# A Genetic Algorithm for Compressive Sensing Sparse Recovery

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

***Abstract*—The advent of compressive sensing (CS) theory opened the possibility of linking the sensing effort, that is, the volume of data being produced by the sensor, to the amount of *information* it conveys, rather than to the desired sensor bandwidth, as traditional sampling theory suggests. Consequently, in the typical CS scenario, one ends up with a set of few measurements and the challenge is to recover a signal whose dimensionality is much higher than the number of measurements, typically under the assumption of being *sparse*. One faces, therefore, a constrained $l_0$ minimization problem. Despite being ubiquitous in nature, finding the solution with lowest $l_0$ norm is known to be NP-hard. In this work we propose mimicking the nature to approach a solution. More specifically, we design a genetic algorithm (GA) that, despite being based on the rules of evolution of biological systems, is fully tailored to our specific problem. Adopting the terminology from genetics, our *chromosomes* are representations of different support configurations, with an associated restricted-support temporal solution. The *fitness* of each chromosome is measured in terms of reprojection error of the associated solution. We deal with the sparsity requirement by means of a generalized *crossover* strategy based on support set overlap, rather than explicitly adding an $l_0$ or $l_1$ regularizer to the fitness function. We show that the proposed algorithm outperforms the generic multiobjective GA NSGA-II for solving the CS constrained $l_0$ minimization in terms of $l_2$ reconstruction error, at no cost in execution time.**

## I. Introduction

Most signal acquisition devices sample the input signals at the Nyquist rate, supposing that they have a bandlimited frequency spectrum. Frequently, the quality of a sensor is measured by its bandwidth, which is desired to be as high as possible. This translates into large Nyquist rates, which in turn lead to huge data streams. Consequently, the data needs to be compressed right after being sensed. A question that naturally arises at this point is whether it is possible to *sense in a compressed fashion*, that is, acquire less but more informative data, rather than compressing right after sensing. A positive answer has been provided by the groundbreaking theory of *compressive* (or *compressed*) sensing, which shows that most real signals can be recovered from a reduced number of measurements, often much lower than that suggested by the Shannon sampling theorem. The reason behind is that the minimum dimensionality in which a signal can be embedded at no information loss relates to the information content of the signal, rather than to its ambient dimensionality. Nevertheless, this does not imply that the original signal can be trivially recovered from the measurements, since this means solving an underdetermined problem. In order to recover the signal one needs to exploit the bounded information constraint, which is typically expressed as an $l_0$ minimization of the signal representation in some basis where it admits a sparse representation.

The relatively large amount of work on methods for recovering the underlying sparse signal from the compressed measurements leads to the question whether the perspective we use to approach the problem is the right one. The concept of *sparse* or *minimal complexity* modeling is a very natural one and, in fact, economy of resources is ubiquitous in the nature, being this principle of economy often used as a guideline for explaining nature in science. Such is the direction pointed by the famous *Occam's razor*, which states that it is futile to do with more [resources] what can be done with fewer. At the end of the day, such beautiful formulations translate into massively underdetermined problems, where the *simplest solution* can very hardly or not at all be approached due to the large ambient dimensionality. Still, nature seems to be good at approaching such optimal solutions. In this work we propose *using the nature to fight the nature*, in other words, using a nature-inspired solver scheme for solving a nature-inspired problem. More specifically, the structure of our algorithm is based on the concept of a *genetic algorithm* (GA), which, in turn, imitates the process of natural adaptation present in biological systems.

Despite we build upon the concept of GAs, our algorithm is tailored to solve the aforementioned constrained $l_0$ minimization in a CS framework. We have tried, nevertheless, to keep the algorithm formulation as general as possible, so that parameters affecting the way the new generation is obtained from the previous can be customized. This may allow accelerating the algorithm by adapting its behavior to the signals we expect to be dealing with. Complementary, we try to rely as less as possible on random and, in fact, we propose a deterministic scheme for selecting the pairings between individuals that comes from error correcting coding. As a result, the proposed algorithm is able to outperform generic multiobjective GAs for the specific task of linearly-constrained $l_0$ minimization at an equivalent cost in terms of population size and execution time.



