Journal of Combinatorial Optimization (2022) 44:583–602
https://doi.org/10.1007/s10878-022-00847-0

A solution approach for cardinality minimization problem based on fractional programming

S. M. Mirhadi[1] · S. A. MirHassani[1]

Accepted: 15 January 2022 / Published online: 22 February 2022

Abstract

This paper proposes a new algorithm for solving the linear Cardinality Minimization Problem (CMP). The algorithm relies on approximating the nonconvex and non-smooth function, $card(x)$, with a linear fractional one. Therefore, the cardinality minimization problem is converted to the sum-of-ratio problem. The new model is solved with an optimization algorithm proposed for finding the optimal solution to the ratio problem. In the numerical experiments, we focus on two types of CMP problems with inequality and equality constraints. We provide a series of examples to evaluate the performance of the proposed algorithm, showing its efficiency.

Keywords Cardinality minimization problem · Cardinality function · Sum-of-ratio problem · Sparse signals

1 Introduction

The cardinality minimization problem, which is to minimize the number of nonzero variables, is a nonconvex optimization problem that can be written as:

$$\begin{aligned} &\min z = card(x) \\ &s.t. \quad x \in X \end{aligned} \tag{1}$$

where $card(x)$ or $\|x\|_0$, denotes the number of nonzero entries in the vector $x \in \mathbb{R}^n$,

$$card(x) = \sum_{i=1}^{n} card(x_i)$$

✉ S. A. MirHassani
a_mirhassani@aut.ac.ir

1 Faculty of Mathematics and Computer Science, Amirkabir University of Technology (Polytechnic Tehran), Tehran, Iran

and

$$card(x_i) = \begin{cases} 1 & if x_i \neq 0 \\ 0 & if x_i = 0 \end{cases}$$

In other words, this problem is looking for the sparsest vector in the feasible region X that is a convex or nonconvex set described by a group of linear or nonlinear constraints. The optimization problems aim to find the sparsest solution for a linear or nonlinear equation system in most cases. CMP has many applications in the different fields of science, such as compressed sensing, signal and image processing (Bruckstein et al. 2009; Bian and Chen 2020; Chen et al. 2019), principal component analysis (Beck and Vaisbourd 2016), statistical regression (Bertsimas et al. 2016), and others.

The $card(x)$ is a nonconvex and non-smooth function, and the optimization problems with $card(x)$ are known as NP-hard problems (Natarajan 1995). In this paper, we consider the following issues:

$$\begin{aligned} &\min z = card(x) \\ &s.t. \quad x \in X \end{aligned} \tag{2}$$

where $X = \left\{x \in \mathbb{R}^n_{\geq 0} : Ax \geq b,\ x \in [0, 1]^n\right\}$ and $A \in \mathbb{R}^{m\times n}(m < n)$ is a real matrix of full row rank, $b \in \mathbb{R}^m$. The optimization problem (2) can be generalized to cover the case $x \in [-1, 1]^n$, by introducing the following variables and constraints

$$\begin{aligned} &x = v - u \\ &w \geq v,\ w \geq u \\ &v,\ w,\ u \geq 0 \end{aligned}$$

Therefore, we solve the following problem instead of (2):

$$\begin{aligned} &\min card(w) \\ &s.t. \quad w \geq v,\ w \geq u,\ x = v - u \\ &v,\ w,\ u \geq 0,\ x \in X' \end{aligned} \tag{3}$$

where $X^{'} = \{x \in \mathbb{R}^n : Ax \geq b,\ x \in [-1, 1]^n\}$. It is not difficult to see that $card(x^*) = card(w^*)$ where x^*, w^* are respectively the optimal solution of problems (2) and (3).

So, the cardinality minimization problem with $x \in [-1, 1]^n$ can be expressed as the equivalent problem (3) that the cardinality function is defined on nonnegative variables $w \in [0, 1]^n$. This problem is actually similar to (2). Therefore, without losing generality, we concentrate on problem (2) and effort to solve the problem (2) based on the desired fractional approach; see Sect. 3 for more detail.

A classical strategy that is proposed to solve the cardinality problem is based on integer programming by introducing axillary binary variables to model (2) and solve the MIP counterpart of the problem, which is:

$$\begin{aligned}
&\min \sum_{i=1}^{n} \gamma_i \\
&s.t. \\
&x_i \leq \gamma_i,\ \gamma_i \in \{0, 1\},\ i = 1, \ldots, n \\
&x \in X
\end{aligned}$$

This method, that used by Bertsimas et al. (2016), is not computationally applicable for large or even medium-sized problems. An old approach, pointed out by Markovsky and Huffel (Markovsky and Huffel 2007), is based on a least-square problem that is an unconstrained convex problem such as (4)

$$min\|Ax - b\|_2 \quad s.t. \quad x \in \mathbb{R}^n \tag{4}$$

The famous convex optimization method to solve (2) is substituted the $card(x)$ by l_1- norm:

$$\min\|x\|_1 \quad s.t. \quad x \in X \tag{5}$$

This convex method was proposed by Chen. et al. (Chen et al. 1998) and applied in many applications (see (Bruckstein et al. 2009; Jokar and Pfetsch 2008; Donoho 2006), among others). In signal processing, that a linear equation system defines the feasible region X, the equality constraint can be relaxed with $\|Ax - b\|_2 \leq \eta$, where $\eta > 0$ represents a predetermined noise level:

$$min\|x\|_1 \quad s.t. \quad \|Ax - b\|_2 \leq \eta,\ x \in \mathbb{R}^n \tag{6}$$

Tibshirani (1996) used the least-square problem (4) with penalized l_1- norm to encourage sparsity. This is called a LASSO-type problem and is written as follows:

$$min\|Ax - b\|_2 + \lambda\|x\|_1 \quad s.t. \quad x \in \mathbb{R}^n,\ \lambda \gg 0 \tag{7}$$

Another convex approximation method for problem (2) is reweighted l_1- minimization, which was proposed by Candès et al. (2008). In each iteration of this algorithm, the positive weights are generated as follows:

$$a_i^{(k+1)} = \frac{1}{\left|x_i^{(k)}\right| + \varepsilon},\ i = 1, \ldots, n \tag{8}$$

where (k) is the number of iteration and $x_i^{(k)} = argmin\{\sum_{i=1}^{n} a_i^{(k)} |x_i| s.t x \in X\}$.

