POSTPRINT VERSION. The final version is published here :
Hachemi, S. et Massicotte, D. (2018, 19-20 Sept. 2018). Boolean Compressive Sensing: An Approximate Trust Region Reconstruction. Communication présentée 2018 7th International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur. doi: 10.1109/ICCCE.2018.8539329

Boolean Compressive Sensing: An Approximate Trust Region Reconstruction

Sofiane Hachemi
Electrical and Computer Engineering Department
Université du Québec à Trois-Rivières
Trois-Rivières, Québec, Canada, G9A 5H7
sofiane.hachemi@uqtr.ca

Daniel Massicotte
Electrical and Computer Engineering Department
Université du Québec à Trois-Rivières
Trois-Rivières, Québec, Canada, G9A 5H7
daniel.massicotte@uqtr.ca

***Abstract*— In this paper, we propose a direct nonlinear optimization method to solve the Boolean Compressive Sensing (BCS) problem for large signals when sparsity level is unknown. While traditional CS results from linear Algebra, BCS, is given by logical operation in the Boolean workspace. To overcome this inconvenience, we relax the problem in an equivalent formulation in the Real workspace using appropriate modeling as a first step. Thereafter we turn out the problem in an unconstrained form that can be solved directly by nonlinear optimization method called Trust Region methods (TRM). Our solution is based on an Approximate version of (TRM). Numerical results are presented to sustain efficiency of our proposal.**

***Keywords*— Boolean Workspace, Compressive Sensing (CS), Approximate Trust Region Methods, Nonlinear Optimization.**

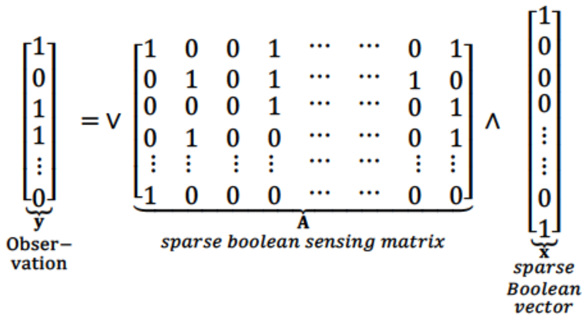

Fig. 1. Boolean Compressive Sensing Model

I. Introduction

Boolean compressive sensing (BCS) is a technique where all operations of traditional compressive sensing are done exclusively in a Boolean workspace (see Fig. 1.).

Such a technique consists to reduce a sparse Boolean signal dimensionality by a compressive set of Boolean observations. Those observations result from a logical vector-matrix multiplication of the Boolean sources-signal by a Boolean sparse sensing matrix in the Boolean algebra. Disjunction and conjunction replace the traditional addition and multiplication. Let's us suppose that an efficient solution could exist. In this case, the simplicity of BCS, allows its potential application in diverse areas such a channel coding, non-orthogonal codes processing, fault discovery, genomics, classification, machine learning or blood screening (groups testing) [2]– [6], and so on…

Specifically, in digital communication field and without loss of generality, many scenarios can be expressed through BCS model. For example, a group of passive sensors, observed by reduced number of receivers, can directly be linked to a basic BCS. In more complex way, we can consider reducing the bandwidth or increase throughput needed to transmit sparse binary messages, seamless rate adaptation (SRA) [7], is a good example, even if the coding schema is not done in the Boolean workspace. Moreover, when it comes to multiusers system, solving such a problem can find its application directly in the technique of superimposed codes within the special case of Boolean nonorthogonal codes. The authors in [8] presented a frame of work where they lay out the connection between BCS and Boolean superimposed codes . Those are just few applications were BCS plays a key role to model the input/output functioning. The scenarios described above share the same modelization issue, however we need to give answer to the structural problem of this model. Firstly, what is the best way to construct the Boolean sensing matrix with minimum number of tests that guarantee a faithful reconstruction of the original Boolean message, in other words, the design of the optimal Boolean sensing matrix needs to be studied. Secondly, how can we proceed to propose a simple but efficient algorithm that is insensitive to time varying of the sparsity level and/or noise corruption. Thirdly, quantifying such a method would be helpful for any hardware implementation. All those interrogations are waiting to find answers. In addition, recent contributions are not focusing on the importance of BCS, much more attention is given to the 1-bit coding CS and its applications. Those are two approaches **completely different**.

