Stochastic Block-Iterative Parallel Subgradient Projections Method with Super Relaxations for Convex Feasibility

Patrick L. Combettes
Department of Mathematics
North Carolina State University
Raleigh, NC 27695-8205, USA

Javier I. Madariaga
Department of Mathematics
North Carolina State University
Raleigh, NC 27695-8205, USA

***Abstract*—The convex feasibility problem consists of finding a point in the intersection of closed convex sets. We propose a new type of algorithm to solve it in which randomly selected blocks of subgradient projectors are activated in parallel at each iteration and averaged through an extrapolation process. A key novelty is that the update is obtained by a random super relaxation step which allows for unbounded relaxation parameters, in sharp contrast with existing methods where they are deterministic and bounded by 2. Almost sure convergence is proved without requiring any regularity assumptions. Additionally, we establish mean-square convergence rates under linear regularity conditions. Numerical applications to signal and image recovery are demonstrated, which illustrate the benefits of super relaxations and random set activation.**

***Index Terms*—Convex feasibility, signal recovery, subgradient projection, stochastic algorithm.**

I. Introduction

The convex feasibility problem is a powerful formalism which captures problems in various areas of signal and image processing, as well as in inverse problems [5], [8]. While our results remain valid for problems with infinitely many sets in general Hilbert spaces [11, Section 5], we focus for simplicity on the following formulation in the Euclidean space $\mathbb{R}^N$.

Problem 1 *For every* $k \in \{1,\dots,p\}$, $f_k\colon \mathbb{R}^N \to \mathbb{R}$ *is a convex function and* $C_k = \big\{x \in \mathbb{R}^N \mid f_k(x) \leqslant 0\big\}$. *It is assumed that* $Z = \bigcap_{1\leqslant k\leqslant p} C_k \neq \varnothing$. *The task is to*

$$\text{find } x \in \mathbb{R}^N \text{ such that } x \in Z. \tag{1}$$

There exists a vast amount of deterministic frameworks dedicated to solve Problem 1 by using exact projections and subgradient projections, see, e.g., [1], [3], [6], [7], [9], [10], [20]. Among them, those that employ extrapolated averages of blocks of (subgradient) projections have shown better results and more flexibility in practice [3], [7], [9]. In recent years, interest has grown towards selecting the sets activated at each iteration in a random rather than deterministic fashion, e.g., [4], [13], [16], [17], [21]. However, none of those results has established the almost-sure convergence of the sequence of iterates to a solution to Problem 1 when using stochastic extrapolation algorithms.

The main contribution of the present paper is to introduce a stochastic extension of the extrapolated method of parallel subgradient projections of [9] with the following original features:

- At each iteration it activates only a block of randomly selected sets in $(C_k)_{1\leqslant k\leqslant p}$, which are combined via a randomly weighted extrapolated average.
- It introduces for the first time the concept of random *super relaxation*, whereby the relaxation parameter can be greater than 2.
- It guarantees almost-sure convergence and convergence in mean of the sequence of iterates to a solution to Problem 1 without any additional assumptions on the sets $(C_k)_{1\leqslant k\leqslant p}$, the functions $(f_k)_{1\leqslant k\leqslant p}$, or their subgradients.
- Under linear regularity of the subgradient projectors, it provides linear mean-square convergence.

The algorithm is presented in Section II-C and applied to signal and image restoration problems in Section III.

II. Stochastic extrapolated method of parallel subgradient projections

A. Notation

The scalar product and associated norm in $\mathbb{R}^N$ are denoted by $\langle\cdot\,|\,\cdot\rangle$ and $\|\cdot\|$ respectively. Let C be a nonempty closed convex subset of $\mathbb{R}^N$. Then proj_C denotes the projection operator onto C and d_C denotes its distance function. Let $f\colon \mathbb{R}^N \to \mathbb{R}$ be a convex function. The subdifferential of f at $x \in \mathbb{R}^N$ is the set

$$\partial f(x) = \big\{u \in \mathbb{R}^N \mid (\forall z \in \mathbb{R}^N)\ \langle z - x\,|\,u\rangle + f(x) \leqslant f(z)\big\}. \tag{2}$$

