A Null-Space-Based Genetic Algorithm for Constrained l_1 Minimization

Miguel Heredia Conde, Dunja Hage, Otmar Loffeld
Center for Sensorsystems (ZESS)
University of Siegen, Paul-Bonatz-Straße 9-11
57076 Siegen, Germany
E-mail: {heredia, hage, loffeld}@zess.uni-siegen.de

***Abstract*—Compressive sensing (CS) theory enables linking the sensing effort, that is, the volume of data that a sensor produces, to the amount of *information* this data conveys, rather than to the desired bandwidth, as prescribed by classical sampling theory. As a consequence, in the typical CS scenario, one ends up with a set of m measurements and the objective is to recover a signal whose dimensionality is $n \gg m$, typically under the assumption of being *sparse*. Recovering the sparsest solution that satisfies the measurements is an NP-hard problem and a common workaround is relaxing it to a linearly-constrained l_1 minimization. In this work we introduce a novel algorithm for solving this problem that exhibits the structure of a genetic algorithm, but fully operates in null-space domain. This allows reducing the dimensionality of the chromosomes to the minimum, i. e., $n-m$. *Crossover* follows a deterministic scheme, with adjustable number of parents and children per pairing. Furthermore, mutations are not random, but guided along the direction of the negative gradient of the *fitness* function. Numerical simulation revealed that the proposed algorithm performs better than comparable alternatives in terms of reconstruction error when the number of iterations is to be kept very low. This, together with its high parallelization potential, paves the way for faster CS reconstruction.**

I. Introduction

The Shannon sampling theorem is an undeniable cornerstone of modern signal acquisition and communication systems. Unfortunately, tying the bandwidth of a sampling system to the bandwidth of the signal being sensed translates into large volumes of data per time unit when the desired bandwidth is large. The most common countermeasure for bounding the transmission and storage requirements is performing some signal compression right after sensing, but the sensing effort remains directly linked to the bandwidth. The groundbreaking theory of *compressive* (or *compressed*) sensing (CS), has shown that most real signals can, in fact, be recovered from a reduced number of measurements, often much lower than that suggested by Shannon, effectively reducing the sensing effort. Unfortunately, recovering a discrete signal from a number of measurements that is lower than its dimensionality is an underdetermined problem. In order to recover the signal one needs to exploit the bounded information constraint, which translates into solving an l_0 minimization of the signal representation in some basis where the signal can be sparsely represented. In CS the measurements are linear and, therefore, reconstructing the signal means solving a linearly-constrained l_0 minimization. Provided that this is known to be an NP-hard problem, it is common to solve a linearly-constrained l_1 minimization instead, which is the convexification of the previous.

In this work we focus on solving this problem operating in the *null space* of the sensing matrix, which models the linear constraints. Obviously, given any initial solution to the underdetermined system of linear equations, the addition of any vector living in the null space of the sensing matrix also yields a solution. In other words, given an initial solution, the challenge is to find a null space vector that added to it yields the minimal l_1 norm. Existing CS recovery methods can be roughly classified between *greedy* algorithms, which sequentially build the sparse signal support, and methods that solve the constrained l_1 minimization instead. The first ones are able to ensure that the reconstructed signal is actually sparse, but the reprojection error may be different from zero. Methods of the second type are typically optimization schemes, in which the cost function is a (weighted) sum of two terms: one for the reprojection error and the other for l_1 regularization. Methods operating in the null space first fix the subspace where the feasible solutions live and then look for the one with minimal l_1 norm, thus always fulfilling the measurements.

Recently, Kalman filters have been suggested as an alternative algorithm architecture for CS signal recovery. The core idea of these approaches is integrating the norm to be minimized, e. g., the l_1 norm, as a pseudomeasurement within the filter. Of special interest is [1], since it proposed for the fist time that the filter operates in null space domain, thus reducing the dimensionality of the state vector to the actual number of remaining degrees of freedom, that is, to the minimum. In this work we adopt the idea of operating exclusively in null space domain, but we do not restrict the algorithm to a single state vector, but to a *population* of feasible solutions. Inspired by the concept of natural adaptation present in biological systems, we make the population evolve by combining *crossover* of the fittest individuals with *mutation* of the offspring, yielding a genetic algorithm (GA). Differently from GAs, mutation is not fully random, but mimics a (restricted) *gradient descent* iteration, which accelerates convergence. The proposed GA outperforms of state-of-the-art CS recovery algorithms for low number of iterations in terms of reconstruction error, while exhibiting much higher parallelization potential.