It turns out that few works have been dedicated to the reconstruction process of BCS in contrast with 1-bit CS. In [11], the authors prospected the capacity of traditional linear programing algorithm to reconstruct noiseless Boolean signal. A dense Boolean matrix is used as sensing matrix and the recovering was performed via ℓ_1 linear programming by relaxing the Boolean constraint to the polytope inequality; $\mathrm{x} \in \mathbb{B}^n \dashrightarrow 0 \leq x \leq 1$, and fractioning the test pool A: (here defining sensing matrix) into two groups $\mathrm{A}_i{}^{fail}\mathrm{x} = 0, \mathrm{A}_i{}^{success}\mathrm{x} = 0, i = 1,2,..m$ where $\mathbb{B} = \{0,1\}$. Experiments show us that the approach suffer of non-accuracy and a random result with no guaranty of convergence and can't be used for large scale signals due to the saturation problem. In recent work [17], we addressed a solution given through nonlineal programming using Langrangian multipliers and assume that sparsity level of input signal to be time varying and unknown.

This paper, can be viewed as a competitive version of the previous work (TR-BCS) [17 that suffer of a computing due to the Lagrange multipliers update needed to compute the global solution and make it computationally expensive. We assume the same working conditions with noise-free observations and unknown sparsity level. The case of noise corruption will be addressed in future work. In this paper, we propose a new algorithm based only on simple approximate trust region model [18], to recover source signals.

We organized the paper in this way: Section II presents a quick review of the fundamental modeling of the Boolean CS and Boolean sensing matrix design. In Section III, we introduce the proposed algorithm based on approximate trust region. In Section IV, evaluations based on simulations are presented. Finally, we conclude this paper in Section V with some recommendation for future works.

II. BOOLEAN CS FRAMEWORK

A. The BCS-clauses Model

Let $\mathbf{x} \in \mathbb{B}^n$ any Boolean sparse signal and $\mathbf{A} \in \mathbb{B}^{m \times n}$ a sparse Boolean sensing matrix with $m < n$. The BCS problem can be modeled using (AND,OR and XOR) clauses: the Boolean measurement $\mathbf{y} \in \mathbb{B}^m$ where $\mathbb{B} = \{0,1\}$ is given by the logical matrix-vector product:

$$\mathbf{y} = \vee (\mathbf{A} \wedge \mathbf{x}) \qquad (1)$$

i.e.

$$y_i = \bigvee_{j=1}^{n} (A_{ij} \wedge x_j)$$

When measurements are noise-affected, then:

$$\mathbf{y} = \vee (\mathbf{A} \wedge \mathbf{x}) \oplus \mathbf{b}, \qquad (2)$$

We assume that $\vee \equiv$ OR, $\wedge \equiv$ AND and $\oplus \equiv$ XOR are a logical operators and $\mathbf{b}$ is a Bernoulli noise vector. As in traditional CS, the goal is to recover a sparse signal $\mathbf{x}$ with minimum nonzero components from the observations $\mathbf{y}$ only under the assumption that $\mathbf{A}$ is known. This problem is formulated as follow:

$$\min \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \mathbf{y} = \vee (\mathbf{A} \wedge \mathbf{x}), \qquad (3)$$

where $\|\cdot\|_0$, is the ℓ_0 pseudonorm, defining the number of non-zero elements of its argument. Moreover, since coding the information of $\mathbf{x}$ inside a reduced length of observations leads to create an under determination of the problem and so on, the solution becomes non-unique.

B. Boolean Sensing Matrix Design

In previous work [16], we presented a frame of work characterizing the matrix form. We showed that using a dense Boolean matrix is prohibitive and leads to a saturation of the observations, in which case it becomes impossible to recover the observed signal. To overcome this problem, we showed that we need a new criterion making a connection between the weight of each line in the sensing matrix and the quality of caught information.

