



A novel algorithm for all normal parameter reductions of a soft set based on object weighting and integer partition

Banghe Han¹ · Ruize Wu² · Shengling Geng³ · Xinyu Nie⁴

Accepted: 5 October 2021 / Published online: 30 January 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

This paper aims to propose a novel algorithm for computing all normal parameter reductions of the soft set (#NPRS for short). Firstly, a weight vector is assigned to objects of the soft set domain. Then, a necessary condition for a normal parameter direction can be derived. A parameter subset is a solution only if the total value of the weighted sum of corresponding parameter approximations is a multiple of a constant number, which is equal to the sum of weights. Based on this necessary condition, we can figure out all potential solutions by using integer partition technique. It needs only to screen out the right ones at last. Experimental results are listed and compared when weight vectors are UNA, BIN, TER, OCT, DEC, DUO and HEX. Comparison results show that our method has a better performance for solving #NPRS.

Keywords Decision support · Soft set · Integer partition · Object weighting method · Normal parameter reduction

1 Introduction

0-1 valued information systems are common in our daily life and research work. It is generally represented in a matrix or a table, where the rows correspond to the objects in the domain, and the columns are for the attributes or parameters. For instances, the black and white picture, the adjacency matrix for a graph [1], and the vote results in multi-attribute

decision making when the parameters mean the experts, who are supposed to choose a subset of objects on their own opinions.

In 1999 Molodtsov brought in the concept of soft set [2], which provided a novel mathematical tool for dealing with uncertainties. A soft set can also be represented as a 0-1 valued information system [3]. Every column stands for a parameter approximation of the soft set. The decision making model is similar with that of multi-attribute decision making. The number of parameter approximations each object belongs to is defined as the choice value function of the soft set. An object is an optimal solution when it belongs to maximum number of parameter approximations. We refer to more details in Section 2.

Many works have been made by researchers on soft set theory and its potential applications [4–12]. Combined with different mathematical structures, soft set has been studied algebraically [13–18] and topologically [19–21]. Various types of vague concepts have been combined with soft set theory such as fuzzy soft sets [22–26, 33], soft rough sets [27–30].

Another important direction for soft set study is the parameter reduction problem of soft sets. Based on different conditions, several kinds of parameter reduction problems of soft sets or fuzzy soft sets [31, 32, 34–44] have been defined and discussed. The parameter reduction problems of soft sets actually deal with a kind of important structure for the parameter domain.

✉ Banghe Han
bhhan@mail.xidian.edu.cn

Ruize Wu
iruizewu@163.com

Shengling Geng
geng_sl@126.com

Xinyu Nie
niexinyu01@163.com

¹ School of Mathematics and Statistics, Xidian University, Xi'an, 710071, China

² School of Physics and Optoelectronic Engineering, Xidian University, Xi'an, Shaanxi 710071 China

³ College of Computer, Qinghai Normal University, Xining, 810008, China

⁴ School of Electronic Engineering, Xidian University, Xi'an, Shaanxi 710071 China

In this paper we will focus on a kind of parameter reduction problem, which is named as normal parameter reduction of soft sets. A normal parameter reduction is related with a parameter subset whose sum of rows are all equal. This means that once these parameters are deleted from the parameter domain, the same amount of choice values will be lost for every object, hence the same rank of the objects will be maintained.

Different algorithms have been proposed for computing the normal parameter reductions of a soft set in an efficient way. A kind of parameter subset whose sum of column values is an integral multiple of the number of rows (i.e., number of objects), was first given by [43]. This kind of parameter subset plays as a necessary condition in normal parameter reduction of soft set, i.e., the mentioned parameter subset is a candidate for normal parameter reduction of soft sets. Ma et al. [43] developed an algorithm for normal parameter reduction of soft set. Unnecessary repetition can be avoided because the sum of each column can be calculated in advance. In [45], a hierarchical algorithm for solving the minimal $k|U|$ parameter subset is studied, and then based on the minimal candidate set, the normal parameter reduction problem of soft set can be solved by testing the disjoint combinations of these minimal $k|U|$ parameter subsets.

Particle Swarm Optimization Algorithm was brought in to solve the normal parameter reductions in soft sets [36]. However, it can't guarantee that an optimal solution will be found [37]. Then a 0-1 linear programming method for finding an optimal normal parameter reduction of a soft set has been investigated in [38]. But the problem of finding all normal parameter solutions has not been investigated in [38].

Recently [44] brought the integer partition technique into the normal parameter reduction problem of soft sets. It was based on the necessary condition pointed out in [43]. Its main contribution is that an explicit method for computing the candidates (i.e., $k|U|$ parameter subsets) satisfying the necessary condition is proposed. However, there are two problems: (1) the number of candidates increases quickly when the size of soft sets grows; (2) the success rate of this method is quite low. Only a small proportion of the found candidates become the solutions.

The motivation of this paper is as follows: we should not be confined to the column sums. We notice that the column sum of a parameter approximation can be represented by each of the following expressions:

$$S_F(e) = \sum_{i=1}^n F(u_i, e), \quad (1.1)$$

$$S_F(e) = \sum_{i=1}^n (1 \times F(u_i, e)). \quad (1.2)$$

From expression (1.2) we see that there exists a weight vector $\omega = (1, 1, \dots, 1)$ for objects of the soft domain. So our idea is: can we define different weight vector for objects and achieve a potential better efficiency? For example, every parameter approximation can be represented as a 0-1 valued vector. So we try to connect the weight vector with Binary system and some other systems.

It's very interesting to investigate the efficiencies of different weight vectors. We find that the method of [44] can be improved.

The remainder of this paper is organized as follows. Section 2 introduces basic concepts such as soft set, and the problem of normal parameter reduction. The theoretical foundation of this paper is proposed in Section 3. Section 4 develops a novel algorithm for the mentioned problem. Experimental results showing the efficiency and success rate of our algorithm in normal parameter reduction problem are listed in Section 5. In Section 6 we make a short introduction to an APP, which is based on the algorithm and can be run as an application of our algorithm in MATLAB. Finally, we come to a conclusion of this article and outlook for potential future work.

2 Preliminaries

In this paper, suppose $U = \{u_1, u_2, \dots, u_n\}$ is a finite set of objects, E is a set of parameters. For example, the attributes in information systems can be taken as parameters. $\wp(U)$ means the power set of U , $|A|$ means the cardinality of set A . By [2] and [38] we have basic concepts about soft sets shown in Definitions 2.1 and 2.2.

Definition 2.1 (Soft set) A soft set on U is a pair $S = (F, A)$, where

- (i) A is a subset of E ;
- (ii) $F : A \rightarrow \wp(U)$, $\forall e \in A$, $F(e)$ means the subset of U corresponding with parameter e , and $F(e)$ is called e -approximation of S . We also use $F(u, e) = 1$ ($F(u, e) = 0$) to mean that u is (not) an element of $F(e)$.

Definition 2.2 (Support set of parameters for objects) Let $S = (F, A)$ be a soft set over U . $\forall u \in U$, define the support set of parameters for u as the set $\{e \in A | F(u, e) = 1\}$, denoted by $supp(u)$.

Definition 2.3 (Choice value function) Let $S = (F, A)$ be a soft set over U . The function $\sigma_S : U \rightarrow \mathbb{N}$ defined by $\sigma_S(u) = |supp(u)| = \sum_{e \in A} F(u, e)$ is called the choice value function of S .

We write σ_S as σ for short if the underlying soft set S is explicit.

Definition 2.4 Let $S = (F, A)$ be a soft set over U , $B \subseteq A$ define $S_F(e) = \sum_{u \in U} F(u, e)$; $S_F(B) = \sum_{e \in B} S_F(e)$.