Numerical experiments show reweighted l_1—minimization improved the performance of the l_1—minimization problem in many situations, so this method was followed by researchers such as (Lai and Wang 2011; Chen and Zhou 2014; Jia et al. 2019; Shi et al. 2020; Abdi 2013).

Table 1 The approximation functions for card(x)

The alternative function	Parameter value	References
$\boldsymbol{f}(\boldsymbol{x}) = \Vert\boldsymbol{x}\Vert_1$	–	Chen et al. (1998)
$\boldsymbol{f}(\boldsymbol{x}) = \Vert\boldsymbol{x}\Vert_{\boldsymbol{p}}{}^{\boldsymbol{p}}$	$p \in (0,\ 1)$	Lai and Wang (2011)
$\boldsymbol{f}_{\boldsymbol{\sigma}}(\boldsymbol{x}) = \frac{\boldsymbol{\sigma}\vert\boldsymbol{x}\vert}{\boldsymbol{\sigma}\vert\boldsymbol{x}\vert+1}$	$\sigma \to \infty$	Cui et al. (2019)
$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{log}\left(1 + \frac{\vert\boldsymbol{x}\vert}{\boldsymbol{t}}\right)$	$t \to 0$	Candès et al. (2008)
$\boldsymbol{f}(\boldsymbol{x}) = 1 - \boldsymbol{e}^{-\frac{\vert\boldsymbol{x}\vert}{\boldsymbol{t}}}$	$t \to 0$	Mangasarian (1999)

Another approach to solving the problem (2) proposed in Lai and Wang (2011); Ge et al. 2011) is to use the nonconvex function $\Vert.\Vert_p{}^p$ for $p \in (0,\ 1)$ instead of$card(x)$. This minimization is motivated by the following equivalence relation:

$$\lim_{p \to 0^+} \Vert x\Vert_p^p = card(x) \tag{9}$$

Mangasarian (1999) suggested the following approximation

$$f(x) = 1 - e^{-\frac{|x|}{t}},\ t \in (0,\ 1)$$

for cardinality function to seek for the solution of CMPs over a polyhedral set. Cui et al. (2019) replace the cardinality function with an increasing penalty function

$$f_\sigma(x) = \frac{\sigma|x|}{\sigma|x| + 1}$$

where the parameter $\sigma \in (0, +\infty)$ and it is a concave function for $x \in [0,\ +\infty)$. Table 1 presents some popular approximation functions of $card(x)$ that are reviewed in this section. A comparison of the behavior of these approximation functions is depicted in Fig. 1.

Other nonconvex relaxations and algorithms have been proposed to improve or outperform l_1—norm regularization such as capped l_1—norm (Zhang 2013), transformed l_1—norm (Dinh and Xin 2020), sorted l_1—norm (Zeng and Figueiredo 2014), the difference of the l_1—norm and l_2—norm (l_{1-2}) Yin et al. (2015), smoothly clipped absolute deviation (SCAD) (Mehranian et al. 2013), minimax-concave penalty (MCP) (Selesnick 2017; Sun et al. 2018). Moreover, some researchers (Gulpinar and An 2010; Gotoh et al. 2017) proposed to use the difference of two convex functions (DC functions) and the proximal gradient algorithm to solve the problems, include $card(x)$, Sun et al. (2019) applied DC function to propose an s-difference type regularization for sparse recovery problem, which is the difference of the penalty function $R(x)$ and its corresponding s-truncated function$R(x^s)$. Another class are heuristics algorithms

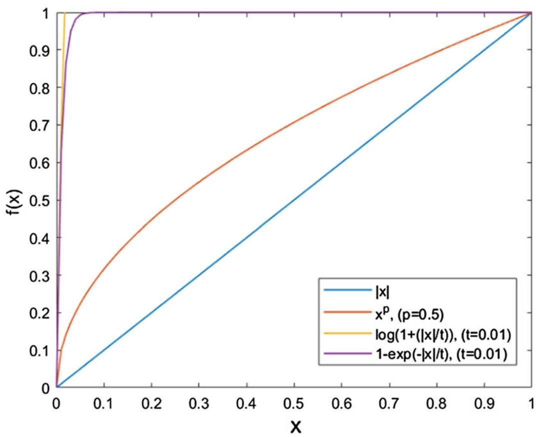

Fig. 1 Comparison of different approximations for cardinality function

such as matching pursuit methods (Dai and Milenkovic 2009; Needell and Tropp 2009) and thresholding methods (Donoho 1995; Voronin and Woerdeman 2013).

The contribution of this paper includes the following aspects. Initially, we propose a linear fractional function to approximate the cardinality function. Then replace the minimum cardinality problem with a sum-of-ratios problem. Secondly, we consider an efficient algorithm that transforms the sum-of-ratios problem to a parametric convex programming problem. A prominent feature of this algorithm lies in the convexity of the problem that is solved in each iteration. In this case, the approximation of the cardinality function is a linear fractional function. Finally, we consider two types of CMPs with inequality constraints and equality constraints and demonstrate their efficiency through a series of simulations.

The rest of this paper is organized as follows. Section 2 introduces the sum-of-ratios problem and studies an optimization algorithm that transforms the sum-of-ratios problem into the parametric convex programming problem and finds the global solution successfully. Section 3 presents a new approximation function and an algorithm to solve CMPs based on the previous algorithm. In Sect. 4, Our computational experiments demonstrates that the proposed technique is efficient for solving CMP, and finally, the conclusions are drawn in Sect. 5.