We refer to [2] for background on convex analysis and optimization. The underlying probability space is $(\Omega, \mathcal{F}, P)$ and $\mathcal{B}$ denotes the Borel σ-algebra of $\mathbb{R}^N$. An $\mathbb{R}^N$-valued

This work was supported by the National Science Foundation under grant CCF-2211123.

random variable is a measurable mapping $x\colon (\Omega,\mathcal{F}) \to (\mathbb{R}^N,\mathcal{B})$. The σ-algebra generated by a family Φ of random variables is denoted by $\sigma(\Phi)$. The independence of a random variable $x\colon \Omega \to \mathbb{R}^N$ with respect to $\sigma(\Phi)$ is denoted by $x \perp\!\!\!\perp \sigma(\Phi)$. Given $A \subset \mathbb{R}^N$, we set $[x \in A] = \{\omega \in \Omega \mid x(\omega) \in A\}$. We use sans-serif letters for deterministic variables and italicized serif letters for random variables.

B. Subgradient projectors

Consider the setting of Problem 1. For every $k \in \{1,\dots,p\}$, let $s_k\colon \mathbb{R}^N \to \mathbb{R}^N\colon x \mapsto s_k(x) \in \partial f_k(x)$ be a selection of ∂f_k and let

$$G_k\colon \mathbb{R}^N \to \mathbb{R}^N\colon x \mapsto \begin{cases} x - \dfrac{f_k(x)}{\|s_k(x)\|^2}s_k(x), & \text{if } f_k(x) > 0;\\ x, & \text{if } f_k(x) \leqslant 0 \end{cases}$$

be the subgradient projector onto C_k associated with (f_k, s_k) [2, Definition 29.40].

Subgradient projectors extend the classical projection operators in the following sense. Let C be a nonempty closed and convex subset of $\mathbb{R}^N$ and suppose that $f_k = d_C$. Then $C_k = C$ and $G_k = \operatorname{proj}_C$ [2, Example 29.44]. Their importance in solving Problem 1 stems from the fact that subgradient projectors are generally much easier to implement than exact ones.

C. Algorithm and convergence

We propose the following stochastic extension of the extrapolated parallel block-iterative subgradient projection algorithms of [9], [10]. It features three levels of stochasticity at each iteration n:

- The block of sets to be activated is randomly selected.
- The subgradient projections are averaged with random weights $(\beta_{i,n})_{1\leqslant i\leqslant M}$.
- The relaxation parameter λ_n is random and not confined to the interval $]0,2[$ as in traditional methods. This *super relaxation* scheme will be shown to result in a speed-up of convergence.

Algorithm 2 Consider the setting of Problem 1. Let $x_0 \in L^2(\Omega,\mathcal{F},P;\mathbb{R}^N)$, $M \in \mathbb{N}\smallsetminus\{0\}$, $\delta \in]0,1/M[$, and $\rho \in [2,+\infty[$. Iterate

for $n = 0,1,\dots$
  for $i = 1,\dots,M$
    $k_{i,n} \sim \text{uniform}(\{1,\dots,p\})$ and $k_{i,n} \perp\!\!\!\perp \sigma(x_0,\dots,x_n)$
    $p_{i,n} = G_{k_{i,n}}x_n$
  $(\beta_{i,n})_{1\leqslant i\leqslant M}$ in $L^1(\Omega,\mathcal{F},P;[\delta,1])$ with $\sum_{i=1}^M \beta_{i,n} = 1$
  $p_n = \sum_{i=1}^M \beta_{i,n}p_{i,n}$
  $L_n = \begin{cases} \dfrac{\sum_{i=1}^M \beta_{i,n}\|p_{i,n}-x_n\|^2}{\|p_n-x_n\|^2}, & \text{if } p_n \neq x_n;\\ 1, & \text{otherwise} \end{cases}$
  $a_n = x_n + L_n(p_n - x_n)$
  $\lambda_n \in L^1(\Omega,\mathcal{F},P;]0,\rho])$ and $\lambda_n \perp\!\!\!\perp \sigma(x_0,\dots,x_n,p_n)$
  $x_{n+1} = x_n + \lambda_n(a_n - x_n)$.