978-1-5386-4662-5/17/$31.00 ©2017 IEEE

II. The Compressive Sensing Scenario

The mathematical theory of compressive sensing (CS) [2], [3] states that real-world signals can often be recovered from much fewer measurements than those suggested by the Shannon sampling theorem. The Shannon sampling theorem links the sampling rate that is required for acquiring a continuous signal to the maximum frequency contained in it. Rather than an upper bounded on the frequency, CS theory requires the signal to be *sparse* in some basis or tight frame. If this holds and some additional requirements regarding the sensing scheme are satisfied, then the signal can be exactly recovered from few non-adaptive measurements. Let $\vec{x} \in \mathbb{C}^n$ be the discrete signal we want to recover, in its sparse representation. The l_0 norm of $\vec{x}$ is defined as:

$$\|\vec{x}\|_0 \coloneqq \lim_{p \to 0} \|\vec{x}\|_p^p = |\operatorname{supp}(\vec{x})| \tag{1}$$

that is, the cardinality of the support of $\vec{x}$, and $\vec{x}$ is called an s-sparse signal if $\|\vec{x}\|_0 \leq s$. Provided that the sparsity requirement is satisfied, the challenge is to reconstruct $\vec{x}$ from $m \ll n$ linear measurements. Thus, the measurement model is described by the underdetermined linear system:

$$\vec{y} = \boldsymbol{A}\vec{x} \tag{2}$$

where $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ is the *measurement matrix*, which explains how the vector of measurements $\vec{y} \in \mathbb{C}^m$ relates to $\vec{x}$ and may be the composition of the actual *sensing matrix*, and a dictionary. Unfortunately, the problem we would like to solve, that is, finding the sparsest $\vec{x}$ satisfying Eq. 2, is known to be NP-hard. A common workaround is convexifying the problem turning the l_0 minimization into l_1. Given any particular solution to Eq. 2, $\vec{x}_0$, finding the constrained l_1 norm minimizer is equivalent to solving:

$$\hat{\vec{x}} = \underset{\vec{x} \in \{\mathcal{N}(\boldsymbol{A}) + \vec{x}_0\}}{\arg\min} \|\vec{x}\|_1 \text{ subject to } \vec{y} = \boldsymbol{A}\vec{x} \tag{3}$$

where $\mathcal{N}(\boldsymbol{A})$ denotes *null space* of $\boldsymbol{A}$.

III. Related Work

Primal-dual interior point methods have shown to be efficient tools for solving Eq. 3. Provided that the problem in Eq. 3 can be directly expressed as a linear program, the classical Newton method can be used to approach a solution, as in the l_1-magic library [4]. The Chambolle and Pock's (C&P) primal-dual algorithm [5], which is often referred as the fastest method for solving Eq. 3, is a first-order primal-dual method for convex optimization problems with saddle-point structure and convergence rate $\mathcal{O}(1/n)$.

Using a Kalman filter for estimating sequences of sparse signals from a reduced set of *compressed* measurements was initially proposed in [6]. In this work, once the signal support is estimated, a Kalman filter with reduced-dimensionality state vector runs in time domain until the Kalman innovation gets too large. Then the support needs to be estimated again. Both support element addition and removal are based on thresholding, thus subjugating the algorithm's performance to threshold tweaking. The same idea of supposing negligible signal support changes between consecutive signals is adopted by in [7]. Supposing constant support, a solution and its corresponding residual are obtained. Then further addition of support elements is considered by performing CS reconstruction using the residual as vector of measurements. As before, both addition and deletion of support elements imply thresholding. A reduced-dimensionality Kalman filter is suggested as superstructure to improve the estimation of the vector coefficients indexed by the support set, supposing that the system model is known. Note that both in [6] and [7] the sparse recovery algorithm is uncoupled from the probabilistic filter. Differently, in [8] the authors propose including the norm to minimize, typically the l_1 norm, but also *quasi-norms* $l_p,\ 0 \leq p < 1$, as an additional measurement of the filter. This way, the filter can perform the signal reconstruction by itself, while dynamically incorporating new compressed measurements as they arrive.