2 The sum-of-ratios problem

The sum-of-ratios problems (SORP), which minimize the sum of fractional functions over a convex set, is written as:

$$
\begin{aligned}
&\min F(x) = \sum_{i=1}^{n} \frac{f_i(x)}{h_i(x)} \\
&s.t. \\
&g_j(x) \le 0, \ j = 1, \dots, m \\
&x \in \mathbb{R}^n
\end{aligned} \tag{10}
$$

where $f_i(x)$, $g_j(x)$, $-h_i(x)$ for $(i = 1, \dots, n,\ j = 1, \dots, m)$ are convex functions. These problems are classified in the nonconvex optimization category. Unfortunately, the sum-of-ratios is more complicated than the single ratio problem; Freund and Jarre (Freund and Jarre 2001) proved that SOR problems are NP-hard. During the last decades, some algorithms are proposed for solving particular forms of the problem (10), several Branch and Bound approaches are mentioned when $f_i(x)$ and $h_i(x)$ are linear functions (Jiao and Liu 2015; Wang and Chu 2017). Shen and Lu (2018) presented a regional division and cut generation algorithm for solving (10) over a polyhedron, and Shen and Hung (2019) interpreted a range division and linearization algorithm for solving (10) where the objective is the sum or product of linear ratio. When $f_i(x)$ and $h_i(x)$ are quadratic functions, Jiao and Liu (2017) proposed a branch-and-bound algorithm for globally solve the quadratic sum-of-ratios fractional problem. If $f_i(x)$ and $h_i(x)$ are polynomial functions over a polyhedron, Kim et al. (2021) proposed the optimization algorithm for finding the globally optimal solution to (10), and Shen. et al. (2019) proposed an approach based on reducing the original nonconvex problem (10) as standard geometric programming (GP) problem by utilizing simple transformation and contraction strategies. Researchers are also mentioned to several schemes for nonlinear cases. For example, Gruzdeva and Strekalovsky (2017) reduced a general fractional programming problem with DC functions to solve an equation with a vector parameter.

As mentioned, Kim.et al. (2021) present an efficient optimization algorithm to the nonlinear sum-of-ratios problem, which it does not trap in local minima. This method transforms the sum-of-ratios problem into a parametric convex programming problem and finds the global solution successfully. It is assumed $f_i(x) \ge 0$ and $h_i(x) > 0$ $(\forall x \in X = \{x \in \mathbb{R}^n : g_j(x) \le 0,\ j = 1, \dots, m\})$ and.

$$
Int X = \{x \in \mathbb{R}^n : g_j(x) < 0,\ j = 1, \dots, m\} \neq \varnothing
$$

It is not difficult to see that the problem (10) is equivalent to

$$
\begin{aligned}
&\min F(x) = \sum_{i=1}^{n} \beta_i \\
&s.t. \\
&\frac{f_i(x)}{h_i(x)} \le \beta_i \quad i = 1, \dots, n \\
&g_j(x) \le 0 \quad j = 1, \dots, m,\ x \in \mathbb{R}^n
\end{aligned} \tag{11}
$$

Theorem (Kim et al. 2021): If $\left(\overline{x}, \overline{\beta}\right)$ is the solution of the problem (11), then there exist $\overline{\alpha}_i (i = 1, \ldots n)$ such that $\overline{x}$ is the solution of the following problem for $\alpha = \overline{\alpha}$ and $\beta = \overline{\beta}$.

$$
\begin{aligned}
&\min \sum_{i=1}^{n} \alpha_i (f_i(x) - \beta_i h_i(x)) \\
&s.t. \\
&x \in X
\end{aligned}
\tag{12}
$$

And also $\overline{x}$ satisfies the following system of equations for $\alpha = \overline{\alpha}$ and $\beta = \overline{\beta}$.

$$
\begin{aligned}
&\alpha_i = \frac{1}{h_i(x)}, \quad i = 1, \ldots, n \\
&f_i(x) - \beta_i h_i(x) = 0, \quad i = 1, \ldots, n
\end{aligned}
\tag{13}
$$

Thus, an optimization algorithm to find a global solution is constructed with the following steps based on the above consideration.

SOR Algorithm:

Step 0: Choose,$\varepsilon and \delta \in (0, 1)$ and let $\widehat{x} \in X$ as a starting point. Let

$$
\beta_i^0 = \frac{f_i(\widehat{x})}{h_i(\widehat{x})}, \alpha_i^0 = \frac{1}{h_i(\widehat{x})}, i = 1, \ldots, n, k = 0
$$

$$
\beta^k = \left(\beta_1^k, \ldots, \beta_n^k\right), \alpha^k = \left(\alpha_1^k, \ldots, \alpha_n^k\right)
$$

Step 1: Find a solution x^k of problem:

$$
\begin{aligned}
&\min \sum_{i=1}^{n} \alpha_i^k \left(f_i(x) - \beta_i^k h_i(x) \right) \\
&s.t. \\
&x \in X
\end{aligned}
$$

Step 2: If $\psi_{i'}\left(\vartheta^k\right) = 0\left(\forall i' = 1, \ldots, 2n\right)$,
where$\vartheta^k = \left(\beta^k, \alpha^k\right)$,

$$
\psi\left(\vartheta^k\right) = 0 \iff \begin{cases} \psi_{i'}\left(\vartheta^k\right) = -f_i\left(x^k\right) + \beta_i^k\left(h_i\left(x^k\right)\right) = 0, \quad i' = 1, \ldots, n, i = i', \\ \psi_{i'}\left(\vartheta^k\right) = -1 + \alpha_i^k\left(h_i\left(x^k\right)\right) = 0, \quad i' = n+1, \ldots, 2n, i = i' - n \end{cases}
$$

then x^k is a global solution and stop. Otherwise, let r_k denote the smallest integer among $r \in \{0, 1, 2, \ldots\}$ satisfying

$$\|\psi\left(\vartheta^{k+1}\right)\|_2 \leq \left(1 - \varepsilon\delta^{r_k}\right)\|\psi\left(\vartheta^k\right)\|_2$$

(Theorem 3 (Kim et al. 2021).

where $\lambda_k = \delta^{r_k}$,$\beta_i^{k+1} = \beta_i^k(1 - \lambda_k) + \lambda_k \frac{f_i(x^k)}{h_i(x^k)}$, $\alpha_i^{k+1} = \alpha_i^k(1 - \lambda_k) + \lambda_k \frac{1}{h_i(x^k)}$, $i = 1, \ldots, n$

Step 3: let $k = k + 1$ and go to *step1*.

A convex optimization problem is solved in each iteration of the SOR algorithm, and it has a global linear and local super-linear / quadratic rate of convergence (Theorem 3 (Kim et al. 2021)).