In the foregoing algorithm, M is the batch size, i.e., the number of activated sets, p_n is the standard average of the selected subgradient projections, $L_n \geqslant 1$ is the extrapolation parameter, a_n is the extrapolated average, and λ_n is the relaxation parameter, which can exceed the standard bound 2 imposed by deterministic methods.

The following convergence result is based on the stochastic framework of [11, Section 5] and it ensures almost sure convergence of the sequence constructed by Algorithm 2 to a solution to Problem 1 without additional assumptions on the sets $(C_k)_{1\leqslant k\leqslant p}$.

Theorem 3 *Let $(x_n)_{n\in\mathbb{N}}$ be the sequence constructed by Algorithm 2 and suppose that*

$$(\exists\, \mu \in]0,1[) \quad \inf_{n\in\mathbb{N}} E\big(\lambda_n(2-\lambda_n)\big) \geqslant \mu. \tag{3}$$

Then there exists $x \in L^2(\Omega,\mathcal{F},P;Z)$ such that $(x_n)_{n\in\mathbb{N}}$ converges to x in $L^1(\Omega,\mathcal{F},P;\mathbb{R}^N)$ and P-a.s.

Under additional assumptions, we obtain the following convergence rate.

Theorem 4 *Let $(x_n)_{n\in\mathbb{N}}$ be the sequence constructed by Algorithm 2. Suppose that that* (3) *holds, together with one of the following:*

(i) *There exists $\chi \in]0,1[$ such that, for every $n \in \mathbb{N}$,*

$$E\big(d_Z^2(x_{n+1}) \,\big|\, \sigma(x_0,\dots,x_n)\big) \leqslant \chi d_Z^2(x_n) \ \text{ P-a.s.} \tag{4}$$

(ii) *Linear regularity for $(G_k)_{1\leqslant k\leqslant p}$ holds, i.e., there exists $\nu \in [1/p,+\infty[$ such that*

$$(\forall x \in \mathbb{R}^N) \quad d_Z^2(x) \leqslant \nu \sum_{k=1}^p \|G_kx - x\|^2, \tag{5}$$

in which case we set $\chi = 1 - \mu\delta(1-\sqrt{1-\mu})^2/(\rho^2\nu p)$.

Then there exists $x \in L^2(\Omega,\mathcal{F},P;Z)$ such that $(x_n)_{n\in\mathbb{N}}$ converges to x in $L^2(\Omega,\mathcal{F},P;\mathbb{R}^N)$ and P-a.s. Further,

$$(\forall n \in \mathbb{N}) \quad E\|x_n - x\|^2 \leqslant 4\chi^n E d_Z^2(x_0). \tag{6}$$

D. Literature comparison

Only a few works have established the almost sure convergence to a solution of Problem 1 by using stochastic projections methods and they do so in specialized settings:

- In [17], a particular case of Algorithm 2 is proposed where $M = 1$ (only one set is activated at each iteration), $\lambda_n \equiv 1$, and exact projectors are used. Almost sure convergence to a solution is shown. This result is also found in [4] and in [13]; the latter also shows convergence in $L^2(\Omega,\mathcal{F},P;\mathbb{R}^N)$. When the sets $(C_k)_{1\leqslant k\leqslant p}$ are half-spaces or when the interior of Z is nonempty, [17] provides a rate for convergence in $L^2(\Omega,\mathcal{F},P;\mathbb{R}^N)$.
- A particular case of Algorithm 2 is analyzed in [18]. It provides almost sure convergence by using subgradient projectors for $M = 1$ and deterministic parameters $(\lambda_n)_{n\in\mathbb{N}}$ in $]0,2[$. Moreover, linear regularity

assumptions of the form of (5) are required, as well as the assumption that the subgradients of $(f_k)_{1\leqslant k\leqslant p}$ are uniformly bounded on $\mathbb{R}^N$.