Example 2.1 Table 1 represents a soft set $S = (F, E)$ over objects domain $U = \{u_1, u_2, \dots, u_6\}$ and parameters domain $E = \{e_1, e_2, \dots, e_7\}$, where $F(e_1) = \{u_1, u_4, u_5\}$, $F(e_2) = \{u_6\}$, $F(e_3) = \{u_1, u_2, u_4, u_5, u_6\}$, $F(e_4) = \{u_1, u_2, u_6\}$, $F(e_5) = \{u_1, u_2\}$, $F(e_6) = \{u_2, u_3, u_6\}$, $F(e_7) = \{u_2, u_3\}$, σ_S can be regarded as the choice value function of soft set S .

According to [38], we have the following concept about normal parameter reduction of soft sets.

Definition 2.5 (Normal parameter reduction) For soft set $S = (F, A)$ over U , $B \subseteq A$, $B \neq \emptyset$, if the constraint

$$\sum_{e \in A-B} F(u_1, e) = \dots = \sum_{e \in A-B} F(u_n, e) \tag{2.1}$$

is satisfied, then B is called a normal parameter reduction of S .

Example 2.2 Take the soft set in Example 2.1 for instance. It's easy to check that

$$\sum_{e \in \{e_1, e_2, e_7\}} F(u_1, e) = \dots = \sum_{e \in \{e_1, e_2, e_7\}} F(u_6, e) = 1,$$

thus $A - \{e_1, e_2, e_7\}$ is a normal parameter reduction.

Note that we don't require the minimality condition for normal parameter reductions as defined in [32].

Main Definition 2.1 ($\#$ NPRS) For soft set $S = (F, A)$ over U , compute the set of all normal parameter reductions of S , denoted by $\#$ NPRS.

Table 1 Tabular representation of a soft set $S = (F, A)$

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	σ_S
u_1	1	0	1	1	1	0	0	4
u_2	0	0	1	1	1	1	1	5
u_3	0	0	0	0	0	1	1	2
u_4	1	0	1	0	0	0	0	2
u_5	1	0	1	0	0	0	0	2
u_6	0	1	1	1	0	1	0	4
$S_F(e_i)$	3	1	5	3	2	3	2	$S_F(E) = 19$

3 Theoretical foundations

3.1 Weighting operation for objects in U of a soft set

Definition 3.1 Suppose $S = (F, A)$ is a soft set over $U = \{u_1, u_2, \dots, u_n\}$. $|U| = n$, $|A| = m$. $\omega = (\omega_1, \omega_2, \dots, \omega_n)$ is a weight vector with non-negative component ω_i for $u_i, i = 1, 2, \dots, n$. $\forall e_i \in A$, define $S_F^\omega(e_i)$ as follows:

$$S_F^\omega(e_i) = \sum_{k=1}^n \omega_k F(u_k, e_i). \tag{3.1}$$

Definition 3.2 Suppose $S = (F, A)$ is a soft set over U . $|U| = n$, $|A| = m$. $A_1 \subset A$. Define

$$S_F^\omega(A_1) = \sum_{e \in A_1} S_F^\omega(e). \tag{3.2}$$

Example 3.1 With respect to the soft set shown in Table 1, we define different weight vectors: (i) The weight vector ω in Table 2 is equal to $(1, 1, \dots, 1)$. In this situation we denote $S_F^\omega(e)$ as $S_F^{UNA}(e)$, and we see that $S_F^{UNA}(e)$ is actually equal to $S_F(e)$ defined in Definition 2.4.

- (ii) When a soft set is represented in a table as Example 2.1 does, every e -approximation $F(e)$ can be regarded as a 0 – 1 vector. Inspired by Binary, we define a weight vector $(2^0, 2^1, \dots, 2^5)$ for objects $u_i, i = 6, 5, \dots, 1$. Here we denote $S_F^\omega(e)$ by $S_F^{BLN}(e)$.
- (iii) Similarly, if we make use of Ternary, Octal, Decimal, Duodecimal, and Hexadecimal, we can define weight vectors $(3^0, 3^1, \dots, 3^5)$, $(8^0, 8^1, \dots, 8^5)$, $(10^0, 10^1, \dots, 10^5)$, $(12^0, 12^1, \dots, 12^5)$, $(16^0, 16^1, \dots, 16^5)$ to objects $u_i, i = 6, 5, \dots, 1$. See Table 3. We denote the function $S_F^\omega(e)$ by $S_F^{TER}(e_i)$, $S_F^{OCT}(e_i)$, $S_F^{DEC}(e_i)$, $S_F^{DUO}(e_i)$ and $S_F^{HEX}(e_i)$, respectively. See Table 4 for more details.

Proposition 3.1 Suppose $S = (F, A)$ is a soft set over U . $|U| = n$, $|A| = m$. As to the function $S_F^\omega(e_i)$, when

Table 2 Function $S_F^{UNA}(e_i)$ for the soft set $S = (F, A)$ in Table 1

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	ω_{UNA}
u_1	1	0	1	1	1	0	0	1
u_2	0	0	1	1	1	1	1	1
u_3	0	0	0	0	0	1	1	1
u_4	1	0	1	0	0	0	0	1
u_5	1	0	1	0	0	0	0	1
u_6	0	1	1	1	0	1	0	1
$S_F^{UNA}(e_i)$	3	1	5	3	2	3	2	$S_F^{UNA}(E) = 19$

Table 3 Different weighting vectors for the objects of soft set $S = (F, A)$ in Table 1

	ω_{UNA}	ω_{BIN}	ω_{TER}	ω_{OCT}	ω_{DEC}	ω_{DUO}	ω_{HEX}
u_1	1	2^5	3^5	8^5	10^5	12^5	16^5
u_2	1	2^4	3^4	8^4	10^4	12^4	16^4
u_3	1	2^3	3^3	8^3	10^3	12^3	16^3
u_4	1	2^2	3^2	8^2	10^2	12^2	16^2
u_5	1	2^1	3^1	8^1	10^1	12^1	16^1
u_6	1	2^0	3^0	8^0	10^0	12^0	16^0

$\omega = BIN, TER, OCT, DEC, DUO, HEX$ we have the following properties:

- (i) $F(e_i) = F(e_j)$ if and only if $S_F^\omega(e_i) = S_F^\omega(e_j)$.
- (ii) If $F(e_i) = \emptyset$, then $S_F^\omega(e_i) = 0$.
- (iii) If $F(e_i) = U$, then $S_F^\omega(e_i) = \sum_{i=1}^n \omega_i$.
- (iv) $\forall e \in A$, we have $0 \leq S_F^\omega(e_i) \leq \sum_{i=1}^n \omega_i$.
- (v) If $F(e_i) \subsetneq F(e_j)$, then $S_F^\omega(e_i) < S_F^\omega(e_j)$.
- (vi) If $S_F^\omega(e_i)$ and n are both given, then we can retrieve $F(e_i)$.