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

## II. Related Work

The works we present here can be classified in two main groups, according to whether a GA is directly used as recovery algorithm for solving Eq. 4 or, differently, is used only as secondary algorithm for adjusting some parameters of the CS framework. The latter are only marginally related to our work.

To this second group belong [1]–[3], where a GA is used to retrieve the optimal number of measurements, $m$, that is, for sensing matrix design, while conventional CS recovery algorithms are used for sparse recovery. More specifically, OMP is used in [1], ROMP in [2] and CoSaMP in [3]. A concern on these works is the fact that, if measurements for the largest value of $m$ considered by the GA during optimization are indeed available, then one should make use of all these measurements, instead of trying to find out *a posteriori* whether acquiring a lower number of measurements would have been sufficient.

Another example of using a GA for parameter optimization in CS is [4], where Total Variation (TV) minimization is used to recover an image from compressed measurements acquired with a single-pixel camera setup. The GA is implemented in an outer optimization loop and its aim is optimizing the free parameters of the TV-minimization model. A weakness of this work is the fact that the authors make explicit use of the real image, which is, in principle, unknown, to evaluate the error term of their multiobjective fitness function.

Representatives of the group of works actually solving Eq. 4 by means of a GA are [5], [6], where the nondominated sorting GA named NSGA-II [7] 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. NSGA-II returns several optimal (nondominated) solutions along the Pareto front. The authors choose the median solution, hoping for a tradeoff between sparsity and reconstruction error.

The signal reconstruction algorithm proposed in [8] combines the GA with bacterial foraging optimization (BFO) and is able to find the global optimum to multimodal optimization problems such as that in Eq. 4. Similarly to our approach, they also define an *elite*, which gets directly transferred to the next generation. The performance of the algorithm is compared to that of OMP. Unfortunately, no details are provided on how the multimodality of the fitness function is handled. Instead of that, the authors affirm that the constrained $l_0$ minimization can be reduced to a simple $l_2$ reprojection error minimization if all the individuals of the initial population have the same sparsity as the solution. For this to hold, not only the sparsity should be the same, but also the sparse support and, furthermore, one should take care that the sparse support gets preserved from one generation to the next. Additionally, if the signal support was known beforehand, the constrained least squares solution can be computed in a closed form by means of the Moore-Penrose pseudoinverse and no iterative method is needed.

In [9] the authors also claim using a GA to solve Eq. 4. The algorithm follows the typical structure of fitness evaluation, crossover and mutation, but no further details are provided. Surprisingly, the fitness function is not multiobjective, but the $l_p$ reprojection error for $p \in \{0, 0.5, 1, 2\}$. This does not enforce sparsity of the solution and remains unclear how the sparsest solution is achieved. The *hybrid* GA of [10] includes a step of modified parallel coordinate descent (PCD) to prevent the algorithm from getting stuck. The algorithm solves Eq. 4 for known $s$ by means of an explicit $s$-thresholding after crossover.

## III. The Compressive Sensing Scenario

The mathematical theory of compressive sensing (CS) [11], [12] 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 states that a continuous signal is completely determined by a number of equidistant samples acquired at a rate that is twice the maximum frequency contained in the signal (Nyquist rate). Instead, CS theory requires the signal to be *sparse* in some basis or tight frame. If this is the case and the sensing scheme satisfy some additional requirements, 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 so-called $l_0$ 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:

$$\|\vec{x}\|_0 \leq s \tag{2}$$

that is, if $\vec{x}$ has, at maximum, $s$ non-zero elements. Now the challenge is to reconstruct $\vec{x}$ from a reduced number of linear measurements $m \ll n$. Thus, the classic CS measurement model is an underdetermined linear system of the form:

$$\vec{y} = \boldsymbol{A}\vec{x} \tag{3}$$

where $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ denotes the *measurement matrix*, which explains how the vector of measurements $\vec{y} \in \mathbb{C}^m$ relates to the signal $\vec{x}$ and may be the composition of the actual *sensing matrix*, modeling the sensing process, and a dictionary, modeling the sparsifying transformation. Ideally, we would like to find the sparsest $\vec{x}$ satisfying Eq. 3, that is, we look for the solution to the following constrained $l_0$ minimization:

$$\hat{\vec{x}} = \underset{\vec{x} \in \mathbb{C}^n}{\arg\min} \|\vec{x}\|_0 \text{ subject to } \vec{y} = \boldsymbol{A}\vec{x} \tag{4}$$

Unfortunately, finding a solution to Eq. 4 is, in general, NP-hard.