The above references provide almost sure convergence of the sequence of iterates only for $M = 1$ and, therefore, no parallel activation nor extrapolation is applied. In this regard, one of the novelties of Theorem 3 is that it guarantees almost sure convergence for $M > 1$ and with extrapolation. We now comment on related works, though they have not established almost sure convergence

- In [16] and [14] an algorithm similar to Algorithm 2 is studied with the following restrictions: deterministic relaxations $(\lambda_n)_{n\in\mathbb{N}}$ in $]0, 2[$, iteration-independent fixed deterministic weights $\beta_{i,n} \equiv 1/M$, and exact projections instead of subgradient projections. Mean-square rates of convergence are established under additional linear regularity assumptions, as well as ergodic convergence results. Almost sure convergence is not proved.
- Similarly, [15] and [19] use a deterministic relaxation sequence $(\lambda_n)_{n\in\mathbb{N}}$ in $]0, 2[$, iteration-independent fixed deterministic weights $\beta_{i,n} \equiv 1/M$, and they require linear regularity assumptions and uniform boundedness of the subgradients of $(f_k)_{1\leqslant k\leqslant p}$ on $\mathbb{R}^N$. They state ergodic convergence and provide rates of convergence in mean-square. Nevertheless, they do not establish almost sure convergence of the sequence of iterates.

These references have in common the addition of linear regularity assumptions, deterministic relaxation parameters $(\lambda_n)_{n\in\mathbb{N}}$ and weights $(\beta_{i,n})_{n\in\mathbb{N},1\leqslant i\leqslant M}$, and the fact that they do not guarantee almost sure convergence of the iterates. Thus, even if Theorem 4 were to assume linear regularity, it would still be novel since it guarantees almost sure convergence and provides rates of convergence in mean-square. We also underline that we do not require uniformly boundedness of the subgradients.

E. Random super relaxation parameters

In both Theorems 3 and 4, random relaxations satisfying (3) are used. When $(\lambda_n)_{n\in\mathbb{N}}$ is deterministic, (3) reduces to

$$(\exists\, \varepsilon \in]0, 1[)(\forall n \in \mathbb{N}) \quad \lambda_n \in [\varepsilon, 2 - \varepsilon]\,, \tag{7}$$

which corresponds to the standard range found in the literature. Even in this case, the results in Theorems 3 and 4 are novel. However, a notable innovation introduced in Algorithm 2 is the use of random super relaxation sequences $(\lambda_n)_{n\in\mathbb{N}}$ satisfying (3). There are various ways to construct super relaxation parameters; here are a few examples.

Example 5 Fix $n \in \mathbb{N}$ and $\mu \in]0, 1[$. Let $\zeta \in]0, 1[$, $\alpha \in]0, +\infty[$, and $\beta \in]0, +\infty[$. We assume that $P([\lambda_n = \alpha]) = \zeta$ and $P([\lambda_n = \beta]) = 1 - \zeta$. Then λ_n satisfies (3) if

$$\zeta\big(\alpha(2 - \alpha)\big) + (1 - \zeta)\big(\beta(2 - \beta)\big) \geqslant \mu. \tag{8}$$

This strategy allows us to use a large number $\alpha \in]0, +\infty[$ as long as the associated probability ζ is small. In particular, (8) holds for $\mu = 0.1$, $\zeta = 1/7$, $\alpha = 2.5$, and $\beta = 1.8$. In such a case, λ_n is a super relaxation parameter with $E\lambda_n = 1.9$. Another selection for $\mu = 0.01$ is to set $\zeta = 0.5$, $\alpha = 2.3$, and $\beta = 1.5$, which again gives $E\lambda_n = 1.9$.

Example 6 Fix $n \in \mathbb{N}$ and $\mu \in]0, 1[$. Let $\alpha \in]0, +\infty[$ and $\beta \in]0, +\infty[$ be such that $\alpha < \beta$, and we assume that $\lambda_n \sim \text{uniform}([\alpha, \beta])$. Then λ_n satisfies (3) if

$$3\alpha + 3\beta - (\alpha^2 + \alpha\beta + \beta^2) \geqslant 3\mu. \tag{9}$$

This condition holds in particular when $\mu = 0.1$, $\alpha = 1.5$, and $\beta = 2.3$. This produces a super relaxation parameter λ_n with $E\lambda_n = 1.9$.