3.2 A new necessary condition for normal parameter reduction problems of soft set

Theorem 3.1 [43] (Necessary Condition I for normal parameter reduction of soft sets) *Given a soft set $S = (F, A)$ over $U, B \subseteq A, B \neq \emptyset$, if B is a normal parameter reduction of S by Definition 3.1, then*

$$\sum_{e \in A-B} S_F(e) = k|U|, \tag{N_{UNA}}$$

i.e.,

$$\sum_{e \in A-B} S_F(e) \text{ mod } |U| = 0. \tag{N_{UNA}}$$

Theorem 3.2 (Necessary Condition II for normal parameter reduction of soft sets) *Suppose $S = (F, A)$ is a soft set over $U = \{u_1, u_2, \dots, u_n\}$. $|U| = n, |A| = m. \omega =$*

Table 4 Function $S_F(e_i)$ for the soft set $S = (F, A)$ in Table 1

	e_1	e_2	e_3	e_4	e_5	e_6	e_7
$S_F^{UNA}(e_i)$	3	1	5	3	2	3	2
$S_F^{BIN}(e_i)$	38	1	55	49	48	25	24
$S_F^{TER}(e_i)$	255	1	337	325	324	103	102
$S_F^{OCT}(e_i)$	32840	1	36937	36865	36864	4609	4608
$S_F^{DEC}(e_i)$	100110	1	110111	110001	110000	11001	11000
$S_F^{DUO}(e_i)$	248988	1	269725	269569	269568	22465	22464
$S_F^{HEX}(e_i)$	1048848	1	1114385	1114113	1114112	69633	69632

$(\omega_1, \omega_2, \dots, \omega_n)$ is a real-valued weight vector with non-negative component ω_i for $u_i, i = 1, 2, \dots, n. B \subseteq A, B \neq \emptyset$. If B is a normal parameter reduction of S by Definition 3.1, then $\exists K \in \mathbb{Z}$ satisfying

$$\sum_{e \in A-B} S_F^\omega(e) = K \sum_{i=1}^n \omega_i, \tag{N_\omega}$$

i.e.,

$$\sum_{e \in A-B} S_F^\omega(e) \text{ mod } \sum_{i=1}^n \omega_i = 0. \tag{N_\omega}$$

Proof

$$\begin{aligned} \sum_{e \in A-B} S_F^\omega(e) &= \sum_{e \in A-B} \left(\sum_{i=1}^n F(u_i, e) \times \omega_i \right) \\ &= \sum_{i=1}^n \left(\sum_{e \in A-B} F(u_i, e) \times \omega_i \right) \\ &= \sum_{i=1}^n \left(\left(\sum_{e \in A-B} F(u_i, e) \right) \times \omega_i \right) \end{aligned}$$

Since B is a normal parameter reduction, we have $\forall i, \sum_{e \in A-B} F(u_i, e)$ is constant. Denote $K = \sum_{e \in A-B} F(u_i, e)$. Hence

$$\sum_{e \in A-B} S_F^\omega(e) = \sum_{i=1}^n K \times \omega_i = K \times \sum_{i=1}^n \omega_i. \tag{N_\omega}$$

□

According to Example 3.1 (i) we know that Theorem 3.1 is a special situation of Theorem 3.2 where $\omega = (1, 1, \dots, 1)$.

Corollary 3.1 (Necessary Condition II for normal parameter reduction of soft set under Binary weight vector) *Suppose $S = (F, A)$ is a soft set over $U = \{u_1, u_2, \dots, u_n\}$. $|U| = n, |A| = m. \omega = (2^{n-1}, 2^{n-2}, \dots, 2^0)$ is a weight*

vector. $B \subseteq A, B \neq \emptyset$. If B is a normal parameter reduction of S by Definition 3.1, then $\exists K \in \mathbb{Z}$ satisfying

$$\sum_{e \in A-B} S_F^{BIN}(e) = K(2^n - 1). \tag{N_{BIN}}$$

Example 3.2 Take the soft set in Example 2.1 where $|U| = 6$ for instance. By Example 2.2 $A = \{e_1, e_2, e_7\}$ is a normal parameter reduction. With respect to the matrix representation in Table 1, by the third row which is green colored in Table 4 we have

$$\sum_{e \in \{e_1, e_2, e_7\}} S_F^{BIN}(e) = 38 + 1 + 24 = 63 = 1 \times (2^{|U|} - 1).$$

It should be noticed that the order of objects doesn't influence our result shown in expression N_ω . In Table 5, after we change the order of objects in the matrix representation of soft set in Table 1, we still have

$$\sum_{e \in \{e_1, e_2, e_7\}} S_F^{BIN}(e) = 28 + 2 + 33 = 63 = 1 \times (2^{|U|} - 1).$$

Note 3.1 (An open problem on the relations among different weight vectors) Suppose $S = (F, A)$ is a soft set over $U = \{u_1, u_2, \dots, u_n\}$. $|U| = n, |A| = m. B \subseteq A$. Given two arbitrary weight vectors ω_1 and ω_2 . It's very interesting to consider the following question: is there any logical implication relation between the statements $(\sum_{e \in A-B} S_F^{\omega_1}(e)) \bmod (\sum_{i=1}^n \omega_1) = 0$ and $(\sum_{e \in A-B} S_F^{\omega_2}(e)) \bmod (\sum_{i=1}^n \omega_2) = 0$, and in which conditions? In this paper we will not investigate this problem thoroughly. However, we try to give some situations as follows:

- (i) Let $\omega_1 = (1, 1, \dots, 1), \omega_2 = (2^{n-1}, 2^{n-2}, \dots, 1)$. Then we have:

$$N_{UNA} \not\Rightarrow N_{BIN}.$$

Table 6 shows a soft set where $A = \{e_1, e_2, \dots, e_8\}$. The values $S_F^{UNA}(e)$ and $S_F^{BIN}(e)$ are listed. Note that $\sum_{i=1}^5 \omega_1(i) = 5$ and $\sum_{i=1}^5 \omega_2(i) = 2^5 - 1 = 31$.

Table 5 Function $S_F^{BIN}(e_i)$ for another representation of the soft set $S = (F, A)$ in Table 1

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	ω
u_3	0	0	0	0	0	1	1	2^5
u_4	1	0	1	0	0	0	0	2^4
u_5	1	0	1	0	0	0	0	2^3
u_1	1	0	1	1	1	0	0	2^2
u_6	0	1	1	1	0	1	0	2^1
u_2	0	0	1	1	1	1	1	2^0
$S_F^{BIN}(e_i)$	28	2	31	7	5	35	33	$S_F^{BIN}(E) = 141$

Table 6 Tabular representation of a soft set $S = (F, A)$ for Note 3.1 (i) and (ii)

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
u_1	1	1	1	1	1	0	1	0
u_2	0	0	0	1	1	1	0	0
u_3	0	1	0	1	0	1	1	0
u_4	1	0	1	1	0	0	1	1
u_5	1	0	1	0	1	0	1	1
$S_F^{UNA}(e)$	3	2	3	4	3	2	4	2
$S_F^{BIN}(e)$	19	20	19	30	25	12	23	3

In Table 7 each 0-1 row-vector means the characteristic function of a subset $A - B$ of A . For example, the first row-vector is (1 1 0 0 0 0 0 0), then the related $A - B$ is equal to $\{e_1, e_2\}$. For each yellow row-vector, the corresponding parameter subset $A - B$ satisfies N_{UNA} but does not satisfy N_{BIN} . Notice that for any green row-vector, the corresponding $A - B$ satisfies both N_{UNA} and N_{BIN} .

- (ii) Let $\omega_1 = (1, 1, \dots, 1), \omega_2 = (2^{n-1}, 2^{n-2}, \dots, 1)$. Then we have:

$$N_{BIN} \not\Rightarrow N_{UNA}$$

Similarly, in Table 8 for any yellow row-vector, the corresponding parameter subset $A - B$ satisfies N_{BIN} but does not satisfy N_{UNA} . Notice that for any green row-vector, the corresponding $A - B$ satisfies both N_{UNA} and N_{BIN} .