3 Solve CMPs with SOR algorithm

In this section, we consider the cardinality minimization problem (2) and derive a fractional approximation to the cardinality function, $card(x)$.

As mentioned before, (2) is NP-hard and the nonconvex cardinality function is a challenging issue for these problems. However, many attempts are made to provide a proper approximation of this function. Here, the following equivalence relation is considered:

$$\begin{aligned} &\lim_{t\to 0^+} \frac{x_i}{x_i + t} = 1 \quad if \quad x_i \neq 0,\ i = 1, \ldots, n \\ &\lim_{t\to 0^+} \frac{x_i}{x_i + t} = 0 \quad if \quad x_i = 0,\ i = 1, \ldots, n \end{aligned} \tag{14}$$

so, we get

$$card(x_i) \underset{t\to 0}{=} lim \frac{x_i}{x_i + t}$$

And

$$card(x) = \sum_{i=1}^{n} \lim_{t\to 0} \frac{x_i}{x_i + t} \tag{15}$$

The cardinality function and its fractional approximation for different values of t, $0 < t < 1$, is plotted in Fig. 2.

According to (12) and Fig. 2, we see that the cardinality function can be approximated by the linear fractional function for a given parameter $t > 0$,

$$card(x) \cong \sum_{i=1}^{n} \frac{x_i}{x_i + t} \tag{16}$$

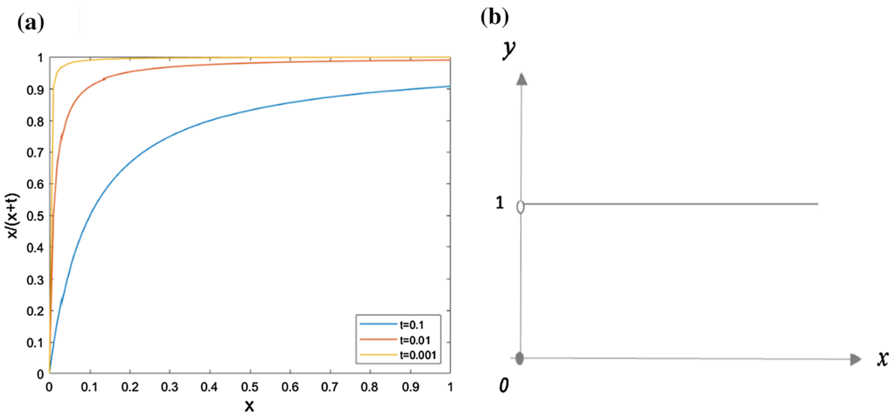

Fig. 2 **a** The graph of function $\frac{x}{x+t}$ for various values of $0 < t < 1$, **b** $y = \mathrm{card}(x),\ x \in R^1_{\geq 0}$

Therefore, Problem (2) can be approximately expressed as the following sum-of-ratios problem:

$$\min F(x) = \sum_{i=1}^{n} \frac{x_i}{x_i + t}$$
$$s.t.$$
$$x \in X \tag{17}$$

where $t > 0$.

As mentioned in Sect. 2, problem (17) is the sum of ratios problem in which $f_i(x) = x_i$ and $h_i(x) = x_i + t(i = 1, \ldots, n)$ are nonnegative continuous convex functions. It's easy to see that (17) can also be expressed as:

$$\min \sum_{i=1}^{n} \beta_i$$
$$s.t. \frac{x_i}{x_i + t} \leq \beta_i i = 1, \ldots, n$$
$$x \in X \tag{18}$$

Since the SOR algorithm for solving the fractional problem was examined in the previous section, we intend to use this algorithm for the problem (18) as a fractional problem. Therefore, we are rewriting the algorithm as follow:

Step 0: Choose,$\varepsilon and \delta \in (0, 1)$ and let $\widehat{x} \in X$ as a starting point.

Also, let

$$\beta_i^0 = \frac{\widehat{x_i}}{\widehat{x_i} + t},\ \alpha_i^0 = \frac{1}{\widehat{x_i} + t},\ i = 1, \ldots, n,\ k = 0$$

$$\beta^k = \left(\beta_1^k, \ldots, \beta_n^k\right),\ \alpha^k = \left(\alpha_1^k, \ldots, \alpha_n^k\right)$$

Step 1: Find a solution x^k of problem:

$$\min \sum_{i=1}^{n} \alpha_i^k \left(x_i - \beta_i^k (x_i + t) \right)$$

$$s.t.$$

$$x \in X$$

Step 2: If $\psi_{i'}\left(\vartheta^k\right) = 0\left(\forall i' = 1, \ldots, 2n\right)$,
where $\vartheta^k = \left(\beta^k, \alpha^k\right)$,

$$\psi\left(\vartheta^k\right) = 0 \iff \begin{cases} \psi_{i'}\left(\vartheta^k\right) = -x_i^k + \beta_i^k\left(x_i^k + t\right) = 0, & i' = 1, \ldots, n, i = i' \\ , \\ \psi_{i'}\left(\vartheta^k\right) = -1 + \alpha_i^k\left(x_i^k + t\right) = 0, & i' = n + 1, \ldots, 2n, i = i' - n \end{cases}$$

then x^k is a global solution to the problem (18), and so stop. Otherwise, let r_k denote the smallest integer among $r \in \{0, 1, 2, \ldots\}$ satisfying

$$\|\psi\left(\vartheta^{k+1}\right)\|_2 \leq \left(1 - \varepsilon\delta^{r_k}\right)\|\psi\left(\vartheta^k\right)\|_2$$

and , $\lambda_k = \delta^{r_k}$

$$\beta_i^{k+1} = \beta_i^k(1 - \lambda_k) + \lambda_k \frac{x_i^k}{x_i^k + t}, \; i = 1, \ldots, n$$

$$\alpha_i^{k+1} = \alpha_i^k(1 - \lambda_k) + \lambda_k \frac{1}{x_i^k + t}, \; i = 1, \ldots, n$$

Step 3: let $k = k + 1$ and go to *Step1*.

It is worth mentioning that the solution of problem (17) obtained by the SOR algorithm is an approximation and is a suboptimal solution for the problem (2).

