A non-convex adaptive regularization approach to binary optimization

A non-convex adaptive regularization approach to binary optimization

V. Cerone, S. M. Fosson*, D. Regruto

***Abstract*— Binary optimization is a long-time problem ubiquitous in many engineering applications, e.g., automatic control, cyber-physical systems and machine learning. From a mathematical viewpoint, binary optimization is an NP-hard problem, to solve which one can find some suboptimal strategies in the literature. Among the most popular approaches, semidefinite relaxation has attracted much attention in the last years. In contrast, this work proposes and analyzes a non-convex regularization approach, through which we obtain a relaxed problem whose global minimum corresponds to the true binary solution of the original problem. Moreover, because the problem is non-convex, we propose an adaptive regularization that promotes the descent towards the global minimum. We provide both theoretical results that characterize the proposed model and numerical experiments that prove its effectiveness with respect to state-of-the-art methods.**

I. Introduction

Mixed-integer programming problems arise in several engineering areas, ranging from model predictive control and industrial production planning to cyber-physical systems. Problems with binary variables are of particular interest in several applications, such as in the optimization of networked systems whose nodes can be either activated or deactivated, as in cyber-physical systems, when controlling the available resources by a suitable selection of sensors and actuators. More precisely, it is important to know how to achieve a given control objective with a minimum or with a prescribed number of activated sensors/actuators; see [1], [2], [3]. In fact, on the one hand, the control of the number of used devices helps to save resources; on the other hand, an opportune sensor selection decreases the computational complexity at the processing unit by selecting the most important observations; see [4].

We encounter a different application of binary programming in detection and localization problems: given a monitored area subdivided into reference points, we have to estimate the presence of one or more targets in those points by estimating a binary vector from linear measurements; see, e.g., [5].

More recently, binary problems have attracted the attention in deep learning, specifically in the construction of sparse neural networks with a limited number of parameters. To control the number of these parameters, binary masks are used in recent literature; see, e.g., [6] and references therein, which recasts the problem into mixed-integer programming.

From a mathematical viewpoint, mixed-integer programming is challenging as it is inherently NP-hard. For this motivation, one can find several low-complex suboptimal methods in the literature. In particular, for quadratic optimization with binary variables, one often exploits semidefinite programming relaxation (SDR), also known as Shor relaxation; see [7]. The key idea is to relax the variable matrix xx^T, $x \in \mathbb{R}^n$, into an $n \times n$ symmetric positive semidefinite matrix X, and solve the so-obtained semidefinite program. The tightness of this approach is studied in [8], [9]. When x is known to be binary, i.e., $x \in \{0,1\}^n$, the constraints $X_{i,i} = X_{1,i}$, $i = 1, \dots, n$ can be added to the SDR; similarly, if $x \in \{-1,1\}^n$, the trace of X is $\mathrm{tr}(X) = n$; see, e.g., [10], [11].

Further developments of SDR for binary quadratic optimization include Lasserre's polynomial optimization methods, where hierarchies of SDR's are solved, see [12], [13]. From Theorem 4.2 in [13], the solution of the Shor relaxation of a binary quadratic problem is the desired optimum if it has rank 1; in this case, one can extract the desired binary minimizer; see, e.g.,[14].

For this motivation, the rank minimization in SDR plays a fundamental role in binary optimization, see [15] for a complete overview. As rank minimization is NP-hard, one often exploits the minimization of the nuclear norm, which is the convex relaxation of the rank; moreover, one can find non-convex approaches like the $\log\text{-}\det$ heuristic; see [16]. In [17], SDR for binary quadratic optimization is enhanced by an eigenvalue analysis of the solution, particularly by the knowledge that the eigenvalues are binary and the multiplicity of the null eigenvalue is $n-1$. This analysis suggests adding non-convex regularizer $-\langle X, X\rangle$, that improves the performance.

In this work, we propose a novel approach to binary optimization. In particular, we focus on problems where the ℓ_0-norm of the solution is known. Specifically, we propose to relax the original problem with $x \in \{0,1\}^n$ into $x \in [0,1]^n$ and add suitable non-convex regularizers that encourage the solution to be binary. Since the overall problem is non-convex, we propose an adaptive method to avoid local minima. The proposed approach is analyzed and characterized in terms of its global optimality and numerical simulations are proposed to evaluate the practical effectiveness.

We organize the paper as follows. In Sec. II, we state the problem. Sec. III, presents and analyzes the proposed approach. In Sec. IV, we illustrate the numerical experiments; in Sec. V, we draw some conclusions.

* Corresponding author. The authors are with the Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy; e-mail: sophie.fosson@polito.it.