## IV. A Genetic Algorithm for Compressive Sensing Sparse Recovery

In the following we present our GA for solving the constrained $l_0$ minimization introduced in Eq. 4. 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 (see [13] for details). In Fig. 1 we provide a block diagram of the algorithm. Operations or processes are depicted as rectangles with rounded vertices and sets of individuals as rectangles with sharp vertices. The individuals of our population are *chromosomes* (adopting the nomenclature from genetics) with $n$ binary genes each, that is, each gene has two *alleles*, leading to $2^n$ different possible genetic structures or *genotypes*. Each chromosome of the population represents a configuration for the signal support and we will denote with $\Omega_i^{(k)}$ the $i^{th}$ chromosome of the $k^{th}$ generation.

### A. Initial Population

We know from null space considerations that the sparsity $s$ should not exceed $m/2$. Therefore, any initial population $\mathfrak{P}^{(0)}$ has to be such that:

$$\mathfrak{P}^{(0)} = \left\{\Omega_i^{(0)}\right\}_{i=1}^{n_{\text{ind}}}, 1 \leq \left|\Omega_i^{(0)}\right| \leq \lfloor m/2 \rfloor \forall i \tag{5}$$

where $n_{\text{ind}}$ is the number of individuals or size of $\mathfrak{P}^{(0)}$. For each chromosome, the number of nonzero genes, i. e., the cardinality of the support set, $s_i^{(0)} = \left|\Omega_i^{(0)}\right|$, is drawn from an uniform probability distribution between 1 and $\lfloor m/2 \rfloor$. Note the possibility of integrating *a priori* knowledge on $s$ by means of custom distributions. Despite the sparsities are drawn from some probability distribution, the $s_i^{(0)}$ indices belonging to the $i^{th}$ support set are not randomly selected. We make use of the *least squares* solution to the underdetermined system in Eq. 3 to select the signal support by thresholding, namely

$$\begin{aligned} \Omega_i^{(0)} &= \text{supp}\left(\tau_{s_i^{(0)}}\left(\vec{x}_{\text{LS}}\right)\right), \ \forall 1 \leq i \leq n_{\text{ind}} \\ \vec{x}_{\text{LS}} &= \boldsymbol{A}^{\dagger}\vec{y} \end{aligned} \tag{6}$$

where $\tau_{s_i^{(0)}}(\cdot)$ denotes $s_i^{(0)}$-thresholding and $\boldsymbol{A}^{\dagger}$ the Moore-Penrose pseudoinverse of $A$. The restriction to a support set of cardinality lower than $m$ implies that each chromosome uniquely defines a temporal solution to Eq. 3. For any arbitrary population at generation $k$, the associated temporal solutions are given by the Moore-Penrose pseudoinverse of the corresponding *support-restricted* measurement matrix, that is:

$$\vec{x}_i^{(k)} \in \mathbb{C}^n \text{ with } \begin{cases} \vec{x}_{\Omega_i}^{(k)} = \left(\vec{x}_i^{(k)}\right)_{\Omega_i^{(k)}} = \boldsymbol{A}_{\Omega_i^{(k)}}^{\dagger}\vec{y} \\ \vec{x}_{\bar{\Omega}_i}^{(k)} = \left(\vec{x}_i^{(k)}\right)_{\bar{\Omega}_i^{(k)}} = \vec{0} \in \mathbb{C}^{n-s_i^{(k)}} \end{cases} \tag{7}$$

where the vector and matrix subscripting denotes restriction to the set denoted by the subscript. Since $\Omega_i^{(k)}$ is uniquely associated to $\vec{x}_i^{(k)}$ the abuse of notation proposed in the left equalities is legit.