Independently from the aforementioned works, a Kalman filter for solving Eq. 3 was proposed in [1] an further studied in [9], [10]. The l_1 norm of the temporal solution is the only measurement of the filter, which has the peculiarity of operating in $\mathcal{N}(\boldsymbol{A})$. At each iteration the algorithm tries to push down the l_1 norm of the temporal solution by incorporating a measurement that is slightly lower than the actual l_1 norm. A major issue of such an algorithm is defining an optimal sequence of step sizes across iterations. Using a simple scaling of the l_1 norm by some factor $0 < \gamma < 1$, as in [1], is not efficient. In our reimplementation we use an Aitken-based convergence acceleration for automatically adjusting the synthetic l_1 shrinkage.

Regarding the use of GAs for sparse signal recovery in CS, most works used the GA for optimizing parameters of the CS framework, be the number of measurements, i. e., number of rows of the sensing matrix, as in [11]–[13], be specific parameters of the reconstruction algorithm, as in [14]. In these works the GA is completely uncoupled from the signal recovery algorithm. More interesting are the works where the GA is directly used as sparse recovery algorithm. In [15], [16] the nondominated sorting GA named NSGA-II [17] is used as multiobjective GA. The fitness function is composed by two terms, namely, the l_0 norm of the solution and its reprojection error. In [18] a GA combined with bacterial foraging optimization (BFO) is used to find the global optimum to multimodal optimization problems such as CS sparse reconstruction, but no details are provided on how the multimodality of the fitness function is handled. The *hybrid* GA of [19] includes a step of modified parallel coordinate descent (PCD) to prevent the algorithm from getting stuck. The algorithm solves the sparse reconstruction problem for known s by means of an explicit s-thresholding after crossover. We suppose that there is no *a priori* knowledge on s, nor on the amplitudes of the non-zero components.

IV. A GA for Constrained l_1 Minimization

In the following we present our GA for solving the constrained l_1 minimization introduced in Eq. 3. Our approach shares with GAs the typical sequential structure of *mutation*, selection of the fittest individuals for breeding, according to some *fitness function*, and *crossover* to generate new individuals for the next generation [20]. A block diagram of the algorithm is provided in Fig. 1. Despite the problem being solved and the idea behind the solving scheme is different, the sequence of genetic operations has been taken from [21] without major modifications. The population individuals or *chromosomes* have $n-m$ genes instead of n. If each gene can adopt b different values or *alleles*, then there exist b^{n-m} different possible genetic structures or *genotypes*. Each chromosome of the population embodies the coefficients of a null space vector in a given null space basis. Recall that $\mathcal{N}(\boldsymbol{A})$ is an $(n-m)$-dimensional subspace of $\mathbb{C}^n$. The number of alleles b corresponds to the number of bits used for quantization. The i^{th} chromosome at generation k^{th} is denoted $\vec{n}_i^{(k)}$.

A. Initial Population

The chromosomes of the initial population $\mathfrak{P}^{(0)}$ are randomly generated. More specifically, we draw the real and imaginary values of each of the genes from *i.i.d.* normal distributions of zero mean and unit variance. For any arbitrary population at generation k, the associated temporal solutions are given by:

$$\vec{x}_i^{(k)} = \vec{x}_0 + \boldsymbol{E}_{\mathcal{N}(\boldsymbol{A})}\vec{n}_i^{(k)} \tag{4}$$

where $\boldsymbol{E}_{\mathcal{N}(\boldsymbol{A})} \in \mathbb{C}^{(n-m)\times n}$ denotes the basis of $\mathcal{N}(\boldsymbol{A})$.

B. Mutation

The process of *mutation* applied over some population $\mathfrak{P}^{(k)}$ means that the chromosomes undergo stochastic changes in the values of their genes. We define a single mutation probability $p_{\text{mut gen}}$, which is the probability that any gene of any chromosome undergoes mutation. Typically, when a gene mutates, the mutated allele (feasible value) is also randomly chosen from the b^{n-m} possibilities. Unfortunately, the GA becomes unacceptably slow for values of b that are not too small, such as those typically used for quantization in conventional computers. In other words, the manifold of directions in which each chromosome can evolve via mutation becomes too large to be explored in a random fashion, unless very large populations are used.