III. Numerical experiments

A. Signal restoration

The goal is to recover the original signal $\overline{x} \in \mathbb{R}^N$ ($N = 1024$) shown in Fig. 1(a) from 20 noisy observations $(r_k)_{1\leqslant k\leqslant 20}$ given by

$$(\forall k \in \{1, \dots, 20\}) \quad r_k = L_k\overline{x} + w_k \tag{10}$$

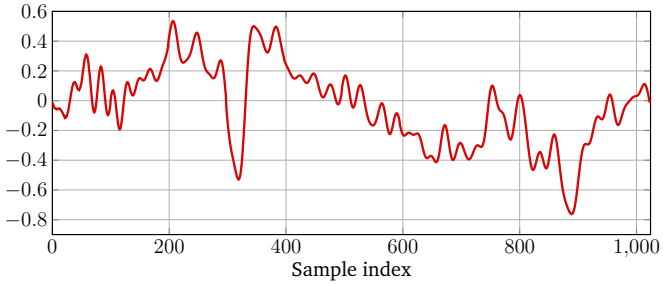
where $L_k \colon \mathbb{R}^N \to \mathbb{R}^N$ is a known linear operator, $\eta_k \in]0, +\infty[$, and $w_k \in [-\eta_k, \eta_k]^N$ is a bounded random noise vector. The parameters $(\eta_k)_{1\leqslant k\leqslant 20} \in]0, +\infty[^{20}$ are known. The operators $(L_k)_{1\leqslant k\leqslant 20}$ are Gaussian convolution filter with zero mean and standard deviation taken uniformly in $[10, 30]$, $\eta_k = 0.1$, and w_k is taken uniformly in $[-\eta_k, \eta_k]^N$. Set, for every $k \in \{1, \dots, 20\}$ and every $j \in \{1, \dots, N\}$,

$$C_{k,j} = \big\{x \in \mathbb{R}^N \mid -\eta_n \leqslant \langle L_k x - r_k \,|\, e_j\rangle \leqslant \eta_n\big\}. \tag{11}$$

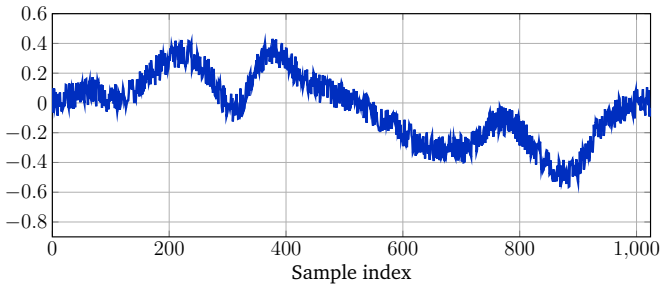
Since the intersection of these sets is nonempty and their projectors are computable explicitly [2, Example 29.21], we solve the feasibility problem

$$\text{find } x \in \mathbb{R}^N \text{ such that } (\forall k \in \{1, \dots, 20\})(\forall j \in \{1, \dots, N\})\ x \in C_{k,j} \tag{12}$$

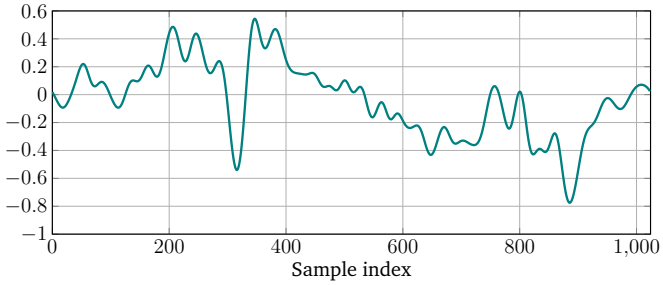
by Algorithm 2 implemented with exact projectors. We run two instances with $x_0 = 0$. In the first one, $M = 1$ and we compare four relaxation schemes: $\lambda_n \equiv 1$, which leads the almost sure convergence result of [17] (see also [13]), $\lambda_n \equiv 1.9$, the random super relaxation strategy of Example 5 where, for every $n \in \mathbb{N}$, $P([\lambda_n = 2.3]) = 1/2$ and $P([\lambda_n = 1.5]) = 1/2$, and strategy of Example 6 where, for every $n \in \mathbb{N}$, $\lambda_n \sim \text{uniform}([1.5, 2.3])$. Note that both random super relaxations schemes satisfy $E\lambda_n \equiv 1.9$. In the second instance $M = 128$ and we compare the four relaxation strategies as above. Fig. 2 displays the normalized error versus execution time for a typical realization.