### B. Mutation

The process of *mutation* applied over some population $\mathfrak{P}^{(k)}$ means that the chromosomes undergo stochastic changes in the values of its genes, according to some probability distribution. In our case, we define the probability that *any* gene of the chromosome undergoes any mutation as $p_{\text{mut}}^{(k)} = \mathcal{P}\left(\Omega_i^{(k)\prime} \neq \Omega_i^{(k)}\right), \ \forall i$, where the prime here denotes the same set after mutation. We use a different (higher) mutation probability for the first iteration and the same ($p_{\text{mut}}$) for all the following, i. e., $p_{\text{mut}}^{(0)} > p_{\text{mut}}^{(k)} = p_{\text{mut}}, \ \forall k \geq 1$. We distribute the aggregate mutation probability uniformly between a subset of genes allowed to mutate. We only allow mutation of genes corresponding to non-support elements, yielding

$$p_{\text{mut gen},i}^{(k)} = 1 - \sqrt[n-s_i^{(k)}]{(1-p_{\text{mut}})}. \tag{8}$$

### C. The Fitness Function

Typically, the fitness function in a GA is a function that takes as an input a population individual or chromosome and outputs a measure of how *fit* it is, i. e., how good it is according to some evaluation criterion. In other words, the fitness function is the function for which we seek an optimizer. Provided that we aim to solve Eq. 4, one would expect a multiobjective fitness function, in which both the $l_0$ norm of the solution and some data fidelity term are to be simultaneously minimized, as in [5], [6]. Differently, our chromosomes are temporal support sets and the gene mutations are, in practice, binary flips. This allows us easing the search for a solution and adopting the reprojection error as single-objective fitness function. We rely on an appropriate initialization and crossover strategy to keep the temporal sparsities as low as possible, still seeking for the solution that minimizes the reprojection error. Therefore, our fitness function is

$$\begin{aligned} f\left(\Omega_i^{(k)}\right) = \|\boldsymbol{A}\vec{x}_i^{(k)} - \vec{y}\|_2 &= \|\boldsymbol{A}_{\Omega_i^{(k)}}\vec{x}_{\Omega_i}^{(k)} - \vec{y}\|_2 \\ &= \|\boldsymbol{P}_{\Omega_i^{(k)}}^{\perp}\vec{y}\|_2 \\ \boldsymbol{P}_{\Omega_i^{(k)}}^{\perp} &= \boldsymbol{I} - \boldsymbol{A}_{\Omega_i^{(k)}}\boldsymbol{A}_{\Omega_i^{(k)}}^{\dagger} \end{aligned} \tag{9}$$

which is equivalent to the data fidelity term of the multiobjective functions in [5], [6] and in [8]. $\boldsymbol{P}_{\Omega_i^{(k)}}^{\perp}$ is an *orthogonal projector*, which projects $\vec{y}$ onto the subspace that is orthogonal to that spanned by the columns of $\boldsymbol{A}$ indexed by $\Omega_i^{(k)}$.

### D. The Selection Scheme

Two main groups of individuals are selected from the population using Eq. 9. The most restrictive group is an *elite* of very few individuals exhibiting the highest fitness, which is denoted $\mathfrak{P}_{\text{elite}}^{(k)}$ at generation $k$ and is of size $\left|\mathfrak{P}_{\text{elite}}^{(k)}\right| = n_{\text{elite}} = k_{\text{elite}} n_{\text{ind}}$. $\mathfrak{P}_{\text{elite}}^{(k)}$ is directly transferred