II. Problem statement

The aim of this paper is to solve binary optimization problems of the kind

$$\min_{x\in\{0,1\}^n} \mathcal{F}(x) \text{ s.t. } \sum_{i=1}^{n} x_i = k \tag{1}$$

where $\mathcal{F}:\mathbb{R}^n\to\mathbb{R}$ is a suitable cost functional that admits a minimum, and $k\in\{1,\dots,n\}$ is known and typically $k\ll n$. This is the case, for example, in resource allocation problems and in sensor/actuator selection, when the number of used devices may be set a priori, according to a given budget.

In many relevant applications, e.g., in hybrid model predictive control and in linear regression, the cost functional is quadratic, i.e., $\mathcal{F}(x)=x^TQx$, where $Q\in\mathbb{S}^n$, $\mathbb{S}^n$ being the space of $n\times n$ symmetric matrices. We remark that the presence of linear terms is envisaged: by adding a component equal to 1 in x, the problem can be written in a homogeneous form, see, e.g., [17] for details.

In this paper, we assume that $\min_{x\in\{0,1\}^n}\mathcal{F}(x) = \min_{x\in[0,1]^n}\mathcal{F}(x)$. This occurs, for example, when the purpose is to find the binary solution to an underdetermined linear equation $Ax=b$, $A\in\mathbb{R}^{m,n}$, $m<n$. If we assume that a binary solution $\widetilde{x}$ exists, the related least squares cost functional $\|Ax-b\|_2^2$ achieves global minimum at $\widetilde{x}$. One application of this problem is binary compressed sensing, see, e.g., [18] and references therein.

A. Related literature

Since Problem (1), is non convex, even when $\mathcal{F}$ is convex, different relaxations are proposed in the literature. If $\mathcal{F}$ is convex, the most natural convex relaxation consists in replacing $x\in\{0,1\}^n$ by $x\in[0,1]^n$. As to quadratic problems $\min_{x\in\{0,1\}^n} x^TQx$, a most refined strategy is the Shor relaxation, see [9], which leads to the SDR

$$\begin{aligned}&\min_{X\in\mathbb{S}_+^n}\langle Q,X\rangle\\ &\text{s. t. } X_{i,i}=X_{1,i},\quad i=1,\dots,n\end{aligned} \tag{2}$$

where $\mathbb{S}_+^n\subset\mathbb{S}_n$ is the subspace of positive semidefinite matrices. The constraint $X_{i,i}=X_{0,i}$ supports the fact that $x_i=x_i^2$. We remark that the constraint $\sum_{i=1}^n x_i=k$ can be relaxed into the minimization of $(\sum_{i=1}^n x_i-k)^2$ and included in the quadratic cost functional. As mentioned in Sec. I, the Shor solution is exact if it has rank 1; in this case $X=xx^T$ and $x\in\{0,1\}^n$. To encourage the 1-rank solution, in the SDR one can penalize the nuclear norm, which corresponds to the trace in $\mathbb{S}_+^n$. On the other hand, if $\sum_{i=1}^n x_i=k$, one can add the constraint $\mathrm{tr}(X)=k$.

Another strategy to reduce the rank is to penalize $\log\det(X)$. This approach is not convex; however, one can exploit iterative local linearization to achieve a local minimum, see [16] for details. In [17], the penalization of the term $-\langle X,X\rangle$ is proposed. The rationale behind this approach is the following. If $\sum_{i=1}^n x_i=k$ then $\mathrm{tr}(X)=\sum_i v_i=k$, where v_i are the eigenvalues. To have 1-rank solutions, one would desire eigenvalues in $\{0,k\}$. To promote this sparse binary structure of the spectrum, one can maximize the energy of the eigenvalues' vector $\mathbf{v}\in[0,1]^n$, which corresponds to $\|\mathbf{v}\|_2^2=\langle X,X\rangle$. This approach is shown to enhance the state-of-the-art methods, see [17, Sec. 4].

III. Proposed approach and analysis

In this work, we propose and analyze a novel approach to Problem (1). Basically, we propose the following relaxation:

$$\begin{aligned}&\min_{x\in[0,1]^n}\mathcal{F}(x)+\lambda\mathcal{P}_\varepsilon(x) \text{ s.t. } \sum_{i=1}^n x_i=k\\ &\mathcal{P}_\varepsilon(x)=\sum_{i=1}^n p_\varepsilon(x_i)\\ &p_\varepsilon:[0,1]\to[0,1]\\ &p_\varepsilon(z)=\frac{\log\left(\frac{z}{\varepsilon}+1\right)}{\log\left(\frac{1}{\varepsilon}+1\right)}\\ &\lambda>0,\ \varepsilon>0.\end{aligned} \tag{3}$$