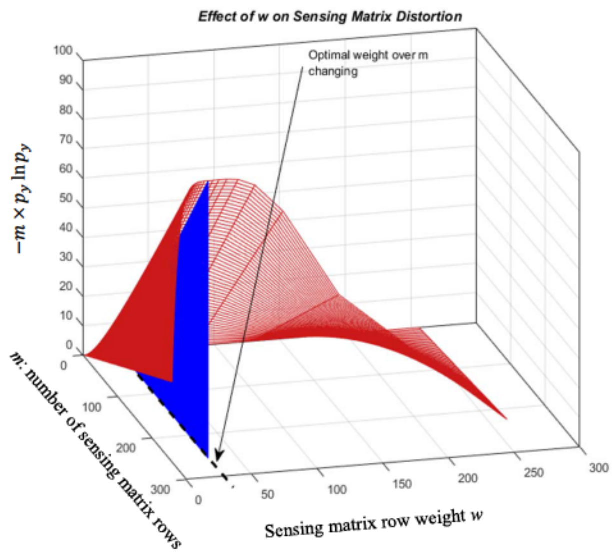

Fig. 2. Evolution of: $-m \times p_y \ln p_y$ for different value of m. The optimal weight w of each lines of the matrix is independent of m varying. number of inputs: n=256.

Let consider p_y denoting the probability of one measurement y_j to be equal to 1 *(i.e. at least one nonzero* $x_i \neq 0$ *)* is connected to y_j and p_x the probability that $x_i = 1$. The relationship between the couple (p_y, p_x) is given by :

$$p_y = \binom{w}{i} p^i{}_{\mathrm{x}} (1 - p_{\mathrm{x}})^{w-i}, \qquad (4)$$

where w is called weight row of $\mathbf{A}$ and i is the value of observation y. In our bipartite model, random inputs generate random observations. Although only degrees of measurements are used in the evolution, all non-zero measurements are needed to recover the value of every element in x. Thus, the rate of measurements is calculated by the relation between the number of rows m and the weight row w of $\mathbf{A}$.The general form of rate-distortion of measurements is given by [10]:

$$R(w,m) = -m\sum_{w=0}^{n} p_y \ln(p_y), \tag{5}$$

For a given fixed rate $R(w,m)$, a good compressive system, i.e. minimizing m, implies maximizing $p_y \ln p_y$ with respect to w. Fortunately, in the Boolean problem y has a unique nonzero possibility, i.e. i=1.

$$\frac{\partial\, p_y \ln p_y}{\partial w} = \frac{\partial}{\partial w}[wp_x(1-p_x)^{w-1}.\ln(wp_x(1-p_x)^{w-1})] = 0 \tag{6}$$

After some manipulations and simplifications, the value of w that maximize $p_y \ln p_y$ is approximated by:

$$w \approx \frac{1}{p_x} \tag{7}$$

Notice that for $p_x = k/n$, $w \approx n/k$.

C. Boolean CS Arithmetic reformulation

Let us consider an arithmetical reformulation of (1), note that the resulting of logical operations can be summarized by: $y_i = \begin{cases}1 & if\ \exists j \mid A_{ij} \wedge x_j \neq 0 \\ 0 & otherwise\end{cases}$. Since $A_{ij} \geq 0$ and $x_j \geq 0\ \forall\ i$, $j \in (1,n)$, $(1,m)$, respectively, we can state that $A_i \wedge \mathrm{x} \Leftrightarrow \Gamma(\langle A_i, \mathrm{x}\rangle)$, such that $\langle a,b\rangle$ defines the dot product of two vectors a and b, and $\Gamma(c) = \{1\ if\ c \neq 0\}$ a one-bit mapping operator. For the noise corruption case, the process is defined by:

$$\mathbf{y}_i = \Gamma(\langle A_i, \mathbf{x}\rangle + \mathrm{n}_i), \tag{8}$$

where $n \sim \mathcal{N}(0,\sigma^2)$ a Gaussian noise with zero mean and σ^2 standard deviation. Encoding signal by (8) is hardware friendly. The above scheme needs very limited material resources and does not require a complex technology to be realized.

Obviously, we are looking for a Boolean sparse vector with minimum non-zero elements that verify the equality $\mathrm{y}-\Gamma(\mathbf{Ax}) = 0$. So, we can define our optimization problem by:

$$\hat{\mathbf{x}}\, \underset{\mathbf{x}}{argmin}\|\mathbf{x}\|_0 \quad \text{s.t. } \mathbf{y} - \Gamma(\mathbf{Ax}) = 0,\ \mathbf{x} \in \mathbb{B}^n, \tag{9}$$

where$\|\,.\,\|_0$ refers to the ℓ_0 pseudo-norm that expresses the cardinal of a vector (number of non-zero elements of a vector).