In order to overcome this issue, we propose applying mutations that are not fully random, but partially seek maximizing the decrease of the cost function. To this end our mutation process is divided in two sequential steps. First, for each chromosome in the population the set of genes that will undergo mutation is defined by means of realizations of *i.i.d.* Bernoulli distributions of $p_{\text{mut gen}}$ success probability. For the chromosome i at generation k the set of gene indices to undergo mutation is denoted $M_i^{(k)}$. Then, for each $\vec{n}_i^{(k)}$ with corresponding gene mutation set $M_i^{(k)}$, the gradient of the *fitness function* restricted to $M_i^{(k)}$ is calculated. The second step is using the $M_i^{(k)}$-restricted gradient to mutate the chromosome $\vec{n}_i^{(k)}$ by means of a (restricted) *gradient descent* step, namely,

$$\begin{cases} \left(\vec{n}_i^{(k)\prime}\right)_{\bar{M}_i^{(k)}} = \left(\vec{n}_i^{(k)}\right)_{\bar{M}_i^{(k)}} \\ \left(\vec{n}_i^{(k)\prime}\right)_{M_i^{(k)}} = \left(\vec{n}_i^{(k)}\right)_{M_i^{(k)}} + \tau_i^{(k)} \nabla_{M_i^{(k)}} f\left(\vec{n}_i^{(k)}\right) \end{cases} \tag{5}$$

where the prime is used to denote the same chromosome after mutation, $(\cdot)_M$ denotes restriction to the dimensions indexed in the set M, and ∇_M denotes the M-restricted gradient, different for each chromosome at each generation. At each iteration the step size $\tau_i^{(k)}$ is calculated by means of a backtracking line search. This controlled mutation scheme allows for using larger $p_{\text{mut gen}}$, speeding the GA at no risk of divergence. Mutation tends to a pure per-chromosome gradient descent iteration as $p_{\text{mut gen}} \to 1$.

C. The Fitness Function

The so-called *fitness function* evaluates how *fit* a population individual or chromosome is according to some evaluation criterion. In other words, the fitness function is the function for which we seek an optimizer. Our fitness function is the l_1 norm of the temporal solution associated to $\vec{n}_i^{(k)}$ and given by Eq. 4, that is,

$$f\left(\vec{n}_i^{(k)}\right) = \left\| \vec{x}_0 + \boldsymbol{E}_{\mathcal{N}(\boldsymbol{A})}\vec{n}_i^{(k)} \right\|_1 . \tag{6}$$

Note that, differently from prior work on GAs for constrained sparse recovery (see related work in [21]), we do not need a reprojection error term, since we keep operating within $\mathcal{N}(\boldsymbol{A})$ and all temporal solutions necessarily satisfy Eq. 2.

D. The Selection Scheme

After evaluating the fitness of the individuals using Eq. 6, groups are formed according to the obtained values. In our case, the *fittest* individuals are those chromosomes with the lowest function values. In short terms, only the fittest ones should take an active role in producing the next generation of the population. We adopt the selection scheme presented in [21], which contemplates both a set of parents, which will breed the next generation, and an *elite*, which is transferred to the next generation unchanged. As in [21], the elite is denoted $\mathfrak{P}_{\text{elite}}^{(k)}$ at generation k and is of size $\left|\mathfrak{P}_{\text{elite}}^{(k)}\right| = n_{\text{elite}} = k_{\text{elite}} n_{\text{ind}}$. The set of parents is denoted $\mathfrak{P}_{\text{breed}}^{(k)}$, with size $\left|\mathfrak{P}_{\text{breed}}^{(k)}\right| = n_{\text{breed}} = k_{\text{breed}} n_{\text{ind}}$, with $0 < k_{\text{elite}} \ll k_{\text{breed}} < 1$. Clearly, $\mathfrak{P}_{\text{elite}}^{(k)} \subset \mathfrak{P}_{\text{breed}}^{(k)} \subset \mathfrak{P}^{(k)}$. In order to prevent the population size from shrinking, a third subset of the population may become necessary: the *contingency population*, $\mathfrak{P}_{\text{cont}}^{(k)} \subset \mathfrak{P}^{(k)} \setminus \mathfrak{P}_{\text{elite}}^{(k)}$.