Intuitively, p_ε is a concave regularizer which promotes the sparsity of the solution. The parameter ε tunes the concavity: if $\varepsilon\to\infty$, $\sum_{i=1}^n p_\varepsilon(x_i)\to\|x\|_1$, while if $\varepsilon\to 0$, $\sum_{i=1}^n p_\varepsilon(x_i)\to\|x\|_0$. On the one hand, p_ε closer to ℓ_0 is preferable as more consistent with the original formulation of sparse optimization problems; on the other hand, p_ε closer to ℓ_1 drives the problem towards convexity. In our model, the sparsity of the solution is set to k, and we typically assume $k\ll n$, therefore a sparsity promoting effect is desirable. However, as the sparsity k is set, the rationale behind the use of p_ε is different, and is more related to its capability of encouraging binary solutions. More precisely, the following theorem proves that, by solving Problem (3), one obtains the solution of Problem (1).

Theorem 1: Let us consider Problem (1), and assume that its solution $\widetilde{x}\in\{0,1\}^n$ with $\sum_{i=1}^n\widetilde{x}_i=k$, is unique. Moreover, let $\widetilde{x}$ be a global minimizer for $\mathcal{F}(x)$, $x\in[0,1]^n$. Then, $\widetilde{x}$ is the global minimizer of Problem (3), for all $\lambda>0$ and $\varepsilon>0$.

Proof: To prove the statement, we start by characterizing the minimizers of $\mathcal{P}_\varepsilon(x)=\sum_{i=1}^n p_\varepsilon(x_i)$ in $[0,1]^n$. As $\sum_{i=1}^n x_i=k$,

$$\begin{aligned}&\mathcal{P}_\varepsilon(x)=\\ &=c_1\left[\sum_{i=1}^{n-1}\left(\log(x_i+\varepsilon)\right)+\log\left(k-\sum_{i=1}^{n-1}x_i+\varepsilon\right)-c_2\right]\end{aligned} \tag{4}$$

where $c_1=\left(\log\left(\frac{1}{\varepsilon}+1\right)\right)^{-1}>0$ and $c_2=n\log(\varepsilon)$. As c_1 and c_2 are not relevant to characterize the minimizers of $\mathcal{P}_\varepsilon$, in the following, we neglect them without loss of generality. Now, we compute the stationary points of $\mathcal{P}_\varepsilon$. The components of the gradient are

$$\begin{aligned}\nabla_i\mathcal{P}_\varepsilon(x)&=\frac{1}{x_i+\varepsilon}-\frac{1}{k-\sum_{j=1}^{n-1}x_j+\varepsilon}\\ &=\frac{k-\sum_{j=1}^{n-1}x_j-x_i}{(x_i+\varepsilon)(k-\sum_{j=1}^{n-1}x_j+\varepsilon)}.\end{aligned} \tag{5}$$

Therefore, $\nabla_i \mathcal{P}_\varepsilon(x) = 0 x_i = k - \sum_{j=1}^{n-1} x_j$, which yields to

$$\nabla \mathcal{P}_\varepsilon(x) = 0 \Leftrightarrow x_i = \frac{k}{n} \in [0,1] \text{ for all } i = 1 \dots, n.$$

This is the unique stationary point. By computing the eigenvalues of the Hessian matrix in this point, one can conclude straightforwardly that this stationary point is a maximum, we omit the details for brevity. As a consequence, the minimum points are on the boundaries of the hypercube $[0,1]^n$.

Moreover, one can prove by induction that the minimum points are at a vertex of the hypercube. In fact, this is trivial to prove for $n = 2$. For $n = 3$, a point on a face has a at least one component in $\{0,1\}$. Then, by minimizing over the other two components, we recast the problem to the case $n = 2$, and obtain that all the components are in $\{0,1\}$. By iterating this argument, we obtain the proof for a generic n.

In conclusion, $\underset{x\in[0,1]^n, \sum_i x_i = k}{\operatorname{argmin}} \mathcal{P}_\varepsilon(x) \in \{0,1\}^n$. More precisely, any $x \in \{0,1\}^n$ with k components equal to 1 a global minimizer for $\mathcal{P}_\varepsilon$, for any $\varepsilon > 0$. Then, if we consider the composite functional $\mathcal{F}(x) + \lambda \mathcal{P}_\varepsilon(x)$, we conclude that $\widetilde{x} \in \{0,1\}^n$ defined in the statement is the global minimum, as both the terms $\mathcal{F}$ and $\mathcal{P}_\varepsilon$ individually achieve global minima at $\widetilde{x}$. The value of λ is not affecting this fact. ■

Theorem 1 states that Problem (3) is well posed, since its global minimum is the desired solution of Problem (1). Actually, we have replaced a combinatorial problem with a continuous one. Nevertheless, Problem (3) is non-convex, therefore finding the global minimum is still challenging. Iterative descent algorithms can be used to look for local minima; for example, the gradient descent algorithm can be used if $\mathcal{F}$ is differentiable. Nevertheless, local minima are generally far from the desired binary solution.