The problem as described in (9) has non- straightforward solution due to the complexity of optimizing the ℓ_0 pseudonorm. Notice that for a Boolean signal, $x_i \in (0,1)$:

$$min\sum_i (x_i \neq 0) \Leftrightarrow min\sum_i |x_i| \tag{10}$$

Consequently, we enforce sparsity of $\mathbf{x}$ by minimizing $\ell_1 = min\|\mathbf{x}\|_1$ instead of ℓ_0 which has a combinatorial solution of dimension n.

In the other side, the gradient of the $\Gamma(\mathbf{A}.\mathbf{x})$ function is not continuous and cannot be directly used during the optimization process. Inspired from neural networks techniques, we propose to use logistic functions $\mathrm{L}(\vartheta) = \frac{1}{1+e^{-\vartheta}}$ to mimic the behaviour of results of our logical gates $(\wedge, \vee)$ combination. This model does not match exactly our system, and the occurring error can be viewed as a noisy smalls positives component that we try to minimize at all. From this point of view, our optimization problem (5) is turned in an equivalent form of:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{argmin}\|\mathbf{x}\|_1 \quad \text{s.t. } \frac{1}{2}\|r(\mathbf{x})\|_2^2 \leq \varepsilon,\ \ \mathbf{x} \in \mathbb{B}^n, \tag{11}$$

where $r(\mathbf{x}) = \mathbf{y} - \mathrm{L}(\mathbf{Ax})$ is said residual function.

III. The Approximate Trust Region Algorithm

The main contribution of this paper is to propose a new reconstruction algorithm for the BCS problem (11). We choose to explore an approximate smoothing variant of the Trust-Region Methods (TRM).

A. Trust-Region Methods Principle

Given the unconstrained nonlinear programming problem:

$$\min_{x \in \mathbb{R}^n} f(x) \tag{12}$$

At a current point $x^k \in \mathbb{R}^n$, we consider that a quadratic approximation $q_k(\mathbf{s})$ of $f(x^k)$ exists around an increment $s = x^{k+1} - x^k$, such that:

$$q_k(s) \triangleq \nabla f(x^k)^T s + 1/2\, s^T \nabla^2 f(x^k) s, \tag{13}$$

where $\nabla f(x^k)$ *and* $\nabla^2 f(x^k)$ express respectively, the gradient and Hessian of the function $f(x)$ evaluated at x^k. In this case:

$$f(x^k) \simeq f(x^{k-1}) + q_{k-1}(s) \tag{14}$$

Trust region methods describes an iterative update of the step $s = x^k - x^{k-1}$, where $s \approx 0$ when solution is nearly attained.

Our approximation $q_k(s)$ should be minimized over restricted total variation of all steps. We define:

$$(TR)^k:\ \min_{\|s\|<\Delta_k} q_k(s) \tag{15}$$

where Δ_k defines the radius of the *trust region* about x^k in which the quadratic approximation $q(s)$ represents accurately $f(x)$. Now we propose to turn the problem in another form; instead of solving problem (15) which is quite difficult to solve, we propose to solve :

$$(TR)^k\colon \min\left(q_k(s) + \frac{1}{2\delta_k}\langle s,s\rangle\right), \qquad (16)$$

where vector of adaptive steps δ_k plays the same role than Δ_k. By choosing properly δ_k we can prove that our proposal converges in the same way than conventional trust region algorithm.

B. Gauss-Newton Approximate

Problem (11) can be linked to the Gauss-Newton form of quadratic approximation in *(TR)* [19], where the minimizing function has ℓ_2 norm of a residual function $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$. We call this a nonlinear least square problem. In this case we don't need to compute the Hessian; the second derivative of r. We can show that only the Jacobian: $J_s(x^k) = r'(x^k)$ is needed, and $\nabla_s r(x^k) = r(x^k)^T J_s(x^k)$. In this particular case, the quadratic approximation of r is given by [19]:

$$q_k(s) \triangleq r(x^k)^T J_s(x^k)s + \frac{1}{2}s^T B(x^k)s, \qquad (17)$$

where $B(x^k) = J_s(x^k)^T J_s(x^k)$. Moreover, if we consider that each residual component at actual point is a smoothing representation of the function $r(x^k)$, then $B(x^k) = J_s(r)^T J_s(r) + \eta I$, with η a smoothing parameter [18].