E. Crossover

In the context of GAs, *crossover* refers to the process of generating a population of children, $\mathfrak{P}_{\text{child}}^{(k)}$, of size $\left|\mathfrak{P}_{\text{child}}^{(k)}\right| = n_{\text{child}}$ from a previously-selected breeding

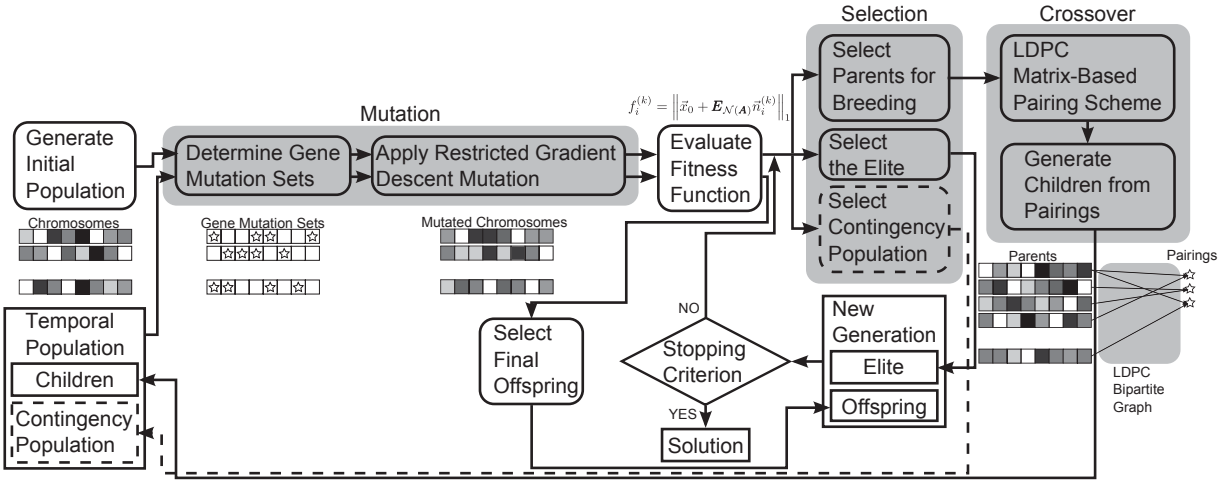

Fig. 1: Block diagram of the proposed genetic algorithm. The *chromosomes* represent different null space vectors. Some genes of each chromosome undergo mutation, being direction of such mutations given by the negative gradient of the fitness function. A small elite of the *fittest* chromosomes is passed to the next generation without modifications. A bigger group of fit individuals is selected for breeding. Pairings between parents are established according to an LDPC matrix.

population $\mathfrak{P}^{(k)}_{\text{breed}}$. The original concept of crossover, as described in [20], only contemplates pairings between two different chromosomes and generates two children per pairing by exchanging mutually-exclusive segments of genes. Both the establishment of pairings and the exchange of genetic information between parents are random processes. In this work we adopt a different crossover strategy. In the following we detail the two tasks of our crossover scheme, namely, establishing pairings and generating children from the pairings.

We adopt the general *pairing scheme* introduced in [21] for defining how n_{breed} selected individuals are implied in n_{pair} pairing events. This pairing strategy is not random, but deterministic, based on Low-Density Parity-Check (LDPC) codes, and allows for a custom number of parents per pairing $d \geq 2$.

For each of the n_{pair} pairings, an adjustable number of children per pairing n_{cpp} is generated. Each child of a pairing is obtained as a different weighted sum of the parent chromosomes implied in the pairing. The weights are all positive, drawn from a uniform distribution, and normalized to sum up 1. The total number of children is $n_{\text{child}} = n_{\text{cpp}} n_{\text{pair}}$.

F. New Generation and Stopping Criterion

The temporal population $\mathfrak{P}^{(k)}_{\text{child}} \cup \mathfrak{P}^{(k)}_{\text{cont}}$ undergoes mutation and posterior fitness evaluation (see Fig. 1). The final offspring, $\mathfrak{P}^{(k)}_{\text{off}}$, is then composed by the $n_{\text{ind}} - n_{\text{elite}}$ fittest individuals of this (mutated) temporal population. The next generation is given by the union $\mathfrak{P}^{(k+1)} = \mathfrak{P}^{(k)}_{\text{off}} \cup \mathfrak{P}^{(k)}_{\text{elite}}$. The fittest chromosome is selected to distill a prospective solution from the population:

$$\begin{aligned} \hat{\vec{x}}^{(k)} &= \vec{x}_0 + \boldsymbol{E}_{\mathcal{N}(\boldsymbol{A})} \vec{n}^{(k)}_{i_k} \\ i_k &= \arg\min_i \left\| \vec{x}_0 + \boldsymbol{E}_{\mathcal{N}(\boldsymbol{A})} \vec{n}^{(k)}_i \right\|_1 \end{aligned} \tag{7}$$

A threshold on the maximum number of generations or on the variation of the minimal l_1 norm of the temporal solutions can be used as stopping criteria.