To overcome this problem, we propose the following iterative strategy. We implement a suboptimal solution to Problem (1) that achieves a local minimum $z_0 \in [0,1]^n$, e.g., a gradient descent algorithm. If z_0 is not binary, we know that it is a local (not global) minimum. Then, we remove this local minimum by adding a penalization to it. For example, we can consider a piecewise penalization $r_{z_0} : (0,1) \to [0,1]$:

$$r_{z_0}(x_i) = \begin{cases} \frac{1}{z_{0,i}} x_i & x_i < z_{0,i} \\ \frac{1}{1-z_{0,i}}(1 - x_i) & x_i \geq z_{0,i}. \end{cases} \tag{6}$$

We notice that r_{z_0} has domain $(0,1)$, that is, the binary components are not penalized.

Then, we locally solve the problem

$$\begin{aligned} &\min_{x\in[0,1]^n} \mathcal{F}(x) + \lambda \mathcal{P}_\varepsilon(x) + \alpha \mathcal{R}_{z_0}(x) \text{ s.t. } \sum_{i=1}^n x_i = k \\ &\mathcal{R}_{z_0}(x) = \sum_{i=1}^n r_{z_0}(x_i) \\ &\lambda > 0, \ \alpha > 0 \ \varepsilon > 0. \end{aligned} \tag{7}$$

Once a new local minimum is obtained, if it does not correspond to the global minimum, the procedure is iterated. In other terms, at each iteration, we adapt the functional in order to transform the current local minimum in a point which is not a local minimum. The new functional is then different, while the global minimum remains the same and corresponds to the desired solution. In fact, since $r_{z_0}(x_i) = 0$ when $x_i \in \{0,1\}$, Problem (7) has the same global minimum of Problem (3). On the other hand, z_0 is the maximum for r_{z_0}, then by considering a suitable α, z_0 is no more a local minimum of the overall cost functional.

By iterating this procedure, we modify the shape of functional each time, without modifying its global minimum and removing local minima. Actually, we are not guaranteed that by this method we do not create other local minima; however, in practice we generally obtain a steeper slope towards the global minimum and the reduction of the basins of attraction of local minima.

Intuitively, the effect is to shake the functional by keeping fixed the global minimum, which is expected to promote the descent to the global minimum, if a descent algorithm is implemented. We illustrate this point through an example in Sec. III-A. On the other hand, in principle we cannot foresee how many times we have to repeat to procedure to make it successful; this strongly depends on the initial conditions. Therefore, this strategy is proposed as an heuristic method. However, in Section IV we show that it is effective.

We remark that by considering a piecewise linear r_{z_0}, the problem is not differentiable. This could modified be considering other polynomial regularizers. However, in practice it suffices to choose a suitable initial point to avoid to incur in z_0 in Problem (7), then if $\mathcal{F}$ is differentiable one can run a gradient descent algorithm.

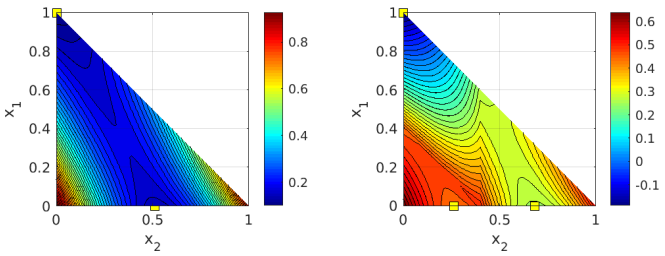

Fig. 1: Illustrative example: the global minimum is at $(1,0,0)^T$. The yellow circles indicate the minima. Left: Problem (3); right: Problem (7). In the second case, we have two local minima instead of one, but the slope is steeper towards the global minimum. If we run a gradient-based algorithm starting from $\frac{1}{3}(1,1,1)^T$ in the second case we achieve the global minimum.

The overall proposed strategy is summarized in Algorithm 1.