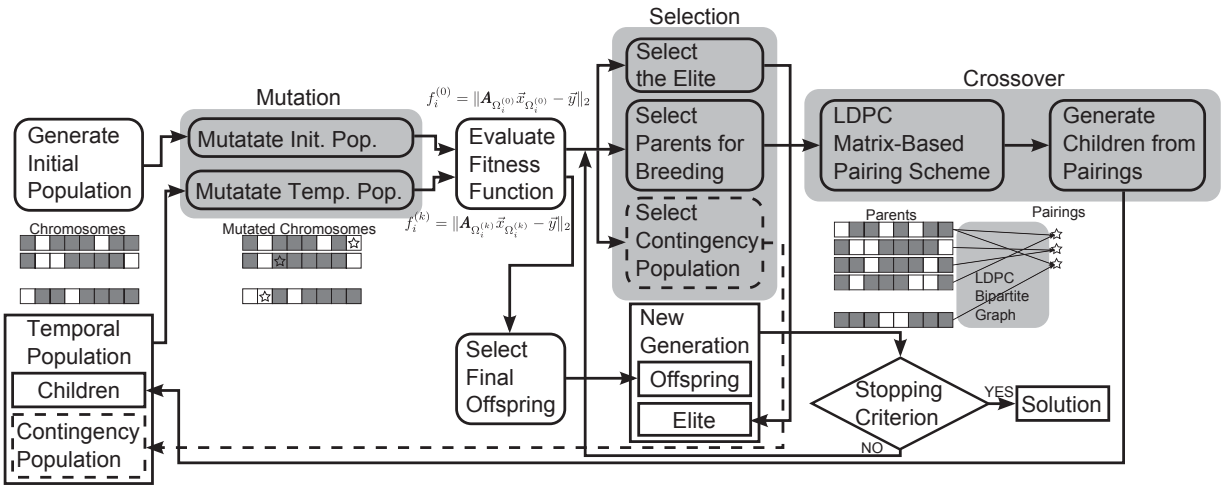


Fig. 1: Block diagram of the proposed genetic algorithm. Population's *chromosomes* are different plausible signal support configurations. Mutations mean changes in the corresponding signal support. The reprojection error of the least squares solution restricted to a given support configuration is adopted as fitness function. Pairings between parents are established according to an LDPC matrix. Generations of chromosomes succeed one another until some stopping criterion is met, and the fittest individual is used to generate a solution.

to the next generation. The other group is constituted by the individuals that will breed the next generation. This group, denoted $\mathfrak{P}^{(k)}_{\text{breed}}$, is also created selecting the fittest individuals and is of size $\left|\mathfrak{P}^{(k)}_{\text{breed}}\right| = n_{\text{breed}} = k_{\text{breed}} n_{\text{ind}}$, with $0 < k_{\text{elite}} \ll k_{\text{breed}} < 1$. Clearly, $\mathfrak{P}^{(k)}_{\text{elite}} \subset \mathfrak{P}^{(k)}_{\text{breed}} \subset \mathfrak{P}^{(k)}$. Eventually, a third subset of the population may become necessary in order to prevent the population size from shrinking in each new generation: the *contingency population*, $\mathfrak{P}^{(k)}_{\text{cont}} \subset \mathfrak{P}^{(k)} \setminus \mathfrak{P}^{(k)}_{\text{elite}}$. In the usual case that $n_{\text{child}} \geq n_{\text{ind}} - n_{\text{elite}}$, then $\mathfrak{P}^{(k)}_{\text{cont}} = \emptyset$.

### *E. Crossover*

We call *crossover* the process of generating a population of children, $\mathfrak{P}^{(k)}_{\text{child}}$, of size $\left|\mathfrak{P}^{(k)}_{\text{child}}\right| = n_{\text{child}}$ from the previously-selected breeding population $\mathfrak{P}^{(k)}_{\text{breed}}$. Differently from the original concept of crossover [13], in which randomness plays a major role and pairings are binary, we propose a more general crossover framework in which pairings are deterministic and of custom size. The number of children per pairing is an adjustable parameter of this deterministic scheme. Let's analyze separately the two tasks of our crossover framework, namely, establishing pairings and generating children from the pairings.

The *pairing scheme* refers to the way $n_{\text{breed}}$ selected individuals are implied in $n_{\text{pair}}$ pairing events. We contemplate here the adjustable parameter $d \geq 2$, which is the number of parents required per pairing. A further novelty is that the selection of the $d$ parents implied in each pairing is not random, but done according to a Low-Density Parity-Check (LDPC) code [14]. We denote the number of rows and columns of the LDPC matrix as $n_{\text{check}}$ and $n_{\text{sym}}$, respectively, with $n_{\text{check}} < n_{\text{sym}}$. In a Tanner graph representation, $n_{\text{sym}}$ is the number of *symbol* nodes and $n_{\text{check}}$ is the number of *check* nodes in the bipartite graph. Each column of the LDPC matrix contains a small number of ones, $d_{\text{s}}$, and each row contains $d_{\text{c}}$ ones, also low. In our case, and in order to account for any desired combination of $n_{\text{breed}}$ and $n_{\text{pair}}$, we set