V. Experiments and Results

The proposed null-space-based l_1-minimizing GA is compared to OMP, the Chambolle and Pock's (C&P) primal-dual algorithm [5], Loffeld's l_1-minimizing Kalman filter [1], a baseline performing pure gradient descent on null-space domain (NSP-GD), and the standard multiobjective GA NSGA-II [17] as CS recovery algorithms. OMP and NSGA-II directly search for an l_0 minimizer, while the others solve Eq. 3 instead, despite not all of them operate in $\mathcal{N}(\boldsymbol{A})$. Our implementation of the l_1-minimizing Kalman filter uses an Aitken-based convergence acceleration for automatically adjusting the synthetic l_1 shrinkage at each iteration. A series of experiments has been carried out to evaluate the different approaches in terms of sparse recovery performance. A different s-sparse signal $\vec{x} \in \mathbb{C}^n$ is randomly generated for each experiment. Both the real and imaginary parts of the nonzero complex coefficients are drawn from *i.i.d.* normal distributions of zero mean and unit variance, and the resulting $\vec{x}$ is then l_2-normalized. We use a *best complex antipodal spherical code* (BCASC) as close-to-optimal measurement matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ in Eq. 2. Our own fast implementation of the method in [22] was used to construct $\boldsymbol{A}$. We consider different experimental cases for different values of the parameters $\delta = m/n$ and $\rho = s/m$, with constant $n = 128$. More specifically, the entire $\delta - \rho$ plane, i. e., $0 \leq \delta \leq 1$, $0 \leq \rho \leq 1$ is evaluated by means of 16 equally-spaced discrete steps per parameter.

The same population size ($n_{\text{ind}} = 50$) and maximum number of generations $K \in \{5, 25, 50\}$ was considered for both our approach and NSGA-II. The maximum number of iterations allowed for all other algorithms is also set to be K, except for OMP, for which $10K$ is used. The

other parameters of our GA are set as follows: $k_{\text{elite}} = 0.1$, $p_{\text{mut gen}} = 1$, $k_{\text{breed}} = 0.5$, $n_{\text{pair}} = 3n_{\text{breed}}$, $d = 2$, $n_{\text{cpp}} = 3$.

Fig. 2 provides the obtained Donoho-Tanner graphs (ρ *vs.* δ) of normalized recovery error. The best-performing algorithm is OMP, for which the failure cases are confined to the top-left corner of the $\delta - \rho$ plane. The non-negligible errors in the top-right corner of the top-left plot of Fig. 2 are due to the fact that OMP cannot estimate the full signal support when its cardinality exceeds the number of iterations. The C&P's algorithm, NSP-GD, and our GA show similar behavior, with failure regions located at the top-left corners of the Donoho-Tanner graphs, but larger than OMP. As few as 50 iterations suffice to attain exact reconstruction for more than half of the Donoho-Tanner graph using C&P's algorithm, being slightly better than the other two alternatives. Our GA widely outperforms NSGA-II, whose performance does not improve much across iterations. The proposed GA is attractive for very low number of iterations (< 25), for which the error is much lower than using the NSP-GD baseline. As the number of iterations grow, the chromosomes in the population become too similar due to the gradient-biased mutation scheme and the GA has no advantage over NSP-GD. The l_1-minimizing Kalman filter requires a larger number of iterations and, in fact, for 500 iterations the C&P's algorithm and the Kalman filter deliver similar Donoho-Tanner graphs. In short terms, the proposed null-space-based GA with gradient-biased mutation exhibit better performance than comparable algorithms for very low number of iterations, while classical approaches, like OMP or the C&P's algorithm remain the best choice for larger number of iterations.

VI. Conclusion

A GA has been proposed for solving a linearly-constrained l_1 minimization, with application as CS reconstruction algorithm. To the best of our knowledge, we are the first proposing a GA that works exclusively in $\mathcal{N}(\boldsymbol{A})$. This translates into individuals of size $n - m$, rather than n, as it was the case in previous works. Also, the integration of the linear constraints in the shape of the basis of the subspace where the *chromosomes* live allows us using the l_1 norm of the temporal solutions as single *fitness function*, while other approaches require a multiobjective fitness function with an additional term for the reprojection error. Mutation is not fully random, but biased towards the gradient descent direction, thus accelerating convergence.