A. Illustrative example

We propose a low-dimensional example, that can be graphically visualized and, at the same time, motivates the proposed strategy. Let us consider $A = (0.3, 1.5, -1)$ and $b = 0.3$. Then, the equation $Ax = b$ has binary solution $\widetilde{x} = (1,0,0)^T$. Our aim is to find it by solving Problem (3), where $\mathcal{F}(x) = \frac{1}{2}\|Ax - b\|_2^2$. For simplicity, we set the constraint $\sum x_i = 1$, so that we can graphically visualize $\mathcal{F}$

Algorithm 1 Non-convex adaptive regularization algorithm

Input: $\mathcal{F}$, $\lambda > 0$, $\varepsilon > 0$, $\alpha > 0$

Output: $x_{T_{stop}}$ = estimate of $\widetilde{x}$

1: Compute a local minimizer $x_0 \in [0,1]^n$ of

$$\mathcal{F}(x) + \lambda \mathcal{P}_\varepsilon(x) \text{ s.t. } \sum_i x_i = k$$

2: If $x_0 \in \{0,1\}^n$, then $x_0 = \widetilde{x} \Rightarrow$ STOP

3: **for all** $t = 1, \dots, T_{stop}$ **do**

4: Compute a local minimizer $x_t \in [0,1]^n$ of

$$\mathcal{F}(x) + \lambda \mathcal{P}_\varepsilon(x) + \alpha \sum_{\tau=0}^{t-1} \mathcal{R}_{x_\tau}(x) \text{ s.t. } \sum_i x_i = k$$

5: If $x_t \in \{0,1\}^n$, then $x_t = \widetilde{x} \Rightarrow$ STOP

6: **end for**

as a function of two variables; this is depicted in Fig. 1, on the left, by a contour line graph. The yellow squares indicate all the minima. We see that a local minimum is present at $z_0 = (0, 0.51, 0.49)^T$. Then, we add the penalization r_{z_0}, and we obtain the cost functional depicted in Fig. 1, on the right. This new functional has two novel local minima. However, since the global minimum remains at the same level, we obtain a steeper slope towards it. For example, if we start from the mid point $\frac{1}{3}(1,1,1)^T$ in the second case we achieve the global minimum, while this is not true in the first case.

B. Different concave regularizations

It is worth noticing that the proposed approach can be extended to other concave regularizers beyond the $\log$ function. For example, Theorem 1 can be straightforwardly proved also for penalties that maximize the energy, e.g., $-\|x\|_2^2$. In this work, we focus on the $\log$ as it is known to be more effective to promote sparsity, also given the possibility to tune its concavity by the parameter ε, see, e.g., [14] for details. If $\mathcal{F}$ is quadratic, the use of $-\|x\|_2^2$ may be preferable to keep the overall problem quadratic, and then use Shor relaxation. However, we remark that by Shor relaxation the term $-\|x\|_2^2$ is relaxed to $-\mathrm{tr}(X)$, which is set to k in our framework. Therefore, this strategy is not effective.

A special remark should be devoted to $-\|x\|_2^4$, which again maximizes the energy. We notice that $-\|x\|_2^4 = -\|x\|_2^2\|x\|_2^2 = x^T x x^T x$. Since by Shor relaxation, xx^T is relaxed to $X \in \mathbb{S}_+^n$, we can relax $-\|x\|_2^4$ into $-x^T X x$, which in turn can be relaxed to $-\langle X, X\rangle$, which is exactly the method proposed in [17].

In this paper, we focus on the $\log$ regularizer, while analyzes on different concave regularizers are left for future work.

IV. Numerical experiments

In this section, we illustrate the results of some numerical simulations. First, we propose a simulation with synthetic data. Second, we propose a real-world experiment that addresses an indoor localization problem via wireless sensor networks.

A. Synthetic experiment

In the first experiment, we consider the following setting. We have an underdetermined linear system $Ax = b$ with a binary solution, with $n = 40$ and $m \in \{8, \dots, 18\}$, $m < n$. We consider different sparsity levels $k \in \{3, 4, 5\}$, $k \ll n$. We assume that the constraint $\sum_i x_i = k$ is one of the m linear equations in $Ax = b$. Our aim is to compute the correct binary solution by applying Algorithm 1 to the cost functional $\mathcal{F}(x) = \frac{1}{2}\|Ax - b\|_2^2$.

In this experiment, we set $\lambda = 10^{-4}$, $\varepsilon = 10^{-2}$, $\alpha = 1$, $T_{stop} = 20$. For local minimization, we implement the gradient algorithm, with initial condition $\frac{1}{2}\mathbf{1} \in [0,1]^n$. We compare the proposed approach to the classical SDR obtained by Shor relaxation and to the algorithm KBE proposed in [17]. In that paper, KBE is shown to perform better than Shor relaxation, nuclear norm and $\log\det$ methods. The matrix A has Gaussian independent entries $\mathcal{N}(0, \frac{1}{m})$. The support of $\widetilde{x}$ is generated uniformly at random. The results, illustrated in Fig. 2, are averaged over 100 random runs. Specifically, we show the exact recover rate for each approach, i.e., how many times $\widetilde{x}$ is correctly identified. The method proposed in this paper is denoted by LOG in the figure.