3.3 Equivalent classes of parameters set of a soft set

Definition 3.3 Suppose $S = (F, A)$ is a soft set over $U = \{u_1, u_2, \dots, u_n\}$. $|U| = n, |A| = m. \omega = (\omega_1, \omega_2, \dots, \omega_n)$ is a real-valued weight vector with non-negative component ω_i for $u_i, i = 1, 2, \dots, n$. Define a relation \mathcal{R}_ω on parameters set A as follows: $\forall e_i, e_j \in A,$

$$e_i \mathcal{R}_\omega e_j \text{ if and only if } S_F^\omega(e_i) = S_F^\omega(e_j). \tag{3.3}$$

Table 7 Tabular representation for examples of Note 3.1 (i)

e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	S_F^{UNA}	S_F^{BIN}
1	1	0	0	0	0	0	0	5	39
0	1	1	0	0	0	0	0	5	39
0	1	0	0	1	0	0	0	5	45
1	0	0	0	0	1	0	0	5	31
0	0	1	0	0	1	0	0	5	31
0	0	0	0	1	1	0	0	5	37
1	0	0	0	0	0	0	1	5	22
0	0	1	0	0	0	0	1	5	22
0	0	0	0	1	0	0	1	5	28

Table 8 Tabular representation for examples of Note 3.1 (ii)

e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	S_F^{BIN}	S_F^{UNA}
1	0	0	0	0	1	0	0	31	5
0	0	1	0	0	1	0	0	31	5
1	1	0	0	0	0	1	0	62	9
0	1	1	0	0	0	1	0	62	9
0	1	0	1	0	1	0	0	62	8
1	0	1	1	1	0	0	0	93	13
1	1	1	0	0	1	1	0	93	14
0	0	0	1	1	1	1	1	93	15

It is easy to get that \mathcal{R}_ω is an equivalence relation. So the following proposition holds:

Proposition 3.2 Suppose $S = (F, A)$ is a soft set over $U = \{u_1, u_2, \dots, u_n\}$. $|U| = n$, $|A| = m$. $\omega = (\omega_1, \omega_2, \dots, \omega_n)$ is a non-negative weight vector. Then when $\omega \in \{BIN, TER, OCT, DEC, DUO, HEX\}$, we have

$$e_i \mathcal{R}_\omega e_j \text{ if and only if } F(e_i) = F(e_j). \tag{3.4}$$

By using \mathcal{R} we can divide A into different equivalent classes. We use $[e_i]_{\mathcal{R}_\omega}$ to denote the equivalent class containing e_i , i.e., $[e_i]_{\mathcal{R}_\omega} = \{e \in A | e \mathcal{R}_\omega e_i\}$.

By Proposition 3.2 we know that the number of equivalent classes induced by \mathcal{R}_ω depends on the number of the same parameter approximations. In other words, the equivalence relation becomes more practical especially when we have more parameter approximations.

Example 3.3 Consider the soft set $S = (F, A)$ represented in Table 9, we can see $S_F^{BIN}(e_1) = 38$, $S_F^{BIN}(e_2) = 1$, $S_F^{BIN}(e_3) = 39$, $S_F^{BIN}(e_4) = 48$, $S_F^{BIN}(e_5) = 48$, $S_F^{BIN}(e_6) = 38$, $S_F^{BIN}(e_7) = 1$. So we have $[e_1]_{\mathcal{R}_{BIN}} = [e_6]_{\mathcal{R}_{BIN}} = \{e_1, e_6\}$, $[e_2]_{\mathcal{R}_{BIN}} = [e_7]_{\mathcal{R}_{BIN}} = \{e_2, e_7\}$, $[e_3]_{\mathcal{R}_{BIN}} = \{e_3\}$, $[e_4]_{\mathcal{R}_{BIN}} = [e_5]_{\mathcal{R}_{BIN}} = \{e_4, e_5\}$.

Table 9 Tabular representation of a soft set $S = (F, A)$

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	σ_S
u_1	1	0	1	1	1	1	0	5
u_2	0	0	0	1	1	0	0	2
u_3	0	0	0	0	0	0	0	0
u_4	1	0	1	0	0	1	0	3
u_5	1	0	1	0	0	1	0	3
u_6	0	1	1	0	0	0	1	3
$S_F^{BIN}(e_i)$	38	1	39	48	48	38	1	$S_F^{BIN}(E) = 213$

We use 2^U to denote the set $\{0, 1, 2, \dots, 2^{|U|} - 1\}$. For example if $|U| = 5$, then we have $2^U = \{0, 1, 2, \dots, 31\}$.

Definition 3.4 Suppose $S = (F, A)$ is a soft set over $U = \{u_1, u_2, \dots, u_n\}$. $|U| = n$, $|A| = m$. $\omega = (\omega_1, \omega_2, \dots, \omega_n)$ is a weight vector. Denote $W = \{0, 1, 2, \dots, \sum_i \omega_i\}$. Define a function $\mathcal{F}_\omega : W \rightarrow \{[e_i]_{\mathcal{R}_\omega} | e_i \in A\} \cup \{\emptyset\}$ by

$$\mathcal{F}_\omega(k) = [e_i]_{\mathcal{R}_\omega} \text{ if and only if } S_F^\omega(e_i) = k, \tag{3.5}$$

otherwise, $\mathcal{F}_\omega(k) = \emptyset$.

According to **Example 3.3**, we have

Example 3.4 Consider the soft set $S = (F, A)$ represented in Table 9, we can see $\mathcal{F}_{BIN}(1) = [e_2]_{\mathcal{R}_{BIN}} = \{e_2, e_7\}$, $\mathcal{F}_{BIN}(38) = [e_1]_{\mathcal{R}_{BIN}} = \{e_1, e_6\}$, $\mathcal{F}_{BIN}(39) = [e_3]_{\mathcal{R}_{BIN}} = \{e_3\}$, $\mathcal{F}_{BIN}(48) = [e_4]_{\mathcal{R}_{BIN}} = \{e_4, e_5\}$, $\mathcal{F}_{BIN}(6) = \emptyset$.

4 A novel algorithm for all normal parameter reductions of a soft set based on the necessary condition N_ω

Now it's time to show our algorithm for all normal parameter reductions of a soft set based on the necessary condition N_ω (see Fig. 1 for its sketch map).

Algorithm 1 A novel algorithm for all normal parameter reductions of a soft set based on the necessary condition N_ω (see Fig. 1 for its sketch map).

- 1: Input soft set $S = (F, A)$ over U , $|U| = n$, $|A| = m$.
- 2: Set up weight vector $\omega = (\omega_1, \omega_2, \dots, \omega_n)$ for objects u_i , and compute $S_F^\omega(e_i)$, $i = 1, 2, \dots, m$, \mathcal{R}_ω , $S_F^\omega(A)$.
- 3: Compute $\sum_i \omega_i$, get the sery of numbers to be partitioned: $k(\sum_i \omega_i)$, $k = 0, 1, 2, \dots, K$, where K is the maximum integer k satisfying $k(\sum_i \omega_i) \leq S_F^\omega(A)$.
- 4: Based on integer partition method, figure out all integer partitions (p_1, p_2, \dots, p_m) of the values $k(\sum_i \omega_i)$, $k = 0, 1, 2, \dots, K$, where $p_i \in \{0, 1\}$, and $\sum_{i=1}^m p_i S_F^\omega(e_i) \bmod \sum_i \omega_i = 0$, i.e., the condition N_ω holds.
- 5: Screen out all the normal parameter reductions B from the set of integer partitions (p_1, p_2, \dots, p_m) satisfying N_ω as follows: B is a normal parameter reduction if and only if there exists an integer partition (p_1, p_2, \dots, p_m) such that $A - B = \{e_i | p_i = 1\}$.