The performance of the proposed approach as sparse recovery algorithm in CS has been evaluated. In the experiments synthetic s-sparse n-dimensional complex signals are to be recovered from $m \ll n$ measurements. Donoho-Tanner graphs were generated for the entire range of the parameters $0 < \delta = m/n \leq 1$ and $0 < \rho = s/m \leq 1$ and five reference algorithms were considered for comparison: OMP, the C&P's primal-dual algorithm, an accelerated l_1-minimizing Kalman filter, a NSP-GD baseline, and NSGA-II. The results show that the proposed GA offers superior performance for very low number of iterations. This, together with its greater parallelization potential paves the way for faster CS reconstruction. If the number of iterations is unconstrained, OMP and the C&P's algorithm still remain the best choices.

References

[1] O. Loffeld, T. Espeter, and M. Heredia Conde, "From weighted least squares estimation to sparse cs reconstruction: l_1-minimization in the framework of recursive kalman filtering," in *3rd International Workshop on Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa)*, Jun. 2015, pp. 149–153.

[2] E. J. Candès, "Compressive sampling," in *Proceedings on the International Congress of Mathematicians*, Aug. 2006, pp. 1433–1452.

[3] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[4] E. J. Candès and J. Romberg, "$l1$-magic: Recovery of sparse signals via convex programming," 2005.

[5] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imaging Vis.*, vol. 40, no. 1, pp. 120–145, May 2011. [Online]. Available: http://dx.doi.org/10.1007/s10851-010-0251-1

[6] N. Vaswani, "Kalman filtered compressed sensing," in *Proceedings of the International Conference on Image Processing, ICIP 2008*, San Diego, California, USA, Oct. 2008, pp. 893–896. [Online]. Available: https://doi.org/10.1109/ICIP.2008.4711899

[7] ——, "LS-CS-residual (LS-CS): Compressive sensing on least squares residual," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4108–4120, Aug. 2010.

[8] A. Carmi, P. Gurfil, and D. Kanevsky, "Methods for sparse signal recovery using kalman filtering with embedded pseudo-measurement norms and quasi-norms," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2405–2409, Apr. 2010.

[9] O. Loffeld, A. Seel, M. Heredia Conde, and L. Wang, "A nullspace based L1 minimizing kalman filter approach to sparse CS reconstruction," in *11th European Conference on Synthetic Aperture Radar (EUSAR 2016)*, Hamburg, Germany, Jun. 2016.

[10] ——, "Sparse cs reconstruction by nullspace-based l_1 minimizing kalman filtering," in *11th International Conference on Communications (COMM 2016)*, Bucharest, Romania, Jun. 2016.

[11] Z. Lin, "Image adaptive recovery based on compressive sensing and genetic algorithm," in *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 2, May 2012, pp. 346–349.

[12] L. Zhang and X. Zeng, "Image adaptive reconstruction based on compressive sensing and the genetic algorithm via ROMP," in *2015 2nd International Conference on Information Science and Control Engineering*, Apr. 2015, pp. 265–268.

[13] L. Zhang, "Image adaptive reconstruction based on compressive sensing via CoSaMP," in *2015 2nd International Conference on Information Science and Control Engineering*, Apr. 2015, pp. 760–763.

[14] S. Bi, N. Xi, K. W. C. Lai, H. Min, and L. Chen, "Multi-objective optimizing for image recovering in compressive sensing," in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2012, pp. 2242–2247.

[15] L. Lorenzi, F. Melgani, and G. Mercier, "Contextual genetic algorithm for compressive sensing reconstruction of vhr images," in *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS*, Jul. 2013, pp. 4150–4153.

[16] ——, "Missing-area reconstruction in multispectral images under a compressive sensing perspective," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 7, pp. 3998–4008, Jul. 2013.

[17] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, Sep. 1994.

[18] D. Li, M. Li, Y. Shen, Y. Wang, and Q. Wang, "GA-BFO based signal reconstruction for compressive sensing," in *2013 IEEE International Conference on Information and Automation (ICIA)*, Aug. 2013, pp. 1023–1028.

[19] J. Shah, I. Qureshi, A. Khaliq, and H. Omer, "Sparse signal recovery based on hybrid genetic algorithm," *Research Journal of Recent Sciences*, vol. 3, pp. 86–93, Jan. 2014.