In Fig. 2, we can see that the proposed method substantially improves with respect to KBE, by reducing the number of measurements that are necessary to find the desired solution. We notice that in principle $m = 1$ might be sufficient, provided that $\widetilde{x}$ is the unique binary solution. However, by increasing m we reduce the number of local minima and we improve the overall performance. At the same m, the proposed approach can achieve an improvement of 20% with respect to KBE. Regarding the number of measurements, the recovery rate obtained by the proposed method at $m = 12$ is close to that of KBE with $m = 14$, which means that we save around the 14% of measurements.

Regarding the run time, in this experiment SDR and KBE require around 10^{-2} seconds to run. We specify that for them we use the Mosek C++ Fusion API, see [19], which is quite fast in solving SDRs. On the other hand, the complexity of the proposed method is low, as it based on gradient descent algorithm. However, by assuming a maximum number of iterations $T_{stop} = 20$, which is quite large, we observe a run time that can vary from 10^{-2} up to 10 seconds. The optimization of the run time, by considering different local minimization algorithms and by refining the tuning of the parameters, is ongoing work.

B. Localization problem

In the last years, indoor localization has been attracting much attention in cyber-physical systems, for different purposes, such as monitoring, surveillance, control of industrial production lines and of unmanned vehicles. Indoor localization is challenging as GPS technologies are not available for

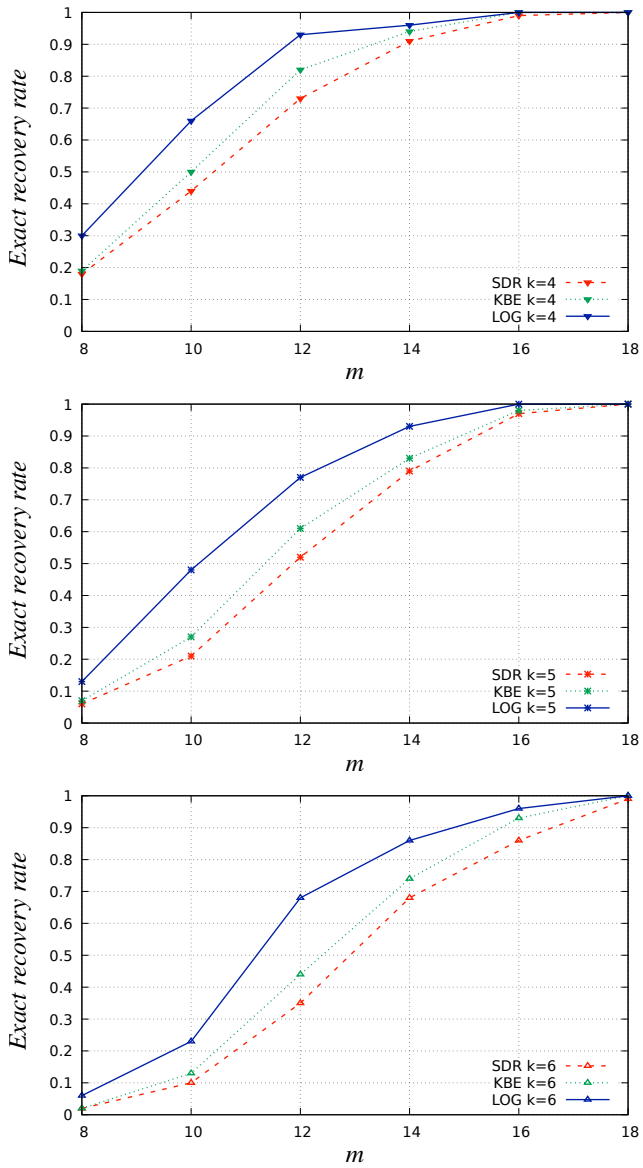

Fig. 2: Synthetic experiment, $n = 40$, $k = 4, 5, 6$: exact recovery rate. We compare the proposed approach, denoted as LOG, to SDR and KBE [17]

it and one can find many methodologies in the literature; see, e.g., [20] for an overview.