$$\begin{aligned} n_{\text{sym}} &= \max\left(n_{\text{breed}}, n_{\text{pair}}\right) \\ n_{\text{check}} &= \min\left(n_{\text{breed}}, n_{\text{pair}}\right) \\ d_{\text{s}}, d_{\text{c}} &\begin{cases} d_{\text{s}} = \frac{n_{\text{breed}}}{n_{\text{pair}}} d, \;\; d_{\text{c}} = d \text{ if } n_{\text{breed}} > n_{\text{pair}} \\ d_{\text{s}} = d, \;\; d_{\text{c}} = \frac{n_{\text{pair}}}{n_{\text{breed}}} d \text{ otherwise} \end{cases} \end{aligned} \tag{10}$$

and each one in the matrix means that a specific individual of $\mathfrak{P}^{(k)}_{\text{breed}}$ is taking part in a specific pairing. We adopt the *Progressive Edge-Growth* method [15] for constructing the LDPC matrix with the parameter values of Eq. 10.

The next step is defining how children are generated from the $d$ parents at each pairing. We propose generating children by levels of support set overlap (shortened SSO). The number of SSO levels, $1 \leq k_{\text{SSO}} \leq d-1$, is equivalent to the number of children that will be generated per pairing. For each SSO level, $l$, the new chromosome (support set) of the pairing $p$ is generated from the $d$ parent chromosomes as follows

$$\Omega^{(k)}_{p,l} = \left\{ j \mid \left| \bigcup_{q=1}^{d} j \cap \Omega^{(k)}_{i_{p,q}} \right| \geq l \right\}, \; 1 \leq l \leq k_{\text{SSO}}, \tag{11}$$

where $i_{p,q}$ is the index of the $q^{th}$ parent chromosome implied in the pairing $p$. Therefore, the total number of children is $n_{\text{child}} = k_{\text{SSO}} n_{\text{pair}}$. In case $n_{\text{child}} < n_{\text{ind}} - n_{\text{elite}}$, a contingency population of size $n_{\text{ind}} - n_{\text{elite}} - n_{\text{child}}$ is added to the children to avoid population shrinkage.

### F. New Generation and Stopping Criterion

The temporal population composed by the union of $\mathfrak{P}^{(k)}_{\text{child}}$ and $\mathfrak{P}^{(k)}_{\text{cont}}$ undergoes mutation and posterior fitness evaluation, as initially done with $\mathfrak{P}^{(0)}$. The $n_{\text{ind}} - n_{\text{elite}}$ fittest individuals of this (mutated) temporal population are selected as final offspring, $\mathfrak{P}^{(k)}_{\text{off}}$. Then the next generation is obtained as

$$\mathfrak{P}^{(k+1)} = \mathfrak{P}^{(k)}_{\text{off}} \cup \mathfrak{P}^{(k)}_{\text{elite}}. \tag{12}$$

The fittest chromosome is selected as prospective solution:

$$\begin{aligned} \hat{\vec{x}}^{(k)} &= \vec{x}^{(k)}_{i_k} \\ i_k &= \arg\min_i \left\| \boldsymbol{P}^{\perp}_{\Omega^{(k)}_i} \vec{y} \right\|_2 \end{aligned} \tag{13}$$

where $\boldsymbol{P}^{\perp}_{\Omega^{(k)}_i}$ is the orthogonal projector defined in Eq. 9. Typical stopping criteria are thresholds on the maximum number of generations and on the reprojection error of $\hat{\vec{x}}^{(k)}$, that is, the fitness of the fittest chromosome.

## V. Experiments and Results

We compare the proposed GA to OMP, the Chambolle and Pock's primal-dual algorithm [16], and the standard multiobjective GA NSGA-II [7] for approaching a solution to Eq. 4 in the CS setup. In order to assess their relative sparse recovery performance, a series of experiments has been carried out. For each experiment, an $s$-sparse signal $\vec{x} \in \mathbb{C}^n$ is generated at random. Both the real and imaginary parts of each nonzero complex coefficient 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 measurement matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ in Eq. 3. BCASCs have been shown to be optimal CS measurement matrices in terms of coherence. We use our own fast implementation of the method in [17] 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, we evaluate the entire $\delta - \rho$ plane, i. e., $0 \leq \delta \leq 1$, $0 \leq \rho \leq 1$ by means of 16 equally-spaced discrete steps per parameter.