(a)

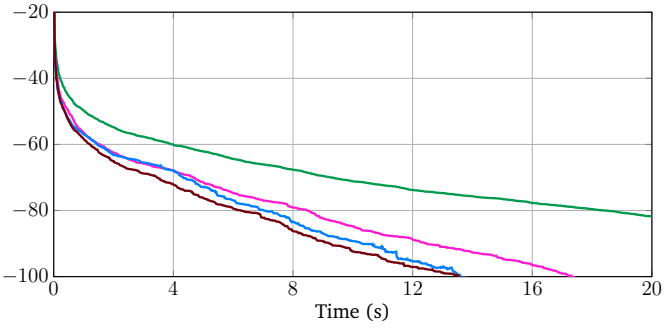


(b)

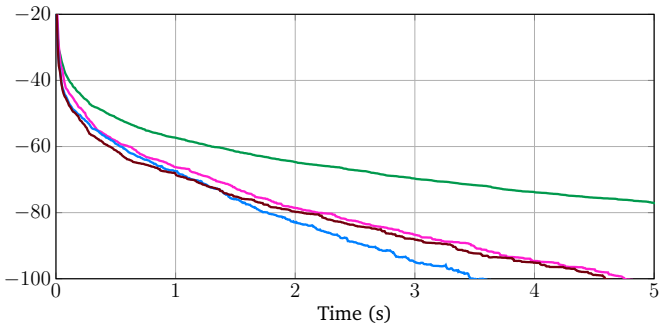


(c)

Fig. 1: Experiment of Section III-A. (a): Original signal $\bar{x}$. (b): Noisy observation r_1. (c): Solution produced by Algorithm 2.



(a)



(b)

Fig. 2: Experiment of Section III-A. Normalized error $20\log(\|x_n - x_\infty\|/\|x_0 - x_\infty\|)$ (dB) versus execution time (s) on a single core machine. Green: $\lambda_n \equiv 1$. Magenta: $\lambda_n \equiv 1.9$. Blue: $\mathsf{P}([\lambda_n = 1.5]) = 1/2$ and $\mathsf{P}([\lambda_n = 2.3]) = 1/2$. Brown: $\lambda_n \sim \text{uniform}([1.5, 2.3])$. (a): $M = 1$. (b): $M = 128$.

B. Image restoration

The goal is to recover the original image $\bar{x} \in \mathbb{R}^{N\times N}$ ($N = 256$) shown in Fig. 3(a) from three observations $\{r_1, r_2, r_3\}$ which are given by the degradation of $\bar{x}$ via a Gaussian kernel with a standard deviation of 6 and the addition of random noise. The noise distribution is uniform$([0,5]^{N\times N})$. Let L be the block-Toeplitz matrix associated with the convolutional blur. Then

$$(\forall k \in \{1,2,3\})\ r_k = L\bar{x} + w_k, \quad \text{where} \quad w_k \sim \text{uniform}([0,5]^{N\times N}). \tag{13}$$