In many algorithms, choosing an appropriate starting point has a considerable impact on results. An essential feature of the SOR algorithm is that it always gave a global solution with any starting point for the sum-of-ratios problem (Kim et al. 2021). Therefore, in solving the problem (17) with the SOR algorithm, selecting a starting point has almost no significant effect on the result. Its effect is negligible in a number of the mentioned instances. Thus, according to the numerical results, it was observed that the starting point could be generated randomly from $[0, 1]^n$.

According to (14), the provided approximation function is a successful approximation for the cardinality function when $t \to 0$. Theoretically, in problem (17), the suitable value for parameter t is the closest number to zero. Still, in practice, according to numerical results, it is observed that any $t > 0$ is acceptable, and the only difference is in the number of iterations. In some numerical experiments, if the parameter t is close to zero, the number of iterations is less. In Sect. 4, we take $t = 0.01$, and the SOR algorithm performs well.

As mentioned in Sect. 2, $f_i(x) \geq 0$ and $h_i(x) > 0 \forall i = 1, \dots, n$ are assumptions for the sum of ratio problem in the SOR algorithm. Therefore, for a CMP with real variables, the algorithm can be applied to the problem (3).

To illustrate the SOR algorithm's consequence to the cardinality minimization problem, let us consider two following examples.

Example 1

$$\begin{aligned}
&\min card(x)\\
&s.t.\\
&x_1 - x_2 + 2x_3 - x_5 - x_7 \geq 1\\
&2x_2 - x_3 - x_4 - 2x_5 + x_6 - x_8 \geq 0\\
&-x_2 + 2x_3 + x_4 + x_6 - x_8 \geq 1\\
&x_2 - x_3 + x_5 - 2x_7 \geq 0, \quad x_i \in [0, 1] \forall i = 1, \dots, 8
\end{aligned}$$

The optimal solution is $x^* = (1, 0, 0, 0, 0, 1, 0, 0)$ and starting point is $\widehat{x} = (0.8, 0, 0.2, 0, 0.2, 0.6, 0, 0)$ that is the solution of $l_1 - norm$ problem.

The fractional problem corresponding to the mentioned problem is:

$$\begin{aligned}
&\min \sum_{i=1}^{8} \frac{x_i}{x_i + t}\\
&s.t.\\
&x_1 - x_2 + 2x_3 - x_5 - x_7 \geq 1\\
&2x_2 - x_3 - x_4 - 2x_5 + x_6 - x_8 \geq 0\\
&-x_2 + 2x_3 + x_4 + x_6 - x_8 \geq 1\\
&x_2 - x_3 + x_5 - 2x_7 \geq 0\\
&0 \leq x_i \leq 1 \quad \forall i = 1, \dots, 8
\end{aligned}$$

The result of the numerical experiment with $t = 0.01$ is shown in the following table that the stopping criterion $||\psi^k||^2 < \varepsilon$ (where $\varepsilon = 1e - 7$) is used.

The results of the SOR algorithm for Example 1

Iteration (k)	x	$\|\psi^k\|_2 < \varepsilon$
0	(0.8, 0, 0.2, 0, 0.2, 0.6, 0, 0)	×
1	(1, 0, 0, 0, 0, 1, 0, 0)	×
2	(1, 0, 0, 0, 0, 1, 0, 0)	✓

Example 2 (Abdi 2013)

$$\begin{aligned}
&\mathrm{P1} : \min card(y)\\
&s.t.\\
&3y_1 - 2y_2 + 4y_3 - y_5 - y_6 = 0\\
&5y_1 - 2y_2 - 3y_3 + 5y_4 + 6y_6 = 5\\
&y_1 + 2y_2 - y_3 - 5y_4 - 6y_5 + 7y_6 = -5\\
&2y_1 - 3y_2 + 4y_4 - 6y_5 = 4\\
&y_i \in [-2, 2] \quad \forall i = 1, \ldots, 6
\end{aligned}$$

The optimal solution is $y^* = (0, 0, 0, 1, 0, 0)$ and the minimal $l_1 - norm$ solution is $y_{l_1} = (0, 0, -1.0078, 0.8532, -0.0978, -0.3327)$. In the beginning, the variables should be transferred in to $[-1, 1]$ by substituting $x_i = \frac{y_i}{2} (i = 1, \ldots, n)$ and next by introducing the variables w_i, v_i, u_i according to (3), the converted problem is

$$\begin{aligned}
&P2 : \min \sum_{i=1}^{6} \frac{w_i}{w_i + t}\\
&s.t.\\
&6(v_1 - u_1) - 4(v_2 - u_2) + 8(v_3 - u_3) - 2(v_5 - u_5) - 2(v_6 - u_6) = 0\\
&10(v_1 - u_1) - 4(v_2 - u_2) - 6(v_3 - u_3) + 10(v_4 - u_4) + 12(v_6 - u_6) = 5\\
&2(v_1 - u_1) + 4(v_2 - u_2) - 2(v_3 - u_3) - 10(v_4 - u_4) - 6(v_5 - u_5) + 14(v_6 - u_6) = -5\\
&4(v_1 - u_1) - 6(v_2 - u_2) + 4(v_4 - u_4) - 6(v_5 - u_5) = 2\\
&w_i \geq u_i, w_i \geq v_i, x_i = v_i - u_i \quad \forall i = 1, \ldots, 6\\
&w_i \geq 0, 0 \leq u_i \leq 1, 0 \leq v_i \leq 1, \quad \forall i = 1, \ldots, 6\\
&x_i \in [-1, 1] \qquad\qquad \forall i = 1, \ldots, 6
\end{aligned}$$

Let $\hat{w} = (0.1, 0.1, 0.1, 0.1, 0.1, 0.1)$ be the starting point and $t = 0.01$. After 2 iterations, the SOR algorithm find the solution $x_i^* = v_i^* - u_i^* = (0, 0, 0, 0.5, 0, 0)$ for $P2$ that is corresponding to the optimal solution, $y^* = (0, 0, 0, 1, 0, 0)$, for the original problem $P1$.