Notice that our weight vectors should not contain any negative coordinates. If the weight vector has negative coordinates, then it can't be guaranteed that the sum of the values while searching along a branch is monotone

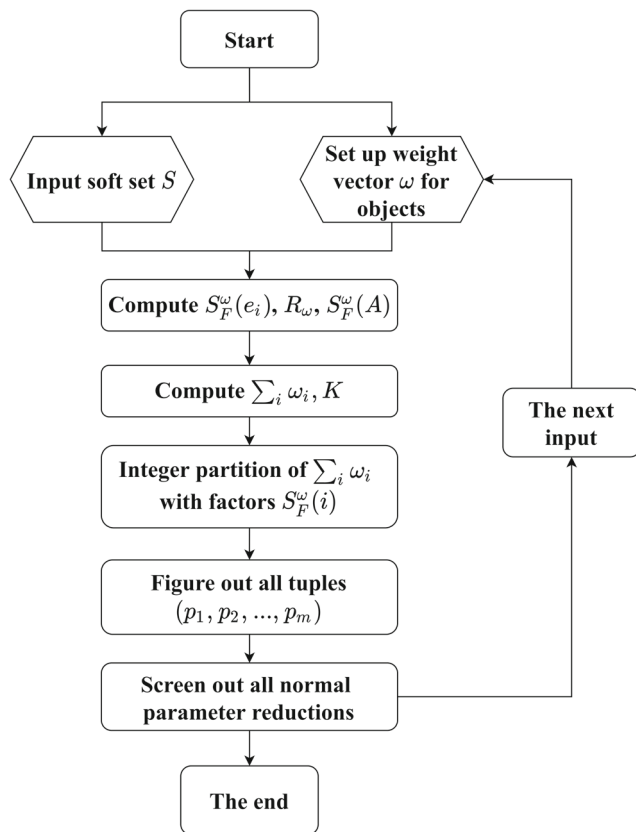


Fig. 1 A sketch map of Algorithm 1 for all normal parameter reductions

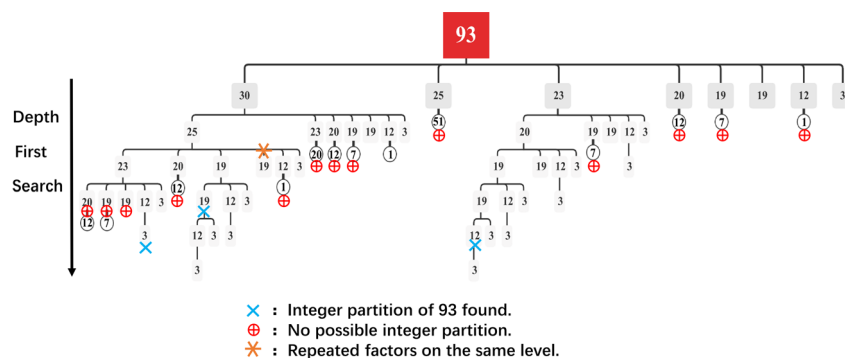
increasing. Thus our pruning method would become theoretically wrong (Fig. 2).

Example 4.1 Here we make an example with the soft set in Table 6.

Step 1. Input the soft set $S = (F, A)$ represented in Table 6.

Step 2. Set up $\omega = (2^4, 2^3, 2^2, 2^1, 2^0)$ by BIN , according to Table 6 we have $S_F^\omega(e_1) = 19, S_F^\omega(e_2) = 20, S_F^\omega(e_3) = 19, S_F^\omega(e_4) = 30, S_F^\omega(e_5) = 25, S_F^\omega(e_6) =$

Fig. 2 A sketch map of the integer partition technique in Algorithm 1



$12, S_F^\omega(e_7) = 23, S_F^\omega(e_8) = 3. S_F^\omega(A) = 151. \mathcal{R}_{BIN} = \{\{e_1, e_3\}, \{e_2\}, \{e_4\}, \{e_5\}, \{e_6\}, \{e_7\}, \{e_8\}\}.$

Step 3. $\sum_i \omega_i = 2^5 - 1 = 31$, get the series of numbers to be partitioned: $k(\sum_i \omega_i), k = 1, 2, 3, 4$ (0 can be deleted since $S_F^{\omega_{BIN}}(i) > 0, \forall i, 1, 2, \dots, 4$), and $K = 4$ since $K(\sum_i \omega_i) = 4 \times 31 \leq S_F^\omega(A) = 151, 5 \times 31 > 151$. So the series of numbers to be partitioned is 31, 62, 93, 124.

Step 4. With factors $\{S_F^\omega(i)\} = \{3, 12, 19, 20, 23, 25, 30\}$, it becomes a question to perform integer partition operation to integers $1 \times 31, 2 \times 31, 3 \times 31$ and 4×31 . Pay attention to the constraint that the times of factor $S_F^\omega(e_i)$ appears should not be bigger than $|[e]_{BIN}|$, where $S_F^\omega(e) = S_F^\omega(e_i)$. In Fig. 2, we give a sketch map for the integer partition process when $k(\sum_i \omega_i) = 93$. Then transform the integer partition results to tuples (p_1, p_2, \dots, p_m) . See Table 10 where $\omega = (2^4, 2^3, 2^2, 2^1, 2^0)$ for $u_i, i = 1, 2, \dots, 5$.

Step 5. Screen out all the normal parameter reductions B from the set of integer partitions (p_1, p_2, \dots, p_m) satisfying N_ω . See the tuples labeled with 1 in the last column of Table 10.

As a comparison, we also set up other kinds of ω . Similar with Table 10, we list the solutions when $\omega \in \{TER, OCT, DEC, DUO, HEX\}$ (See Table 11), $\omega = UNA$ (See Table 12).

For each $\omega \in \{TER, OCT, DEC, DUO, HEX\}$, the corresponding set of all integer partitions is the same. What's more, each of the three integer partitions relates to a normal parameter reduction B . That's 100 percent accuracy. When $\omega = UNA$, we have 45 integer partitions (See Table 12), and only three of them correspond to the normal parameter reduction solutions. The success rate is equal to 0.066. Hence as to this example, we can see that our new algorithm is much more efficient.

5 Experimental results for the proposed algorithm

In this section we introduce our experimental results of the proposed Algorithm 1. Particularly, when $\omega = UNA$, it

Table 10 The solutions of Example 4.1 when $\omega = BIN$

e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	S_F^{UNA}	S_F^{BIN}	Solutions (with label 1)
1	0	0	0	0	1	0	0	5	31	1
0	0	1	0	0	1	0	0	5	31	1
0	1	0	1	0	1	0	0	8	62	0
1	1	0	0	0	0	1	0	9	62	0
0	1	1	0	0	0	1	0	9	62	0
0	0	0	1	1	1	1	1	15	93	1
1	0	1	1	1	0	0	0	13	93	0
1	1	1	0	0	1	1	0	14	93	0

is actually the same with the method proposed in [43]. So we do give a comparison between the existing method by setting up $\omega = UNA$ and other kinds of weighting vectors.

Our experiments perform on a PC with an AMD Ryzen 5 3500U 2.10GHz CPU, 8GB RAM and the Win10 professional operating system. Our data is generated in the following way: first, we use the rand function of MATLAB to generate a uniformly distributed matrix of random numbers in the [0,1] interval. Numbers less than or equal to N in the matrix are then changed to 1, and the rest of the numbers are changed to 0. We can thus obtain a matrix with a ratio of 1 as N .

The controlling parameters in our experiments are as follows:

- (i) The number of rows, i.e., the number of objects $|U|$.
- (ii) The number of columns, i.e., the number of parameters $|A|$.
- (iii) The ratio of value 1, i.e., the proportion of 1 values of the generated data.

For every combination of the above controlling parameters, the running times of our algorithm are set up as 100. In the following subsections we list our experimental results with respect to the time efficiency, success rate, the rate of variance of these corresponding rows in the initial soft set, the average number of total integer partitions and the average number of normal parameter reductions.

5.1 Experimental results of the time cost

- (i) Let $|A| = 16, |U| = 8, 10, \dots, 16$. The ratio of 1 is equal to 0.3. The results of the time cost are shown

at the top-level of Fig. 3 (In Fig. 3, the subfigure on the right side of each level is the partial enlargement of the corresponding left one). With respect to $\omega = UNA$, the time cost decreases when $|U|$ increases. When $\omega = BIN, TER, OCT, DEC, DUO, HEX$, the time cost is much lower when compared with $\omega = UNA$.

- (ii) Let $|U| = 8, |A| = 10, 12, \dots, 20$. The ratio of 1 is equal to 0.3. The results of the time cost are shown at the middle-level of Fig. 3. With respect to $|A|$, the time cost increases when $|A|$ increases. When $\omega = BIN, TER, OCT, DEC, DUO, HEX$, the time cost is much lower than $\omega = UNA$ particularly when $|A|$ is larger than 14.
- (iii) Let $|U| = 8, |A| = 16$. The ratio of 1 is from 0.1 to 0.5. The results of the time cost are shown at the bottom-level of Fig. 3. The time cost increases when the ratio of 1 increases. When $\omega = BIN, TER, OCT, DEC, DUO, HEX$, the time cost is much lower than the situation when $\omega = UNA$.