C. Proposed Solution

Let us construct the problem to solve now, first notice that problem (11) has an unconstrained form using a relaxation method of:

$$(\mathbf{x}) = \|\mathbf{x}\|_1 + \frac{\mu}{2}\|r(\mathbf{x})\|_2^2 \quad s.t. \;\; \mathbf{x} \in \mathbb{B}^n \qquad (18)$$

Problem (18) is often easiest to solve than (11), and its second order Taylor expansion in the vicinity of the solution is given by:

$$m_s(\mathbf{x}^k) = f_s(\mathbf{x}^k) + \mu . q_k(s) \qquad (19)$$

Solving this problem using our approximate TR (16) leads to:

$$(TR)^k\colon \min\left(\|\mathbf{x}^k\|_1 + \frac{\mu}{2}\|r(\mathbf{x}^k)\|_2^2 + \mu q_k(s) + \frac{1}{2\delta_k}\|s\|_2^2\right) \qquad (20)$$

After initiating an estimate $\mathbf{x}^0$ and δ_0 step size, at iteration *k*, we compute $p_s(\mathbf{x}^k)$, and we compare the real reduction from previous iteration $\alpha_{red} = f(\mathbf{x}^k) - f(\mathbf{x}^{k-1})$ to quadratic approximate reduction obtained also from previous iteration $\beta_{red} = m(\mathbf{x}^k) - m(\mathbf{x}^{k-1})$. Given the ratio: $\gamma_s = \frac{\alpha_{red}}{\beta_{red}}$, at the end of each iteration, we decide on whether δ_k is increased or decreased depending on how well γ_s expresses the similarity.

Algorithm: Approximate Trust Region

1. Define $\mu, 0 < \eta_1 < \eta_2$ and $0 < w_{down} < 1 < w_{up}$
2. Given an initial point $\mathbf{x}^0$, $step\ size\ \delta^0 > 0, k = 0$
3. New trial : compute $\mathbf{x}^k$ from (16)
4. Compute γ_s by :
$$\gamma_s = \frac{f(\mathbf{x}^k) - f(\mathbf{x}^{k-1})}{\beta_{red}}$$
5. Accept or reject $\mathbf{x}^k$, $f(\mathbf{x}^k)$, r^k, $(r')^k$
If $\gamma_s > \eta_1$ then accept
6. else reject
7. Update δ^k
If $\gamma \geq \eta_2$ then $\delta^k = min(w_{up}\delta^{k-1}, \delta_{max})$
Else $\gamma_s \leq \eta_1$ then $\delta^k = w_{down}\delta^{k-1}$
8. If converged then stop, else $k = k + 1$ go to 3.

Hence, we compute the new trial by shrinking to the positive quadrant an estimation of a curvilinear gradient descent in the proximity of previous trial.

$$\mathbf{x}^k = \Pi\big(0, \mathbf{x}^{k-1} - \delta^k(\mu[B(x^{k-1})]^{-1} r(x^{k-1})^T J_s(x^{k-1}) + +1)\big), \qquad (21)$$

where Π expresses nonlinear function max(.). Equation (21) comes from the first order optimality condition of (15):

$$\frac{\partial m_s(\mathbf{x}^k)}{\partial \mathbf{s}}\Big|_{\mathbf{s}=\mathbf{s}^*} = 0 \qquad (22)$$

D. Discussion

The similarity between the reduction of function *f* and its approximation, is measured by the ratio γ_s and

$$\beta_{red} = \|\mathbf{x}^k\|_1 - \|\mathbf{x}^{k-1}\|_1 - \mu r(\mathbf{x}^k)^T J_s(\mathbf{x}^k)(\mathbf{x}^{k-1} - \mathbf{x}^k) - \frac{\mu}{2}\|J_s(\mathbf{x}^k)(\mathbf{x}^{k-1} - \mathbf{x}^k)\|_2^2 \qquad (23)$$

We compare γ_s to η_1 and η_2 such $0 < \eta_1 < 1 < \eta_2$, we accept each new value of $\mathbf{x}^k$, r^k, $(r')^k$ and $f(\mathbf{x}^k)$ as the new values of the present trial if $\gamma_s > \eta_1$, otherwise, we set the previous values unchanged.