We use the same initial population ($n_{\text{ind}} = 50$) and maximum number of generations $K \in \{5, 25, 50\}$ for both our approach and the NSGA-II. The maximum number of iterations allowed for OMP and the Chambolle and Pock's algorithm is also set to be $K$. The rest of the parameters are set as follows: $k_{\text{elite}} = 0.1$, $p^{(0)}_{\text{mut}} = 0.9$, $p_{\text{mut}} = 0.9$, $k_{\text{breed}} = 0.5$, $n_{\text{pair}} = 3n_{\text{breed}}$, $d = 3$, $k_{\text{SSO}} = d - 1$.

Fig. 2 provides the results in the shape of Donoho-Tanner graphs of normalized recovery error. OMP exhibits the best performance in terms of recovery error, restricting the failure cases to the top-left corner of the $\delta - \rho$ plane. For only 50 iterations (top left plot) OMP cannot estimate the full signal support when its cardinality exceeds this value and, consequently, non-negligible errors appear also in the top-right corner of the graph. The Chambolle and Pock's algorithm exhibits similar behavior, also showing a top-left failure region in 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. NSGA-II delivers acceptable results even for a low number of iterations. In general, for most of the the $\delta - \rho$ cases considered, the result does not improve much with the number of iterations. The proposed GA for constrained $l_0$ minimization (last row) outperforms the reference GA, showing much better performance in the cases of large $\delta$. Also the bottom region of exact reconstruction is larger than in the NSGA-II case and slightly improves with the number of iterations. The top-left failure area is significantly pushed towards that corner in comparison to NSGA-II, getting closer to the behavior observed for the Chambolle and Pock's algorithm, which we locally outperform in the bottom-left region for 50 iterations.

In terms of time, OMP is the fastest option, specially when $s$ is low, followed by the Chambolle and Pock's algorithm. The GAs exhibit execution times that are two orders of magnitude longer than the Chambolle and Pock's algorithm. Both NSGA-II and our approach showed similar execution times. Both for our approach and the Chambolle and Pock's algorithm the execution time slightly increases with $\delta$ and is independent of $\rho$, while NSGA-II shows a more uniform behavior.

## VI. Conclusion

In this work a GA has been proposed that approaches a solution to the linearly-constrained $l_0$ minimization problem that needs to be solved in a conventional CS framework. Differently from generic multiobjective GAs, which can also be used for this task, the structure of our algorithm has been specifically designed for solving this problem. The *chromosomes* represent different sparse support configurations and a temporal solution is uniquely associated to each of them. A single *fitness function* is used to evaluate how well each temporal solution fits the measurements and the sparsity requirement is enforced by means of a crossover strategy based on support set overlap (SSO). Pairings between chromosomes are established following a deterministic scheme and a custom number of individuals can take part in each pairing.

We evaluate the performance of our approach in a typical CS scenario, in which an $s$-sparse $n$-dimensional signal is to be recovered from $m$ measurements. Our experiments covered the whole range of the parameters $0 < \delta = m/n \leq 1$ and $0 < \rho = s/m \leq 1$ and three reference algorithms were considered for comparison: OMP, the Chambolle and Pock's primal-dual algorithm and NSGA-II. OMP and the Chambolle and Pock's algorithm showed to be superior to the GAs, both in terms of reconstruction error and execution time. Restricting the attention to the GAs, the proposed GA widely outperforms NSGA-II, significantly pushing the failure area towards the top-left corner of the Donoho-Tanner graphs, approaching the performance of the Chambolle and Pock's algorithm. The huge parallelization potential of GAs w.r.t. conventional algorithms, together with the encouraging results presented here, boosts the attractiveness of further research on this topic.