5.2 Experimental results of the success rate

- (i) Let $|A| = 12, |U| = 8, 10, \dots, 14$. The ratio of 1 is equal to 0.3. The results of success rate are shown at the left subfigure of the top-level of Fig. 4. As far as to our experiments, we see that the success rate increases when ω changes from UNA to HEX (in the order of $BIN, TER, OCT, DEC, DUO, HEX$). The success rate is equal to 1 when $\omega = OCT, DEC, DUO, HEX$. When $|A| = 18$, we see

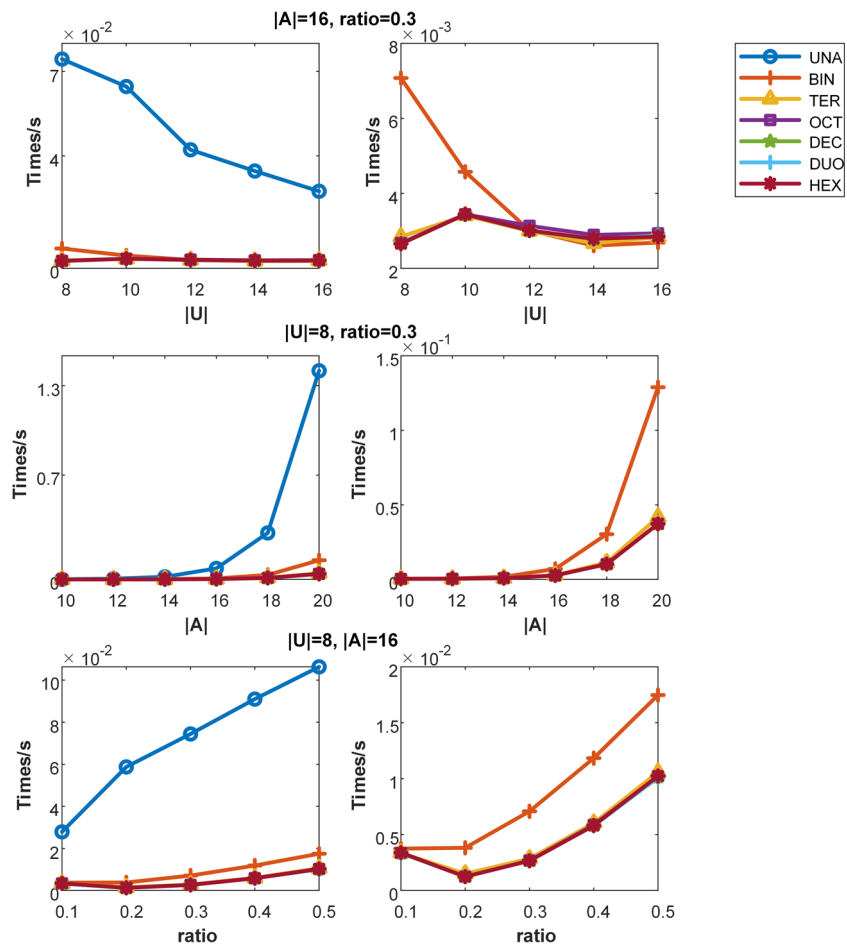
Table 11 The solutions of Example 4.1 when $\omega = TER, OCT, DEC, DUO, HEX$

e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	S_F^{TER}	S_F^{UNA}	Solutions (with label 1)
1	0	0	0	0	1	0	0	121	5	1
0	0	1	0	0	1	0	0	121	5	1
0	0	0	1	1	1	1	1	363	15	1

Table 12 The solutions of Example 4.1 when $\omega = UNA$

e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	S_F^{UNA}	Solutions (with label 1)
1	1	0	0	0	0	0	0	5	0
0	1	1	0	0	0	0	0	5	0
0	1	0	0	1	0	0	0	5	0
1	0	0	0	0	1	0	0	5	1
0	0	1	0	0	1	0	0	5	1
0	0	0	0	1	1	0	0	5	0
1	0	0	0	0	0	0	1	5	0
0	0	1	0	0	0	0	1	5	0
0	0	0	0	1	0	0	1	5	0
0	1	0	1	0	0	1	0	10	0
0	0	0	1	0	1	1	0	10	0
0	0	0	1	0	0	1	1	10	0
1	0	1	1	0	0	0	0	10	0
1	0	1	0	0	0	1	0	10	0
1	0	0	1	1	0	0	0	10	0
1	0	0	0	1	0	1	0	10	0
0	0	1	1	1	0	0	0	10	0
0	0	1	0	1	0	1	0	10	0
0	1	0	1	0	1	0	1	10	0
0	1	0	0	0	1	1	1	10	0
1	1	1	0	0	1	0	0	10	0
1	1	0	0	1	1	0	0	10	0
0	1	1	0	1	1	0	0	10	0
1	1	1	0	0	0	0	1	10	0
1	1	0	0	1	0	0	1	10	0
0	1	1	0	1	0	0	1	10	0
1	0	1	0	0	1	0	1	10	0
1	0	0	0	1	1	0	1	10	0
0	0	1	0	1	1	0	1	10	0
1	1	0	1	0	1	1	0	15	0
0	1	1	1	0	1	1	0	15	0
0	1	0	1	1	1	1	0	15	0
1	1	0	1	0	0	1	1	15	0
0	1	1	1	0	0	1	1	15	0
0	1	0	1	1	0	1	1	15	0
1	0	0	1	0	1	1	1	15	0
0	0	1	1	0	1	1	1	15	0
0	0	0	1	1	1	1	1	15	1
1	1	1	1	1	0	0	0	15	0
1	1	1	0	1	0	1	0	15	0
1	0	1	1	1	1	0	0	15	0
1	0	1	0	1	1	1	0	15	0
1	0	1	1	1	0	0	1	15	0
1	0	1	0	1	0	1	1	15	0
1	1	1	0	1	1	0	1	15	0

Fig. 3 The time efficiency of Algorithm 1



that the success rate is not equal to 1 when $\omega = HEX$.

- (ii) Let $|U| = 10, |A| = 10, \dots, 16$. The ratio of 1 is equal to 0.3. The results of success rate are shown at the left subfigure of the middle-level of Fig. 4. As far as to our experiments, we see that the success rate increases when ω changes from *UNA* to *HEX* (in the order of *BIN, TER, OCT, DEC, DUO, HEX*). The success rate is equal to 1 when $\omega = OCT, DEC, DUO, HEX$. When $|U| = 14$, we have similar results (see the right subfigure of the middle-level of Fig. 4). Take *TER* for example, we see that when $|A|$ increases, the rate of success decreases.
- (iii) Let $|U| = 8, |A| = 16$, the ratio of 1 is from 0.1 to 0.5. The results of success rate are shown at the left subfigure of the bottom-level of Fig. 4. As far as to our experiments, for each ratio of 1, we see that the success rate increases when ω changes from *UNA* to *HEX* (in the order of *BIN, TER, OCT, DEC, DUO, HEX*). The success rate is equal to 1 when $\omega = OCT, DEC, DUO, HEX$. Let $|U| = 14, |A| = 20$,

we see that the success rate is not equal to 1 when $\omega = HEX$ and the ratio of 1 is equal to 0.2 and 0.3 (see the right subfigure of the bottom-level of Fig. 4).