4 Numerical experiments

In this section, several numerical experiments are executed to evaluate the performance of the proposed algorithm. In test problems, which are generated based on the method expressed by Candès et al. (2008), a sparse vector x_0 is selected with s-sparsity level ($\|x_0\|_0 \leq s$) and the matrix $A_{m\times n}$ randomly generated with Normal distribution $N(0, 1)$ with zero mean and unit variance, then the vector b was put $b := Ax_0$. We consider two types of CMP problems with inequality and equality constraints. Each instance was repeatedly performed 10 times, and then the averaged results are presented in the following tables. The proposed algorithm is implemented in AIMMS

4.6. Also, CPLEX 12.8 was used as the MIP solver included in the AIMMS software. The experiments were carried out on a laptop with 2.0 GHz and 8 GB memory.

In the first study, we focus on the accuracy and efficiency of the proposed algorithm for the problem with feasible region$X = \{x \in [0, 1]^n : Ax \geq b\}$. We use the SOR algorithm to recover, respectively, the sparse vectors with $m = n/4,\ n = 10i,\ i = 1, 2, \ldots, 10$ and different sparsity levels $\|x_0\|_0 \leq s$, where s is considered in three levels$\{\lfloor m/3\rfloor, \lfloor m/4\rfloor, \lfloor m/5\rfloor\}$. In this table, the Success (%) shows the number of times (out of a 10) that the cardinality of the solution obtained by the SOR algorithm is equal to the cardinality of the original vector (Percent). The Gap (%) is defined as below:

$$\frac{|Incumbent - best\ bound|}{Incumbent} \times 100$$

where *Incumbent* is the objective value of the current integer solution found, and the *best bound* is the objective value of the solution obtained by the SOR algorithm. Here, the average Gap (%) is reported for each instance. The best solution is the best integer solution for the MIP problem, which is found in the specified time. The "optimal solution" is the optimal integer solution for the MIP problem. T_{SOR}, T_{MIP} are respectively the computation time of the SOR algorithm and MIP problem.

Table 2 shows that the SOR algorithm is completely successful for the tested problem with $s = \lfloor m/5\rfloor$ where the gap is zero. For $s = \lfloor m/4\rfloor$ and $s = \lfloor m/3\rfloor$ the algorithm, respectively, is successful in 77% and 34% of the examples. It is clear that the proposed algorithm has a better computation time compared to CPLEX as the MIP solver, and for $n \geq 400,$ this time is significantly shorter.

In the second study, we consider the test problems with $X = \{x \in [0, 1]^n : Ax \geq b\}$ and $(m, n, sparsity) = (256i, 1024i, 48i),\ i = 1, 2, \ldots, 10$. This data was proposed by Yuli et al. (2019). The results of the new algorithm are presented in Table 3.

As can be seen in Table 3, the "Gap (%)" column shows the accuracy of the solutions of the SOR algorithm for $n \leq 5120,$ which is obtained in less time against CPLEX. Also, as shown in this table, for the large size problem, i.e., $n \geq 6144$, the SOR algorithm can find a solution, but the solver could not find any feasible solution in the specified time.

In the next two experiments, CMP problems with equality constraints and free variables are considered, i.e., $X = \{x \in [-1, 1]^n : Ax = b\}$. This problem is used widely in compressed sensing (CS) with noiseless recovery sparse signals.

In the third study, the test problems with $= n/4$,$n = 10i,\ i = 1, 2, \ldots, 10$, and three levels $\{\lfloor m/3\rfloor, \lfloor m/4\rfloor, \lfloor m/5\rfloor\}$ are studied. We use the relative error (Rel-err) and successful rate (Suc-rat) to judge the algorithm's performance. Here, the definition of Rel-err (Bian and Chen 2020) of $\overline{x}$ concerning x^* is considered as:

$$Rel - err := \frac{\|\overline{x} - x^*\|_2}{\|x^*\|_2}$$

Table 2 Results of the SOR algorithm for random data

Instances description				SOR Algorithm			MIP-B&B		
Problem #	n	m	Sparsity(s)	T_{SOR} (sec)	Gap (%)	Success (%)	T_{MIP} (sec)	Best solution	Optimal solution
1	100	25	8	0.062	2.8	80	0.175		✓
2	100	25	6	0.044	0	100	0.088		✓
3	100	25	5	0.015	0	100	0.017		✓
4	200	50	16	0.096	6.7	60	10.83	✓	
5	200	50	12	0.086	0	100	0.447		✓
6	200	50	10	0.048	0	100	0.11		✓
7	300	75	25	0.154	3.1	60	138	✓	
8	300	75	18	0.121	0	100	61.3	✓	
9	300	75	15	0.092	0	100	2.47	✓	
10	400	100	33	0.212	5.4	20	900	✓	
11	400	100	25	0.154	0	100	900	✓	
12	400	100	20	0.131	0	100	184.7	✓	
13	500	125	41	0.318	6.6	20	1000	✓	
14	500	125	31	0.22	4	80	1000	✓	
15	500	125	25	0.162	0	100	437.35	✓	
16	600	150	50	0.596	7.38	20	1100	✓	
17	600	150	37	0.563	2.4	60	1100	✓	
18	600	150	30	0.217	0	100	1100	✓	
19	700	175	58	0.83	7.54	20	1200	✓	
20	700	175	43	0.720	2	60	1200	✓	
21	700	175	35	0.344	0	100	1200	✓	
22	800	200	66	1.144	7.1	20	1300	✓	
23	800	200	50	1.112	2.3	60	1300	✓	
24	800	200	40	0.97	0	100	1300	✓	
25	900	225	75	1.29	7.3	20	1400	✓	
26	900	225	56	1.354	6.7	60	1400	✓	
27	900	225	45	1.25	0	100	1400	✓	
28	1000	250	83	1.806	3.4	20	1500	✓	
29	1000	250	62	1.778	1.5	50	1500	✓	
30	1000	250	50	1.683	0	100	1500	✓	
Average				0.586	2.541		804.5	✓	

where, x^* is the original sparse vector. The running tests are regarded as successful if the relative error is smaller than 10^{-5}. As shown in Table 4, for sparsity levels, $s = \lfloor m/5 \rfloor$ and $s = \lfloor m/4 \rfloor$, the proposed algorithm is successful in all randomly generated problems and for $s = \lfloor m/3 \rfloor$ the success rate is 60%. Therefore, the SOR algorithm had good performance in suitable time.