The random variables $(w_k)_{1\leqslant k\leqslant 3}$ are i.i.d. Therefore, as shown in [12], for every $k \in \{1,2,3\}$, with a 95% confidence coefficient

$$\bar{x} \in C_k = \left\{x \in \mathbb{R}^{N\times N} \mid \|r_k - Lx\|^2 \leqslant \xi\right\}, \tag{14}$$

where $\xi = N^2 E|u|^2 + 1.96N\sqrt{E|u|^4 - E^2|u|^2}$ with $u \sim$ uniform$([0,5])$. For every $k \in \{1,2,3\}$, we compute the subgradient projector onto C_k via the function $f_k : x \mapsto \|r_k - Lx\|^2 - \xi$. In addition, the boundedness on pixel values is incorporated as the property set $C_4 = [0,255]^{N\times N}$. Finally, it is assumed that the discrete Fourier transform $\mathfrak{F}(\bar{x})$ of $\bar{x}$ is known on a portion of its support for low frequencies in both directions. That is, let S be the set of frequency pairs $\{0,\dots,N/8-1\}^2$ as well as those resulting from the symmetry properties of the 2D discrete Fourier transform of real images. The associated set is $C_5 = \left\{x \in \mathbb{R}^{N\times N} \mid \mathfrak{F}(x)1_S = \mathfrak{F}(\bar{x})1_S\right\}$ and its projection is given by $\text{proj}_{C_5} : x \mapsto \mathfrak{F}^{-1}(\mathfrak{F}(\bar{x})1_S + \mathfrak{F}(x)1_{\complement S})$. We run Algorithm 2 with $x_0 = 0$ and $M = 2$. We compare four relaxation strategies: $\lambda_n \equiv 1$, $\lambda_n \equiv 1.9$, the random super relaxation strategy where, for every $n \in \mathbb{N}$, $\mathsf{P}([\lambda_n = 2.5]) = 1/7$ and $\mathsf{P}([\lambda_n = 1.8]) = 6/7$, and the random super relaxation where, for every $n \in \mathbb{N}$, $\lambda_n \sim$ uniform$([1.5, 2.3])$. Fig. 4 displays the normalized error versus execution time for a typical realization.

First, these experiments show the advantage of using random blocks, as reflected in the execution time of the algorithm, even on a single-core machine. This performance can naturally be further improved if Algorithm 2 is implemented on a multi-core architecture where, at each iteration, each subgradient projector is assigned

(a) (b)

(c)

Fig. 3: Experiment of Section III-B. (a) Original image $\bar{x}$. (b) Noisy observation r_1. (c) Solution produced by Algorithm 2.

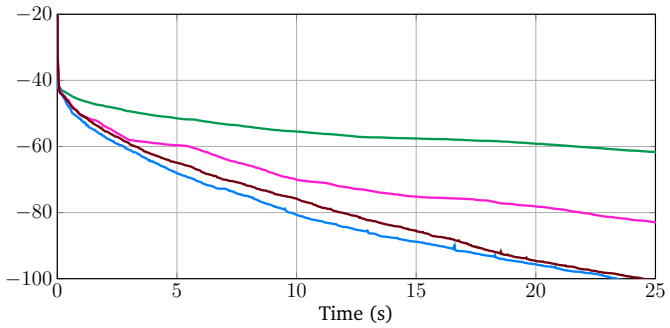


Fig. 4: Experiment of Section III-B using $\mathsf{M} = 2$. Normalized error $20\log(\|x_n - x_\infty\|/\|x_0 - x_\infty\|)$ (dB) versus execution time (s) on a single core machine. Green: $\lambda_n \equiv 1$. Magenta: $\lambda_n \equiv 1.9$. Blue: $\mathsf{P}([\lambda_n = 1.8]) = 6/7$ and $\mathsf{P}([\lambda_n = 2.5]) = 1/7$. Brown: $\lambda_n \sim \text{uniform}([1.5, 2.3])$.

to a dedicated core and all the cores work in parallel. The numerical results also show the benefits of using relaxation parameters bigger than 1 with extrapolation. This behavior has been already observed for deterministic methods, see, e.g., [3], [7], [9], [20]. Finally, our experimental results suggest that the use of the proposed random super relaxation scheme further improves the speed of convergence.

IV. Conclusion

We have introduced a stochastic block-iterative extrapolated parallel subgradient projections method for solving the convex feasibility problem. Unlike the state of the art, the proposed method guarantees the almost sure convergence of the sequence of iterates to a solution when more than one set is activated at each iteration without assuming regularity conditions on the sets or uniformly boundedness on the subdifferentials. Additionally, it introduces random super relaxation parameters which may exceed 2. This feature was shown to be numerically advantageous.