A valuable approach is to use wireless sensor networks: sensors are deployed in the indoor area to monitor and they acquire radio signals from the targets that they aim to localize. For example, one can use sensors that measure the received signal strength (RSS). In the so-called fingerprinting approach, one performs a training phase in which the area is split into n references points, and, in turn, the target broadcasts a signal from each point. The m sensors acquire the related RSSs and build a dictionary $A \in \mathbb{R}^{m,n}$. After the training phase, the localization is performed as follows: each sensor takes one RSS measurement, so that one collects a vector $y \in \mathbb{R}^m$, and the localization can be performed by solving $Ax = y$, corrupted by some noise. Actually, x is binary vector and $\sum_i x_i = l$ is equal to the number of targets to localize, which can be known a priori; see [21], [22] for details.

In this section, we propose a localization experiment via wireless sensor network, by considering the RSS-fingerprinting model. We notice that we assume that the processing of the data is performed by a central processing unit, that gathers all the measurements taken from the wireless sensor network. However, distributed algorithms might be also considered, to perform the localization in-network.

We consider the following setting. We aim to localize $k = 5$ targets a 10×10 m^2 indoor area. We assume to split the area into $n = 100$ square cells of side 1 m and our aim is to identify in which cell the target is. Then, we consider a wireless sensor network composed by $m \in \{20, 22, \dots, 34\}$ sensor nodes, deployed uniformly at random on the area. To simulate to RSS measurement, we exploit the model defined by the IEEE 802.15.4 standard, as reported in [21, Equation 11]. The targets are deployed uniformly at random on the area. A measurement noise is added, corresponding to an SNR of 25dB. We perform 100 random runs.

We show the configuration and the results of a single experiment in Fig. 3. In this figure, we can see a 10×10 m^2 area, where we have randomly deployed 20 sensors, represented by the green circles; then, we place 5 targets, represented by the red squares, in different reference points. The green crosses are the positions estimated by KBE, which is this experiment provides the same result than SDR. Two targets are not correctly localized, with an error of 1 m for both. Instead, the proposed method, represented by the blue triangles, provides a correct localization of all the targets.

We illustrate the performance in Fig. 4, in terms of exact recovery rate, by comparing to SDR and KBE, as in the previous experiment. We can see that also in this realistic experiment, the proposed method has a better accuracy with respect to state-of-the-art methods, for all the considered numbers of measurements m.

V. Conclusions

In this work, we propose a novel approach to binary optimization. Given a minimization problem with binary variables, by assuming the knowledge of the sparsity of the solution, we propose to regularize it by an opportune non-convex penalization. We prove that the proposed problem is well-posed, that is, the global minimum is the desired solution. As the problem is non-convex, and it may be challenging to achieve the global minimum, we propose an adaptive iterative algorithm, in which, first, we run a descent algorithm that achieves a local minimum. Then, we add an adaptive regularizer that penalizes the obtained local minimum, without affecting the global minimum. This adaptive regularization encourages the descent to the global minimum. This result is confirmed by an improvement in the exact recovery rate observed in numerical simulations. Future work will be devoted to a rigorous analysis of the

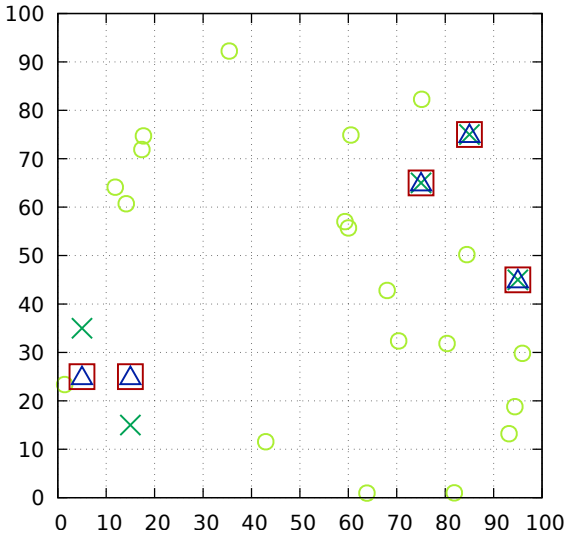

Fig. 3: Indoor localization target from RSS compressed measurements, in a 10×10 m^2 area: an example of configuration. The green circles represent the sensors; the red squares are the targets; the green crosses are the estimates by KBE, while the blue triangles are the estimates obtained by the proposed approach, which correctly identifies all the targets' positions.