Table 3 The results of the SOR algorithm for random data

Instances description				SOR algorithm				MIP-B&B	
number	n	m	Sparsity	$\|x_{SOR}\|_0$	T_{SOR}(s)	Gap (%)	Iteration	Best solution	T_{MIP}(s)
1	1024	256	48	31	1.06	0	10	31	1600
2	2048	512	96	69	10.8	0	10	69	2200
3	3072	768	144	125	56	0	15	125	2400
4	4096	1024	192	178	143	0	21	178	2600
5	5120	1280	240	195	224	0	10	541	2800
6	6144	1536	288	278	494	NA	21	NA	3000
7	7168	1792	336	282	873	NA	34	NA	3200
8	8192	2048	384	345	1530	NA	13	NA	3400
9	9216	2304	432	396	1957	NA	25	NA	3600
10	10,240	2560	480	403	2582	NA	28	NA	3800
Average					787.09				2860

NA: Not Available

Table 4 The numerical results of the SOR algorithm for the CMP problems with equality constraints with three sparsity levels

Instance #	n	m	Sparsity(s)	T_{SOR}	Iteration	Suc-rat (%)	Rel-err
1	100	25	8	0.228	2	100	1.5e−16
2	100	25	6	0.02	2	100	3.8e−16
3	100	25	5	0.022	2	100	1.2e−16
4	200	50	16	0.068	3	80	0.008
5	200	50	12	0.060	2	100	6e−16
6	200	50	10	0.053	2	100	6.6e−16
7	300	75	25	0.291	5	80	9e−3
8	300	75	18	0.184	3	100	1.8e−15
9	300	75	15	0.120	2	100	1e−15
10	400	100	33	0.642	4	60	0.48
11	400	100	25	0.515	4	100	3.2e−15
12	400	100	20	0.381	2	100	2e−15
13	500	125	41	1.584	5	60	0.066
14	500	125	31	0.720	2	100	2.3e−15
15	500	125	25	0.567	2	100	2.2e−15
16	600	150	50	5.012	11	40	0.118
17	600	150	37	1.637	3	100	5.8e−15
18	600	150	30	1.136	2	100	5.6e−15
19	700	175	58	8.898	9	60	0.08
20	700	175	43	2.589	2	100	7.1e−15
21	700	175	35	2.338	2	100	6e−15
22	800	200	66	16.49	15	80	0.048
23	800	200	50	5.788	3	100	9.5e−15
24	800	200	40	3.362	2	100	1.1e−14
25	900	225	75	26.81	19	20	0.24
26	900	225	56	5.710	2	100	2.5e−14
27	900	225	45	5.684	2	100	1e−14
28	1000	250	83	34.50	20	20	0.27
29	1000	250	62	9.370	3	100	1.3e−14
30	1000	250	50	8.300	2	100	1.1e−15
Average				4.731		86.7	0.036

In the next study, the test problems with $(m, n, sparsity) = (256i, 1024i, 48i)$, $i = 1, \ldots, 5$ considered that these data were proposed by Sun et al. (2019) for CMP under noiseless condition. Table 5 compares the mean and standard deviation of relative error for the mentioned algorithm and the s-difference regularization method

Table 5 Mean and standard deviation of relative error for the SOR algorithm and s-difference regularization method Sun et al. (2019)

Instances Description			SOR Algorithm		s-difference (l_1)		s-difference ($l_1 - l_2$)		s-difference (l_2)	
N	m	Sparsity	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
1024	256	48	1.13e−14	5.1e−15	1.55e−05	3.67e−11	1.55e−05	1.81e−10	1.56e−05	1.77e−10
2048	512	96	7.64 e−14	4.9e−14	1.32e−05	2.58e−12	1.28e−05	2.31e−12	1.29e−05	2.34e−12
3072	768	144	3.5e−13	3.5e−13	1.37e−05	2.12e−12	1.32e−05	1.89e−12	1.32e−05	1.9e−12
4096	1024	192	3.8 e−13	3.8e−13	1.44e−12	1.46e−12	1.25e−05	1.45e−12	1.26e−05	1.45e−12
5120	1280	240	7.2e−13	7.2e−13	1.32e−05	1.53e−12	1.29e−05	1.48e−12	1.3e−05	1.49e−12

by Sun et al. (2019), they focused on the following problem

$$\min \frac{1}{2}\|Ax - b\| + \rho P(x)$$

where ρ is a regularization parameter, and $P(x)$ is the Regularization function to penalize the sparsity of x, which can be decomposed into the difference of two convex functions (DC function). This DC function is considered as $P(x) = R(x) - R(x^s)$ that the experiments were performed for $\mathrm{R}(x) = \|x\|_1$, $R(x) = \|x\|_2$ and$R(x) = \|x\|_1 - \|x\|_2$.

In Table 5 shown the comparison of the SOR algorithm and s-difference regularization method (Sun et al. 2019), s-difference (l_1), s-difference ($l_1 - l_2$) and s-difference (l_2). As we can see the SOR algorithm has the best performance compared to the others, which show minimum mean and standard deviation of relative error.

5 Conclusion

This paper proposes a new algorithm to solve the CMP with linear constraints as an LP problem. This idea approximates the cardinality function with a linear fractional one that transfers the CMP to a sum-of-ratios problem. The proposed algorithm has efficiently solved the fractional programming problem as a parametric convex programming problem. The numerical experiments were conducted to evaluate the performance of the proposed algorithm in CMP problems demonstrate that the algorithm has been successful for 70.3% of the generated problem in Table 2 and 86.7% of the created problem in Table 4, and it is pretty successful for the problem with $n/4$ constraints and $s \leq \lfloor m/4 \rfloor$. In addition, from Tables 3 and 5, we can find that the algorithm leads to less error and provide a quite competitive or slightly superior performance compared with the s-difference regularization method under the noiseless conditions.