5.3 Experimental results: the rate of variance for these corresponding rows in initial soft set, the average number of total integer partitions and the average number of normal parameter reductions

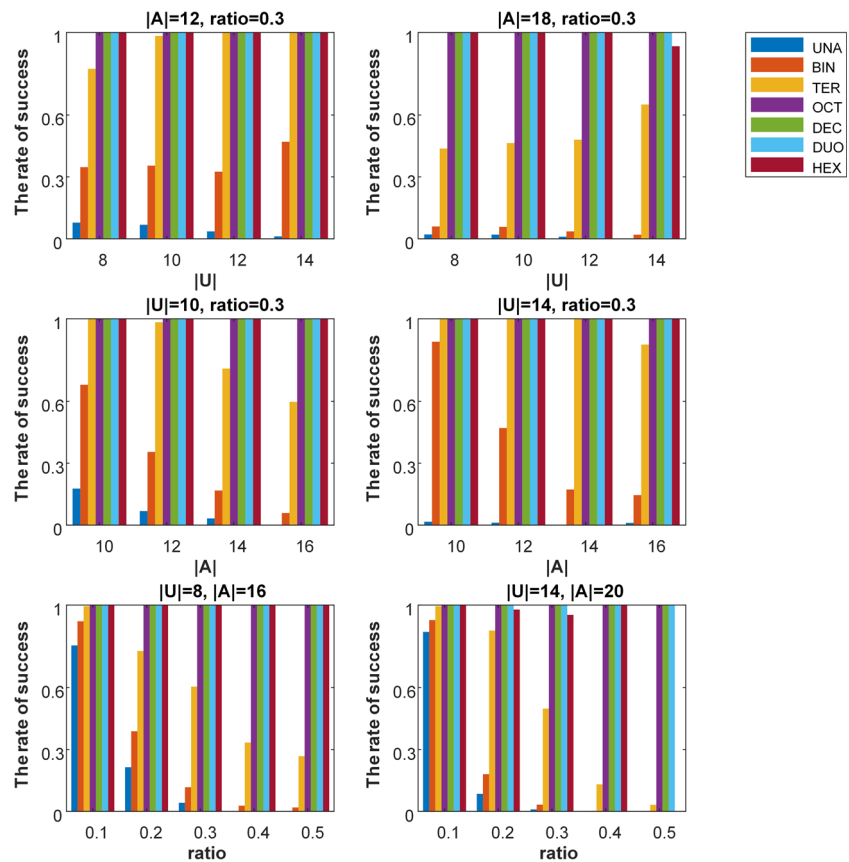
- (i) For an input soft set S , we denote the number of all normal parameter reductions as N_{NPRS} , and we also use N_{IPS}^ω to mean the number of all integer partitions with weight vector ω . Then we define

$$success\ rate_i = \frac{N_{NPRS}}{N_{IPS}^\omega}. \tag{5.1}$$

According to the experimental results $success\ rate_i, i = 1, 2, \dots, 100$, we have the average rate of success as follows:

$$\sum_{i=1}^{100} \frac{success\ rate_i}{100}. \tag{5.2}$$

Fig. 4 Experimental results of the success rate for **Algorithm 1**



See the experimental results in the fifth column of both Tables 13 and 14.

It should be noticed that when $N_{NPRS} = 0$ we always get $success\ rate_i = 0$. But it doesn't mean our algorithm is inefficient. This is because there doesn't exist any normal parameter reduction.

As far as our experiments, the rate of success becomes larger when weight vectors change from *UNA* to *HEX* in general. However, we couldn't prove it theoretically. We also get a counterexample in the last row of Table 14.

- (ii) Although not all the integer partitions are the normal parameter reductions of the input soft set, we can use statistical method to describe the extent in which it can be taken as a normal parameter reduction. In other words, we choose analysis of variance to characterize how far an integer partition is from a normal parameter reduction.

Suppose the j_{th} integer partition we get in the i_{th} experiment for the same size and the same weight vector is equal to (p_1, p_2, \dots, p_m) . Denote $V_{IP_S}^\omega(i, j) = var.(\sum_{p_j=1} F(u_1, e_j), \dots, \sum_{p_j=1} F(u_n, e_j))$, here $n = |U|$.

Assume the number of integer partition solutions we get in the i_{th} experiment for the same size and the same weight vector is equal to $N(i)$, then define

$$V_{IP_S}^\omega(i) = \frac{\sum_j V_{IP_S}^\omega(i, j)}{N(i)}, \tag{5.3}$$

So we can compute the average degree of variance by

$$\frac{\sum_{i=1}^{100} V_{IP_S}^\omega(i)}{100}. \tag{5.4}$$

See the experimental results in Tables 13 and 14. Generally, the larger the average rate of success is, the lower the average degree of variance is.

- (iii) The number of normal parameter reductions of a soft set is determined by its own data, and has nothing to do with our weight vectors. The last columns in Tables 13 and 14 give the experimental results in the average sense. Notice that the number of normal parameter reductions of a soft set is quite different when its size and the ratio of 1 are different.

Table 13 Experimental results for the proposed algorithm (to be continued)

$ U $	$ A $	Weight	Average time(s)	Rate of success	Average degree of variance	Average number of total integer partitions	Average number of normal parameter reduction
8	10	1	0.0018	0.16	0.87	66.93	1.64
8	10	2	0.00049	0.53	1.00	3.02	1.64
8	10	3	0.00039	0.93	1.23	1.73	1.64
8	10	8	0.00035	1.00	0.00	1.64	1.64
8	10	10	0.00038	1.00	0.00	1.64	1.64
8	10	12	0.00034	1.00	0.00	1.64	1.64
8	10	16	0.00036	1.00	0.00	1.64	1.64
12	10	1	0.00096	0.08	0.83	32.8	0.13
12	10	2	0.00016	0.65	0.98	0.21	0.13
12	10	3	0.00016	1.00	0.00	0.13	0.13
12	10	8	0.00014	1.00	0.00	0.13	0.13
12	10	10	0.00014	1.00	0.00	0.13	0.13
12	10	12	0.00015	1.00	0.00	0.13	0.13
12	10	16	0.00015	1.00	0.00	0.13	0.13
16	10	1	0.00084	0.03	0.84	23.18	0.04
16	10	2	0.00012	0.83	0.56	0.06	0.04
16	10	3	0.00011	1.00	0.00	0.04	0.04
16	10	8	0.00014	1.00	0.00	0.04	0.04
16	10	10	0.00011	1.00	0.00	0.04	0.04
16	10	12	0.00014	1.00	0.00	0.04	0.04
16	10	16	0.00011	1.00	0.00	0.04	0.04
8	14	1	0.02076	0.04	1.06	1352.05	5.18
8	14	2	0.00217	0.19	1.14	42.46	5.18
8	14	3	0.00103	0.76	1.31	6.91	5.18
8	14	8	0.00099	1.00	0.00	5.18	5.18
8	14	10	0.00098	1.00	0.00	5.18	5.18
8	14	12	0.00098	1.00	0.00	5.18	5.18
8	14	16	0.00104	1.00	0.00	5.18	5.18

5.4 Experiments compared with other algorithms designed for #NPR

Firstly, we need to collect the algorithms right designed for the #NPRS. Before this we list and analysis the references which are closely related to normal parameter reduction problems. See Table 15 where the reference numbers for this paper, their excellent idea and the corresponding contribution are given.

Based on the above preparation, we decide to choose the recent published algorithms from references [38], [43] and [44] as comparative ones with our method. As mentioned in the Table 15, [38] didn't propose an explicit algorithm for #NPRS, we need at first to revise the 0-1 linear method of optimal normal parameter reduction for suiting the #NPRS problems. For self-contained we cite an example in [38] here to show how we do this based on the work of [38].

Example 5.1 Table 16 represents a soft set $S = (F, E)$ over objects domain $U = \{u_1, u_2, \dots, u_6\}$ and parameters domain $E = \{e_1, e_2, \dots, e_8\}$. According to [38], we have the elements $D_{i \leftarrow i+1}$ and $D_{i+1 \leftarrow i}$, $i = 1, 2, \dots, 5$ in the matrix of dominant support parameters as shown in Table 17.