Moreover if $\gamma_s > \eta_2$ which means that there is no similarity between the decrease in the real function at the trial point and its approximate reduction than we expand the search region by $\delta^k = w_{up}\delta^{k-1}$, otherwise we decrease the step-size $\delta^{k+1} = w_{down}\delta^k$ assuming that $0 < w_{down} < 1 < w_{up}$.

The pseudocode shown in Algorithm window, resumes the proposed Approximate Trust Region solution for the Boolean Compressive Sensing problem (ATR-BCS).

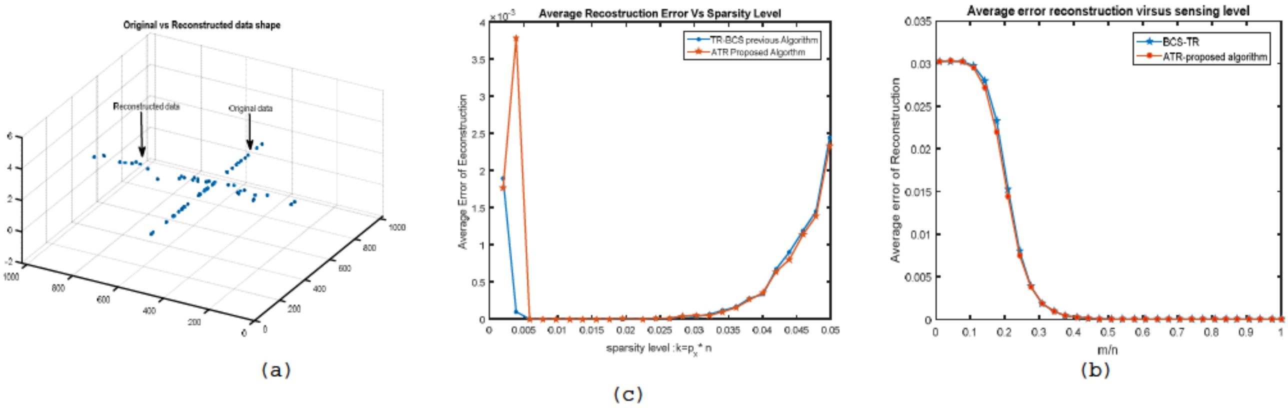

Fig. 3. Algorithm comparison with different measures for proposed algorithm ATR-BCS vs previous based TR-BCS procedure [17] : (a) Reconstruction example, (b) Average Reconstruction Error vs sparsity level and (c) Average Reconstruction Error vs sensing level.

IV. Simulation Results and Discussion

We perform different experiments to evaluate the proposed algorithm compared to previous works [17]. TR-BCS which uses both Augmented Lagrangian formulation and Trust Region solution was created specially to deal with BCS and is conceptually based on the same approach. However, in the new algorithm instead of modeling our problem by an Augmented Lagrangian approach before solving the unconstrained form by Trust Region method, we consider that we face an unconstrained nonlinear least square problem with μ penalty that we solve directly by approximate TRM. In this manner we reduce considerably the number of iterations

The setup of the experiments is as follows: firstly, we create a Boolean k-sparse signal $\mathbf{x} \in \mathbb{B}^n$ with $k = p_x \times n$ non-zero elements. The positions of its non-zero entries are drawn from standard uniform distribution. The sensing matrix $\mathbf{A} \in \mathbb{B}^{m\times n}$ is created by selecting randomly $w \approx 1/p_x$ positions at the first row and fill them with ones. After that, we generate a right shift of the first row to generate m-1 remained rows. $\mathbf{y} \in \{0,1\}^m$ is computed by (1). Once iterations ended, we decide if zero or one is affected to the signal components using a decision threshold value of 1 for both algorithms. The algorithms parameters are set to: $\eta_1 = 0.1$, $\eta_2 = 0.75$, $w_{down} = 0.25$, $w_{up} = 1.25$ and $\mu^0 = 40$, $\lambda^0 = 1$.

A. *Level of Compression $\sigma = m/n$*

We set n =1024 and $p_x = 3\%$, a typical value of a sparsity level and we change the sensing level σ in the range of (0, 1] with step size of 4%, generating a total of 25 ascending values. For each value, we perform 300 trials.

