz.*, 17(4):1205–1223, 2006.
- 10 J. Bolte, A. Daniilidis, O. Ley, and L. Mazet. Characterizations of Łojasiewicz inequalities: Subgradient flows, talweg, convexity. *T. Am. Math. Soc.*, 362(6):3319–3363, 2010.
- 11 J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program.*, 146(1-2):459–494, 2014.
- 12 J. Bolte, S. Sabach, and M. Teboulle. Non-convex Lagrangian-based optimization: monitoring schemes and global convergence. *Math. Oper. Res.*, 43(4):1210–1232, 2018.
- 13 J. Bolte, S. Sabach, M. Teboulle, and Y. Vaisbourd. First order methods beyond convexity and Lipschitz gradient continuity with applications to quadratic inverse problems. *SIAM J. Optimiz.*, 28(3):2131–2151, 2018.
- 14 C. Byrne. Iterative oblique projection onto convex sets and the split feasibility problem. *Inverse Probl.*, 18(2):441–453, 2002.
- 15 C. Byrne. A unified treatment of some iterative algorithms in signal processing and image reconstruction. *Inverse Probl.*, 20(1):103–120, 2004.
- 16 Y. Censor, T. Bortfeld, B. Martin, and A. Trofimov. A unified approach for inversion problems in intensity-modulated radiation therapy. *Phys. Med. Biol.*, 51(10):2353–2365, 2006.
- 17 Y. Censor and T. Elfving. A multiprojection algorithm using Bregman projections in a product space. *Numer. Algorithms*, 8(2):221–239, 1994.
- 18 C. Chen, T. K. Pong, and L. Tan. A difference-of-convex approach for split feasibility with applications to matrix factorizations and outlier detection. <https://arxiv.org/abs/1903.01101>, 2019.
- 19 D. Gabay and B. Mercier. A multiprojection algorithm using Bregman projections in a product space. *Comput. and Math. Appl.*, 2:221–239, 1976.
- 20 A. Gibali, K. H. Küfer, and P. Süß. Successive linear programming approach for solving the nonlinear split feasibility problem. *J. Nonlinear Convex A.*, 15(2):345–353, 2014.
- 21 A. Gibali, L.-W. Liu, and Y.-C. Tang. Note on the modified relaxation CQ algorithm for the split feasibility problem. *Optim. Lett.*, 12(4):817–830, 2018.
- 22 A. Gibali, D. T. Mai, and N. T. Vinh. A new relaxed CQ algorithm for solving split feasibility problems in Hilbert spaces and its applications. *J. Ind. Manag. Optim.*, 15(2):963–984, 2019.
- 23 R. Glowinski and A. Marroco. Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 9(R2):41–76, 1975.
- 24 B. He and X. Yuan. On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. *SIAM J. Numer. Anal.*, 50(2):700–709, 2012.
- 25 K. Kurdyka. On gradients of functions definable in o-minimal structures. *Ann. Inst. Fourier*, 48(3):769–783, 1998.
- 26 T. Liu, T. K. Pong, and A. Takeda. A successive difference-of-convex approximation method for a class of nonconvex nonsmooth optimization problems. *Math. Program.*, 176(1-2):339–367, 2019.
- 27 S. Łojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. *Les équations aux dérivées partielles*, 117:87–89, 1963.
- 28 D. R. Luke, N. H. Thao, and M. K. Tam. Quantitative convergence analysis of iterated expansive, set-valued mappings. *Math. Oper. Res.*, 43(4):1143–1176, 2018.
- 29 E. Masad and S. Reich. A note on the multiple-set split convex feasibility problem in Hilbert space. *J. Nonlinear Convex A.*, 8(3):367–371, 2007.

- 30 T. Pock and S. Sabach. Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems. *SIAM J. Imaging Sci.*, 9(4):1756–1787, 2016.
- 31 M. Raeisi, G. Z. Eskandani, and M. Eslamian. A general algorithm for multiple-sets split feasibility problem involving resolvents and Bregman mappings. *Optimization*, 67(2):309–327, 2018.
- 32 R. T. Rockafellar and Roger J-B. Wets. *Variational analysis*, volume 317. Springer, 2009.
- 33 S. Sabach and M. Teboulle. Lagrangian methods for composite optimization. *Handb. Numer. Anal.*, 20, 2019.
- 34 S. Sabach, M. Teboulle, and S. Voldman. A smoothing alternating minimization-based algorithm for clustering with sum-min of Euclidean norms. *Pure Appl. Funct. Anal.*, 3:653–679, 2018.
- 35 R. Shefi and M. Teboulle. Rate of convergence analysis of decomposition methods based on the proximal method of multipliers for convex minimization. *SIAM J. Optimiz.*, 24(1):269–297, 2014.
- 36 F. Wang and H.-K. Xu. Cyclic algorithms for split feasibility problems in Hilbert spaces. *Nonlinear Anal.-Theor.*, 74(12):4105–4111, 2011.
- 37 J. Xu, E. C. Chi, M. Yang, and K. Lange. A majorization–minimization algorithm for split feasibility problems. *Comput. Optim. Appl.*, 71(3):795–828, 2018.