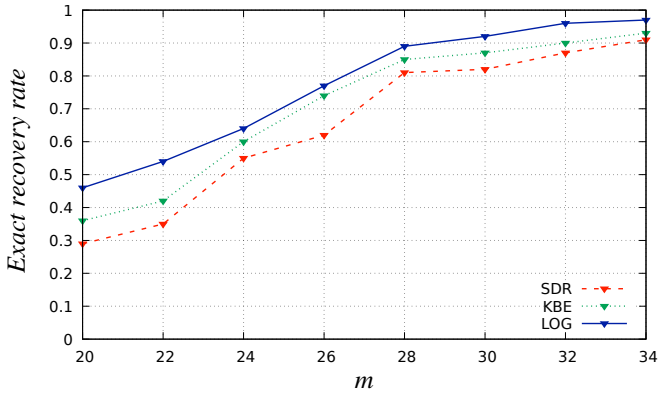

Fig. 4: Indoor localization, $n = 100$, $k = 5$: exact recovery rate. We compare the proposed approach, denoted as LOG, to SDR and KBE [17]

adaptive method and to verify its effectiveness in large scale problems.

REFERENCES

[1] A. F. Taha, N. Gatsis, T. Summers, and S. Nugroho, "Actuator selection for cyber-physical systems," in *Proc. Amer. Control Conf. (ACC)*, 2017, pp. 5300–5305.

[2] A. F. Taha, N. Gatsis, T. Summers, and S. A. Nugroho, "Time-varying sensor and actuator selection for uncertain cyber-physical systems," *IEEE Trans. Control Netw. Syst.*, vol. 6, no. 2, pp. 750–762, 2019.

[3] C.-Y. Chang, S. MartÃnez, and J. CortÃ©s, "Co-optimization of control and actuator selection for cyber-physical systems," *IFAC-PapersOnLine*, vol. 51, no. 23, pp. 118–123, 2018, 7th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2018.

[4] A. Carmi and P. Gurfil, "Sensor selection via compressed sensing," *Automatica*, vol. 49, no. 11, pp. 3304–3314, 2013.

[5] A. Bay, D. Carrera, S. M. Fosson, P. Fragneto, M. Grella, C. Ravazzi, and E. Magli, "Block-sparsity-based localization in wireless sensor networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 2015, no. 182, pp. 1–15, 2015.

[6] P. Savarese, H. Silva, and M. Maire, "Winning the lottery with continuous sparsification," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 11 380–11 390.

[7] N.-Z. Shor, "Quadratic optimization problems," *Sov. J. Comput. Syst. Sci*, vol. 26, 1987.

[8] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.

[9] Y. Nesterov, "Semidefinite relaxation and nonconvex quadratic optimization," *Optim. Methods Softw.*, vol. 9, no. 1-3, pp. 141–160, 1998.

[10] J. B. Lasserre, "A MAX-CUT formulation of 0/1 programs," *Oper. Res. Lett.*, vol. 44, no. 2, pp. 158–164, 2016.

[11] P. Wang, C. Shen, A. v. d. Hengel, and P. H. S. Torr, "Large-scale binary quadratic optimization using semidefinite relaxation and applications," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 39, no. 3, pp. 470–485, 2017.

[12] J. B. Lasserre, "Convergent lmi relaxations for nonconvex quadratic programs," in *Proc. IEEE CDC*, 2000, pp. 5041–5046.

[13] J. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optim.*, vol. 11, no. 3, pp. 796–817, 2001.

[14] S. M. Fosson and M. Abuabiah, "Recovery of binary sparse signals from compressed linear measurements via polynomial optimization," *IEEE Signal Process. Lett.*, vol. 26, no. 7, pp. 1070–1074, 2019.

[15] A. Lemon, A. M.-C. So, and Y. Ye, "Low-rank semidefinite programming: Theory and applications," *Found. Trends Optim.*, vol. 2, no. 1-2, pp. 1–156, 2016.

[16] M. Fazel, H. Hindi, and S. Boyd, "Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices," in *Proc. IEEE ACC*, vol. 3, 2003, pp. 2156–2162.

[17] V. Cerone, S. M. Fosson, and D. Regruto, "Enhancing low-rank solutions in semidefinite relaxations of boolean quadratic problems," in *Proc. IFAC World Congress*, 2020.

[18] S. M. Fosson, "Non-convex approach to binary compressed sensing," in *Asilomar Conf. Signals Syst. Comput.*, 2018, pp. 1959–1963.

[19] MOSEK, *Fusion API for C++*, 2019. [Online]. Available: www.mosek.com/documentation/

[20] D. Dardari, P. Closas, and P. M. Djurić, "Indoor tracking: Theory, methods, and technologies," *IEEE Trans. Vehic. Tech.*, vol. 64, no. 4, pp. 1263–1278, 2015.

[21] C. Feng, S. Valaee, and Z. Tan, "Multiple target localization using compressive sensing," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, 2009, pp. 1–6.

[22] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan, "Received-signal-strength-based indoor positioning using compressive sensing," *IEEE Trans. Mobile Comput.*, vol. 11, no. 12, pp. 1983–1993, 2012.