References

Abdi MJ (2013) Cardinality Optimization Problems. phd thesis,School of Mathematics The University of Birmingham

Beck A, Vaisbourd Y (2016) The sparse principal component analysis problem: optimality conditions and algorithms. J Optim Theory Appl 1:119–143

Bertsimas D, King A, Mazumder R (2016) Best subset selection via a modern optimization lens. Ann Stat 44(2):813–852

Bian W, Chen X (2020) A smoothing proximal gradient algorithm for nonsmooth convex regression with cardinality penalty. SIAM J Numer Anal 58(1):858–883

Bruckstein AM, Donoho DL, Elad M (2009) From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Rev 51(1):34–81

C. F. Wang and. X. Y. Chu, (2017) A new branch and bound method for solving sum of linear ratios problem. Int J Appl Math 47(3):276–281

Candès EJ, Wakin MB, Boyd SP (2008) Enhancing sparsity by reweighted l1 minimization. J Fourier Anal Appl 14:877–905

Chen SS, Donoho DL, Saunders MA (1998) Atomic Decomposition by Basis. SIAM J Sci Comput 20(1):33–61

Chen Z, Huang C, Lin S (2019) A new sparse representation framework for compressed sensing MRI. Knowl-Based Syst 188:1–10

Chen X, Zhou W (2014) Convergence of the reweighted ℓ 1 minimization algorithm for ℓ 2–ℓ p minimization. Comput Optim Appl 59:47–61

Cui A, Peng J, Li H, Wen M (2019) Nonconvex fraction function recovery sparse signal by convex optimization algorithm. J Latex Class Files 31(5):1626–1637

Dai W, Milenkovic O (2009) Subspace pursuit for compressive sensing signal reconstruction. IEEE Trans Inform Theory 55:2230–2249

Dinh T, Xin J (2020) Convergence of a Relaxed Variable Splitting Method forLearning Sparse Neural Networks via l1, l0, andTransformed-l1 Penalties. arXiv:1812.05719

Donoho DL (1995) De-noising by soft-thresholdinng. IEEE Trans Inform Theory 41:613–627

Donoho DL (2006) Compressed sensing. IEEE Trans Inf Theory 52(4):1289–1306

Freund RW, Jarre F (2001) Solving the sum-of-ratios problem by an interior-point method. J Global Optim 19:83–102

Ge D, Jiang X, Ye Y (2011) A note on the complexity of Lp minimization. Math Program 129:285–299

Gotoh J-Y, Takeda A, Tono K (2017) DC formulations and algorithms for sparse optimization problems. Math Program 169(1):141–176

Gruzdeva TV, Strekalovsky AS (2017) On solving the sum-of-ratios problem. Appl Math Comput 1–10

Gulpinar N, An LTH (2010) Robust investment strategies with discrete asset choice constraints using DC programming. J Math Program Oper Res 59:45–62

Jia P, Zhang M, Shen Y (2019) Hypergraph learning and reweighted ℓ1-norm minimization for hyperspectral unmixing. IEEE J Sel Top Appl Earth Observ Rem Sens 12(6):1898–1904

Jiao HW, Liu SY (2015) A practicable branch and bound algorithm for sum of linear ratios problem. Eur J Oper Res 243(3):723–730

Jiao H, Liu S (2017) An efficient algorithm for quadratic sum-of-ratios fractional programs problem. Numer Funct Anal Optim 38(11):1426–1445

Jokar S, Pfetsch M (2008) Exact and approximate sparse solutions of underdetermined linear equations. SIAM J Sci Comput 31:23–44

Kim Y, Jong Y, Yu J (2021) A parametric solution method for a generalized fractional programming problem. Indian J Pure Appl Math

Lai M-J, Wang J (2011) An unconstrained lq minimization with 0<q<=1 for sparse solution of underdetermined linear system. SIAM J Optim 21(1):82–101

Mangasarian O (1999) Minimum-support solutions of polyhedral concave programs. J Math Program Oper Res 45(1–4):149–162

Markovsky I, Huffel SV (2007) Overview of total least-squares methods. Signal Process 87(10):2283–2302

Mehranian A, Saligheh Rad H, Ay MR, Rahmim A (2013) Smoothly clipped absolute deviation (SCAD) regularization for compressed sensing MRI using an augmented Lagrangian scheme. In: IEEE nuclear science symposiwn and medical imaging conference record (NSS/MIC), pp 3646–3653

Natarajan BK (1995) Sparse approximate solutions to linear systems. SIAM J Comput 24(2):227–234

Needell D, Tropp JA (2009) Iterative signal recovery from incomplete and inaccurate samples. Appl Comput Harmon Anal 26:301–321

Selesnick I (2017) Sparse regularization via convex analysis. IEEE Trans Signal Process 65(17):4481–4494

Shen P, Huang B (2019) Range division and linearization algorithm for a class of linear ratios optimization problems. J Comput Appl Math 350:324–342

Shen PP, Lu T (2018) Regional division and reduction algorithm for minimizing the sum of linear fractional functions. J Inequal Appl 63:1

Shen P, Zhu Z, Chen X (2019) A practicable contraction approach for sum of the generalized polynomial ratios problem. Eur J Oper Res 278:36–48

Shi Z, Zhou H, Xia Y, Liu W, Wang Y (2020) Random noise attenuation of common offset gathers using iteratively reweighted ℓ2,1 norm minimization. IEEE Geosci Remote Sens Lett 1:1–5

Sun Y, Chen H, Tao J (2018) Sparse signal recovery via minimax-concave penalty and ℓ1-norm loss function. IET Signal Proc 12(9):1091–1098

Sun Y, Tan X, Li X, Lei L, kaung G (2019) Sparse optimization problem with s-difference regularization. Signal Process

Tibshirani R (1996) Regression Shrinkage and Selection via the Lasso. J R Stat Soc Ser B (methodological) 58(1):267–288

Voronin S, Woerdeman HJ (2013) A new iterative firm-thresholding algorithms for inverse problems with sparsity constraints. Appl Comput Harmonic Anal 35:151–164

Yin P, Lou Y, He Q, Xin J (2015) Minimization of L1–2 for compressed sensing. SIAM J Sci Comput 37(1):A536–A563

Zeng X, Figueiredo MAT (2014) Decreasing weighted sorted l1 regularization. IEEE Signal Process Lett 21(10):1240–1244

Zhang T (2013) Multi-stage convex relaxation for feature selection. Bernoulli Soc Math Stat Probab 19(58):2277–2293