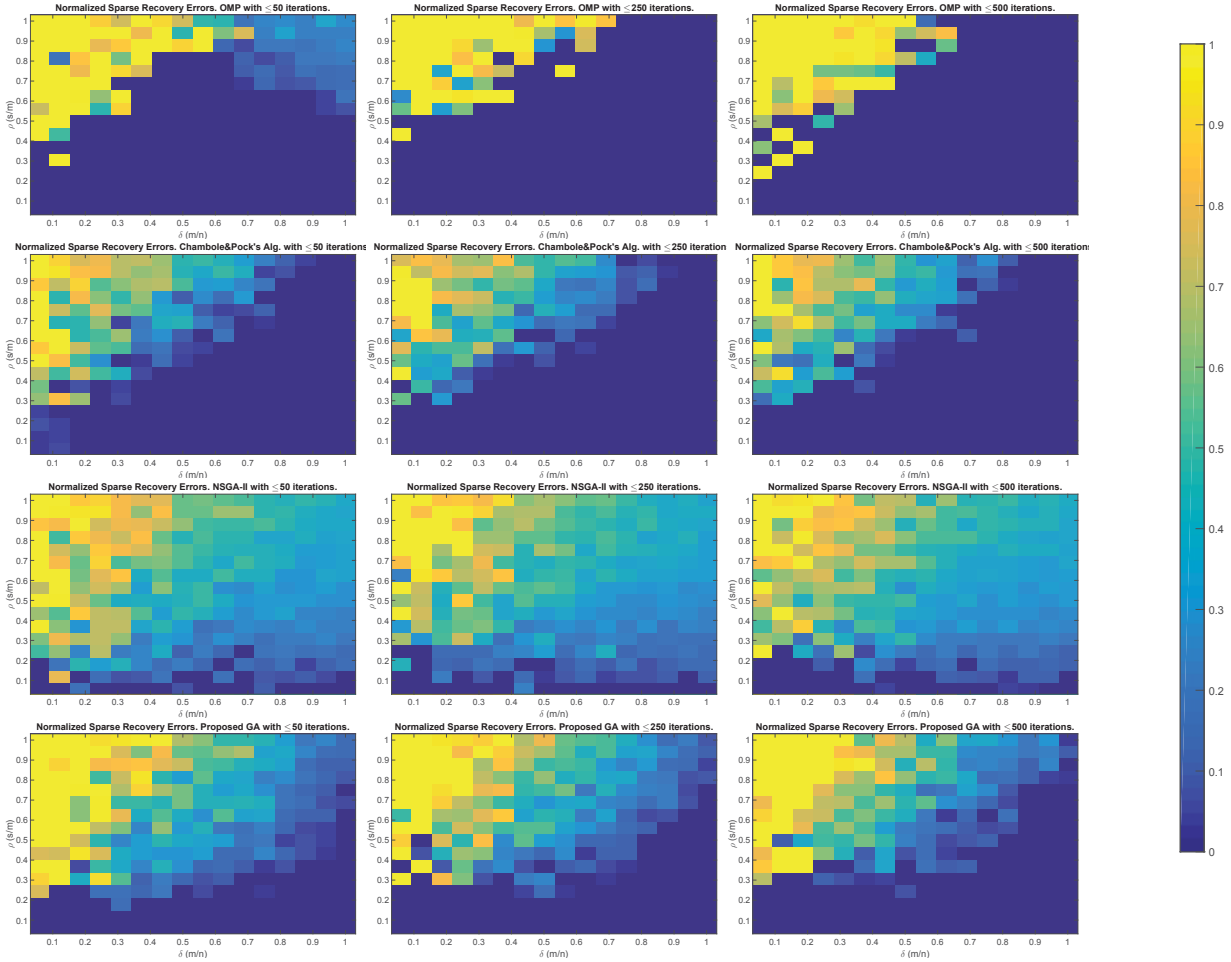


Fig. 2: Donoho-Tanner graphs of the recovery errors obtained using (rowwise from top to bottom) OMP, the Chambolle and Pock's algorithm, NSGA-II and our approach. The number of iterations/generations increases from left to right.

## REFERENCES


[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] ——, "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.

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

[8] 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.

[9] S. E. El-Khamy, M. B. Abdel-Malek, and S. H. Kamel, "An improved reconstruction technique for wavelet-based compressive spectrum sensing using genetic algorithm," in *2014 31st National Radio Science Conference (NRSC)*, Apr. 2014, pp. 99–106.

[10] 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.

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

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

[13] 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.

[14] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[15] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Progressive edge-growth tanner graphs," in *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, vol. 2, 2001, pp. 995–1001 vol.2.

[16] 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

[17] 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.