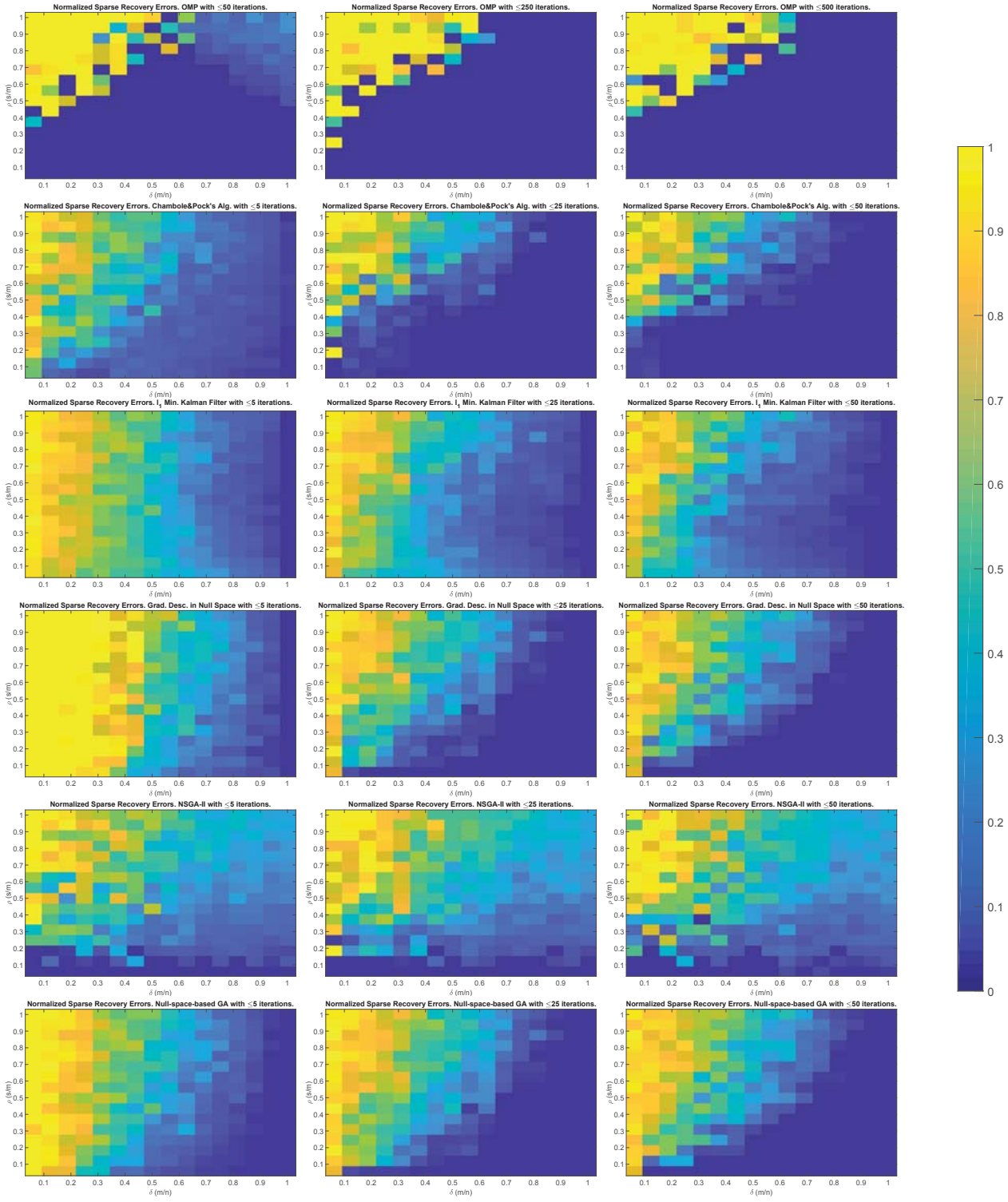

Fig. 2: Donoho-Tanner graphs of the recovery errors obtained using (rowwise from top to bottom) OMP, the Chambolle and Pock's algorithm, the l_1-minimizing Kalman filter, gradient descent on null-space domain, NSGA-II and our approach. The number of iterations/generations increases from left to right.

[20] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[21] M. Heredia Conde and O. Loffeld, "A genetic algorithm for compressive sensing sparse recovery," in *2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT) (ISSPIT-2017)*, Bilbao, Spain, Dec. 2017.

[22] H. Zörlein and M. Bossert, "Coherence optimization and best complex antipodal spherical codes," *IEEE Transactions on Signal Processing*, vol. 63, no. 24, pp. 6606–6615, Dec. 2015.