References

[1] H. H. Bauschke and J. M. Borwein, On projection algorithms for solving convex feasibility problems, *SIAM Rev.*, vol. 38, pp. 367–426, 1996.

[2] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2nd ed. Springer, New York, 2017.

[3] H. H. Bauschke, P. L. Combettes, and S. G. Kruk, Extrapolation algorithm for affine-convex feasibility problems, *Numer. Algorithms*, vol. 41, pp. 239–274, 2006.

[4] L. Briceño-Arias, J. Deride, and C. Vega, Random activations in primal-dual splittings for monotone inclusions with a priori information, *J. Optim. Theory Appl.*, vol. 192, pp. 56–81, 2022.

[5] C. L. Byrne, *Iterative Optimization in Inverse Problems*. CRC Press, Boca Raton, FL, 2014.

[6] A. Carmi, Y. Censor, and P. Gurfil, Convex feasibility modeling and projection methods for sparse signal recovery, *J. Comput. Appl. Math.*, vol. 236, pp. 4318–4335, 2012.

[7] Y. Censor, W. Chen, P. L. Combettes, R. Davidi, and G. T. Herman, On the effectiveness of projection methods for convex feasibility problems with linear inequality constraints, *Comput. Optim. Appl.*, vol. 51, pp. 1065–1088, 2012.

[8] P. L. Combettes, The convex feasibility problem in image recovery, in: *Advances in Imaging and Electron Physics* (P. Hawkes, Ed.), vol. 95, pp. 155–270. Academic Press, New York, 1996.

[9] P. L. Combettes, Convex set theoretic image recovery by extrapolated iterations of parallel subgradient projections, *IEEE Trans. Image Process.*, vol. 6, pp. 493–506, 1997.

[10] P. L. Combettes, Hilbertian convex feasibility problems: Convergence of projection methods, *Appl. Math. Optim.*, vol. 35, pp. 311–330, 1997.

[11] P. L. Combettes and J. I. Madariaga, A geometric framework for stochastic iterations, 2025. https://arxiv.org/abs/2504.02761

[12] P. L. Combettes and H. J. Trussell, The use of noise properties in set theoretic estimation, *IEEE Trans. Signal Process.*, vol. 39, pp. 1630–1641, 1991.

[13] V. R. Kostić and S. Salzo, The method of randomized Bregman projections for stochastic feasibility problems, *Numer. Algorithms*, vol. 93, pp. 1269–1307, 2023.

[14] I. Necoara, Stochastic block projection algorithms with extrapolation for convex feasibility problems, *Optim. Methods Softw.*, vol. 37, pp. 1845–1875, 2022.

[15] I. Necoara and A. Nedić, Minibatch stochastic subgradient-based projection algorithms for feasibility problems with convex inequalities, *Comput. Optim. Appl.*, vol. 80, pp. 121–152, 2021.

[16] I. Necoara, P. Richtárik, and A. Patrascu, Randomized projection methods for convex feasibility: Conditioning and convergence rates, *SIAM J. Optim.*, vol. 29, pp. 2814–2852, 2019.

[17] A. Nedić, Random projection algorithms for convex set intersection problems, *Proc. 49th IEEE Conf. Decis. Control*, pp. 7655–7660, 2010.

[18] A. Nedić, Random algorithms for convex minimization problems, *Math. Program.*, vol. B129, pp. 225–253, 2011.

[19] A. Nedić and I. Necoara, Random minibatch subgradient algorithms for convex problems with functional constraints, *Appl. Math. Optim.*, vol. 80, pp. 801–833, 2019.

[20] G. Pierra, Decomposition through formalization in a product space, *Math. Program.*, vol. 28, pp. 96–115, 1984.

[21] T. Strohmer and R. Vershynin, A randomized Kaczmarz algorithm with exponential convergence, *J. Fourier Anal. Appl.*, vol. 15, pp. 262–278, 2009.