B. *Robustness to Sparsity Level (k) Variation*

In the second experiment, we set n = 1024, $\sigma = \frac{m}{n} = 0.5$ and we vary the level of sparsity k from 0.1% to 5% with step size of 0.02%, finally we obtain 25 values, for each value a trend of 300 trials is generated and averaged.

The results are depicted in Fig. 3. We use the Average Reconstruction Error for comparison:

$$Rec_{err(\mathbf{x})} = \frac{1}{n}\left[\sum_{i=1}^{n} \mathrm{xor}(\mathbf{x},\hat{\mathbf{x}}) \neq 0\right], \qquad (24)$$

to evaluate reconstruction accuracy for both compression and sparsity level changes. From Fig. 3, we observe that both algorithms show the same trend as σ increases or k, both algorithms give comparable performances with little outperforming for proposed method, ATR-BCS obtains a little bit smaller reconstruction error Rec_{avg} than previous (TR-BCS) algorithm.

C. *Convergence comparison*

Since the performances of both algorithms are comparable, the outperforming of the proposed algorithm can be shown during convergence process where number of iterations is crucial to obtain a global solution. ALM-TR is based on a nested technique where we solve the internal problem followed by a new trial for external AL problem and so on until convergence reached, while ATR is based only on a unique procedure with one level of iterations.

We propose now to compare the convergence speed of the cost functions for both algorithm. In fact, it appears from the previous simulations that the two algorithms are equivalent in term of quality of reconstruction. The difference resides on their efficiency: number of iterations needed to reach stationary point.

We set n = 1024 and $p_x = 3\%$, and the sensing level $\sigma = \frac{m}{n} = 1/2$ we store 500 trials of residual error :

$$\|r(\hat{\mathbf{x}})\|_2^2 \quad | \quad r(\mathbf{x}) = \mathbf{y} - \mathrm{L}(\mathbf{A}\mathbf{x}) \qquad (25)$$

Fig. 4. depicts evolution of the averaged residual error for both algorithm over the number of iterations.

An interesting observation can be made, we notice that the new algorithm suffers of a holding up until the iteration 15, while TR-BCS began its convergence at the iteration 10. For instance, we have no explanation why such delay appears. However, we suppose that it is due to the growing area of search linked to parameter δ_k until reaching sufficient surface that includes a possible line of search. Fig. 4. shows us that the proposed algorithm has less residual error and so, it can be viewed as a substitute of a previous one with better performance and less complexity.

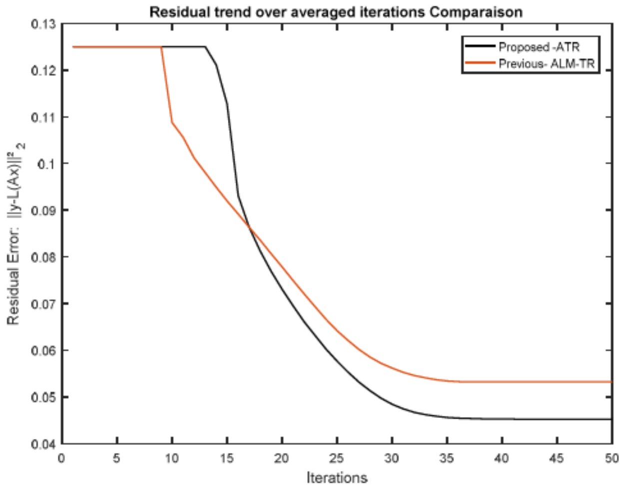

Fig. 4. Residual Function Evaluation for both algorithm

V. CONCLUSION

By converting the BCS problem in an unconstrained form using a penalty parameter and considering a residual expression of the distance between observations and their estimates, we could present a direct solution to the BCS problem using an approximate trust region algorithm without need to Augmented Lagrangian solvers procedure in contrast with previous work. Experiments show us that both algorithms present similar performance and accuracy. However, the number of iterations is considerably reduced, and the cost of relative complexity is also reduced. Next step will be to consider the effect of relaxing the Boolean constraint replaced here by a threshold. Obviously, we will present a framework considering fight of noise corruption effect on observations by coding information via sensing matrix.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their careful reading of our manuscript and their insightful comments and remarks. We are providing a revised version in the final paper that we hope will reflect their suggestions and comments. We feel that this has resulted in a stronger clarity of our contribution.

The authors also thank the National Sciences and Engineering Research Council of Canada (NSERC) for their financial support.

REFERENCES

[1] Y. C. Eldar and G. Kutyniok, Compressed Sensing: Theory and Applications, Cambridge Univ. Press, 2012.

[2] Z. Han, H. Li, and W. Yin, "Compressive Sensing for Wireless Networks" , Cambridge Univ. Press, 2013.

[3] K. Hayashi, M. Nagahara, and T. Tanaka, "A user's guide to compressed sensing for communications systems," IEICE Trans. Commun., vol. E96-B, no. 3, pp. 685-712, March 2013.

[4] R. Dorfman, "The detection of defective members of large populations," Annals of Mathematical Statistics, vol. 14, no. 4, pp. 436–440, 1943.

[5] M.B. Malyutov, "On planning of screening experiments," in Proceedings of 1975 *IEEE-USSR Workshop on Inform. Theory* , 1976, pp. 144–147

[6] Y. Erlich, A. Gordon, M. Brand, G. J. Hannon and P. P. Mitra, "Compressed Genotyping," in *IEEE Transactions on Information Theory*, vol. 56, no. 2, pp. 706-723, Feb. 2010.

[7] CUI, Hao, LUO, Chong, WU, Jun, *et al.* Compressive coded modulation for seamless rate adaptation. *IEEE Transactions on Wireless Communications*, 2013, vol. 12, no 10, p. 4892-4904.

[8] M. Wang, J. Wu, S. F. Shi, C. Luo and F. Wu, "Fast Decoding and Hardware Design for Binary-Input Compressive Sensing," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 591-603, Sept. 2012.

[9] G. K. Atia and V. Saligrama, "Boolean Compressed Sensing and Noisy Group Testing," in *IEEE Transactions on Information Theory*, vol. 58, no. 3, pp. 1880-1901, March 2012.

[10] M.B. Malyutov, "Recovery of sparse active inputs in general systems: A review," in Int. Conf. on Comp. Tech. in Electrical and Electronics Eng., Region 8, SIBIRCON, 2010, pp. 15 – 22

[11] F. Wu, J. Fu, Z. Lin, and B. Zeng, "Analysis on rate-distortion performance of compressive sensing for binary sparse source," 2009 Data Compression Conference, (Washington, DC, USA), pp. 113–122, IEEE Computer Society, 2009.

[12] D. Malioutov and M. Malyutov, "Boolean compressed sensing: LP relaxation for group testing," *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, 2012, pp. 3305-3308.

[13] A. R. Conn, N. I. M. Gould, and P. L. Toint, "Trust-region methods", ser. MPS/SIAM Series on Optimization. Philadelphia, PA: *Society for Industrial and Applied Mathematics* (SIAM), 2000.

[14] J. Nocedal and S. J. Wright, "Numerical Optimization", 2nd ed., ser. Springer Ser. in Operations Res. and Financial Eng. Springer, 2006.

[15] J. B. Erway, R. J. Plemmons, L. Adhikari and R. F. Marcia, "Trust-region methods for nonconvex sparse recovery optimization," *2016 International Symposium on Information Theory and Its Applications (ISITA)*, Monterey, CA, 2016, pp. 275-279.

[16] S. Hachemi and D. Massicotte, "Binary input-output compressive sensing: A sub-gradient reconstruction," 2015 *IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Halifax, NS, pp.565-570, 2015.

[17] S. Hachemi and D. Massicotte "Recovery of Boolean Sparse-Signals with Unknown Sparsity Level", 2017, *IEEE European Signal Processing Conference (EUSIPCO'2017)*, Kos Island, Greece, 28'Aug.-02Sept.

[18] Chen, X., Du, S. & Zhou, Y. Sci. China Math. (2016) 59: 999. https://doi.org/10.1007/s11425-015-5116-z

[19] Mokhtar S. Bazaraa, Hanif D. Sherali 'Nonlinear programming : theory and algorithms', C. M. Shetty.-3rd ed. "Wiley-Interscience." Includes bibliographical references and index. ISBN-13: 978-0-471-48600-8

[20] J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, and C.A. Sagastizábal, "Numerical Optimization: Theoretical and Practical Aspects", Springer, Third edition in 2015.