Then by [38] we have the set of constraints for a normal parameter reduction:

$$\begin{cases} C_{1,2} : e_2 = e_5 + e_6 \\ C_{2,3} : e_3 + e_7 + e_8 = 0 \\ C_{3,4} : e_5 + e_6 = e_2 + e_7 \\ C_{4,5} : 0 = 0 \\ C_{5,6} : e_2 = e_1 + e_3 + e_5 \end{cases}$$

Based on these constraints, we can give the revised algorithm for #NPRS by using implicit enumeration. We

Table 14 Experimental results for the proposed algorithm (the continuation of Table 13)

$ U $	$ A $	Weight	Average time(s)	Rate of success	Average degree of variance	Average number of total integer partitions	Average number of normal parameter reduction
12	14	1	0.01181	0.01	1.04	724.28	0.17
12	14	2	0.00103	0.10	1.11	2.05	0.17
12	14	3	0.00092	0.94	0.94	0.18	0.17
12	14	8	0.00107	1.00	0.00	0.17	0.17
12	14	10	0.00095	1.00	0.00	0.17	0.17
12	14	12	0.00096	1.00	0.00	0.17	0.17
12	14	16	0.00099	1.00	0.00	0.17	0.17
16	14	1	0.00803	0.00	0.99	424.1	0.02
16	14	2	0.00069	0.20	1.11	0.1	0.02
16	14	3	0.00071	1.00	0.00	0.02	0.02
16	14	8	0.00083	1.00	0.00	0.02	0.02
16	14	10	0.00083	1.00	0.00	0.02	0.02
16	14	12	0.00093	1.00	0.00	0.02	0.02
16	14	16	0.00077	0.67	0.58	0.03	0.02
8	18	1	0.32876	0.00	1.27	22485.25	23.87
8	18	2	0.02961	0.04	1.32	680.99	23.87
8	18	3	0.01057	0.35	1.42	63.5	23.87
8	18	8	0.00949	1.00	0.00	23.87	23.87
8	18	10	0.00969	1.00	0.00	23.87	23.87
8	18	12	0.00952	1.00	0.00	23.87	23.87
8	18	16	0.00958	1.00	0.00	23.87	23.87
12	18	1	0.17816	0.02	1.19	11847.37	0.76
12	18	2	0.01272	0.05	1.28	36.6	0.76
12	18	3	0.01232	0.58	1.26	1.31	0.76
12	18	8	0.01327	1.00	0.00	0.76	0.76
12	18	10	0.01292	1.00	0.00	0.76	0.76
12	18	12	0.01307	1.00	0.00	0.76	0.76
12	18	16	0.01303	1.00	0.00	0.76	0.76
16	18	1	0.14966	0.00	1.21	9092.54	0.06
16	18	2	0.01474	0.04	1.29	2.15	0.06
16	18	3	0.01659	0.57	1.09	0.09	0.06
16	18	8	0.01672	1.00	0.00	0.06	0.06
16	18	10	0.01686	1.00	0.00	0.06	0.06
16	18	12	0.01744	1.00	0.00	0.06	0.06
16	18	16	0.01715	0.80	1.01	0.07	0.06

Table 15 Diagram of references for #NPRS

Ref. No.	[32]	[43]	[36]	[37]	[38]	[44]
Author	Kong et al.	Ma et al.	Kong et al.	B.H. Han	Han et al.	Geng et al.
Pub. Year	2008	2011	2015	2016	2017	2020
Main Contribution	Definition with heuristic techniques	Column-sum oriented heuristic techniques	Particle swarm optimization algorithm	0-1 linear programming	0-1 linear programming	Integer partition
Suitability for #NPR	YES	YES	No	No	YES (after revisionment)	YES

Table 16 Tabular representation of a soft set $S = (F, A)$

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
u_1	0	1	1	1	0	0	1	1
u_2	0	0	1	1	1	1	1	1
u_3	0	0	0	1	1	1	0	0
u_4	0	1	0	1	0	0	1	0
u_5	0	1	0	1	0	0	1	0
u_6	1	0	1	1	1	0	1	0

choose to use the width-first searching method, with which we prune the searching branches once we find any contradiction with these constraints.

Algorithm 2 The Algorithm for #NPRS revised from [38].

Step 1: Compute the matrix of dominant support parameters.

Step 2: Construct the 0-1 linear constraints with the matrix generated in Step 1 by the method proposed in [38].

Step 3: Take the parameters as 0-1 valued variables. When $e = 1$, it means the parameter e is deleted. Perform width-first searching method, by cutting the branches once any contradiction is found with these constraints worked out in Step 2.

Step 4: Record and output all the satisfying parameters and the corresponding valuations.

With **Algorithm 2** we get all the normal parameter reductions $B_4 = A - \{e_4\}$, $B_2 = A - \{e_2, e_4, e_5\}$, $B_3 = A - \{e_1, e_2, e_4, e_6\}$, $B_1 = A$, $B_5 = A - \{e_1, e_2, e_6\}$, $B_6 = A - \{e_2, e_5\}$ for Example 5.1.

Figures 5 and 6 are sketch maps for comparison results. It is clearly implied that the proposed method in this paper by integer partition and object weighting is better than the mentioned methods above as far as to the data simulated in the experiments.

6 Operational system based on MATLAB

In order to show the potential application of our algorithm, we integrate our **Algorithm 1** into an APP, which can be installed in MATLAB 2019a or newer versions. Figure 7 is

Table 17 Dominant parameter subsets of adjacent objects for generating constraints in the linear programming models

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
$D_{i \leftarrow i+1}$	$\{e_2\}$	$\{e_3, e_7, e_8\}$	$\{e_5, e_6\}$	\emptyset	$\{e_2\}$
$D_{i+1 \leftarrow i}$	$\{e_5, e_6\}$	\emptyset	$\{e_2, e_7\}$	\emptyset	$\{e_1, e_3, e_5\}$

the interface after running an example. We would like to make a brief introduction to it as follows:

As can be seen from Fig. 7, the interface of this APP is divided into several functional parts and marked with serial numbers.

Area ① is where we input the data. The soft set data can be generated randomly. The controlling parameters for the size of soft set data and the ratio of 1 can be set up there. As far as to the example shown in Fig. 7, there are 5 objects in the soft set object domain and 7 parameters in the soft set parameter domain. And the ratio of 1 represents the proportion of 1 in the data, the number 0.5 in the figure means that 50% of the numbers in this soft set data matrix are equal to 1, which are randomly distributed in the matrix. The soft set data can be uploaded by the button marked with “Open excel”. The generated or uploaded data is shown in area ⑧.

In area ② we can choose or enter the weight vectors. As instances, we list UNA, BIN, TER, OCT, DEC, DUO, HEX over there.

Since we want to make comparison among different weight vectors, we list the current results in area ③, while the best result is kept in area ④. In each of these two areas we can see the running results such as the time cost, weight vector, the factors involved in integer partition process, the number of all integer partitions satisfying the condition N_ω . The number “3” in the local area labeled with “Success” means there are 3 partitions which correspond to the normal parameter reductions of the input soft set. The number “6” in the local area labeled with “All solutions” means there are 6 partitions in total. That’s why we see the Success rate is equal to 0.5 there. Area ⑨ contains all the integer partition results over there, and the solutions of the soft set are labeled with number 1 in the last column. All the maximal normal parameter reduction solutions can be easily got and are listed in area ⑩.

By button ⑤ we can start the computation. The final results can be restored in an excel form and downloaded by button ⑥. If we click button ⑦, we can clear the data for the next round.

7 Conclusion and future work

7.1 Aims fulfilled and novelty

The main contributions of this paper are as follows:

- (i) We have solved #NPRS problem of a soft set (i.e., all of the normal parameter reductions of a soft set). Weight vectors for objects in the soft set domain are firstly brought in. As a result, we come up to a new necessary condition for a normal parameter reduction

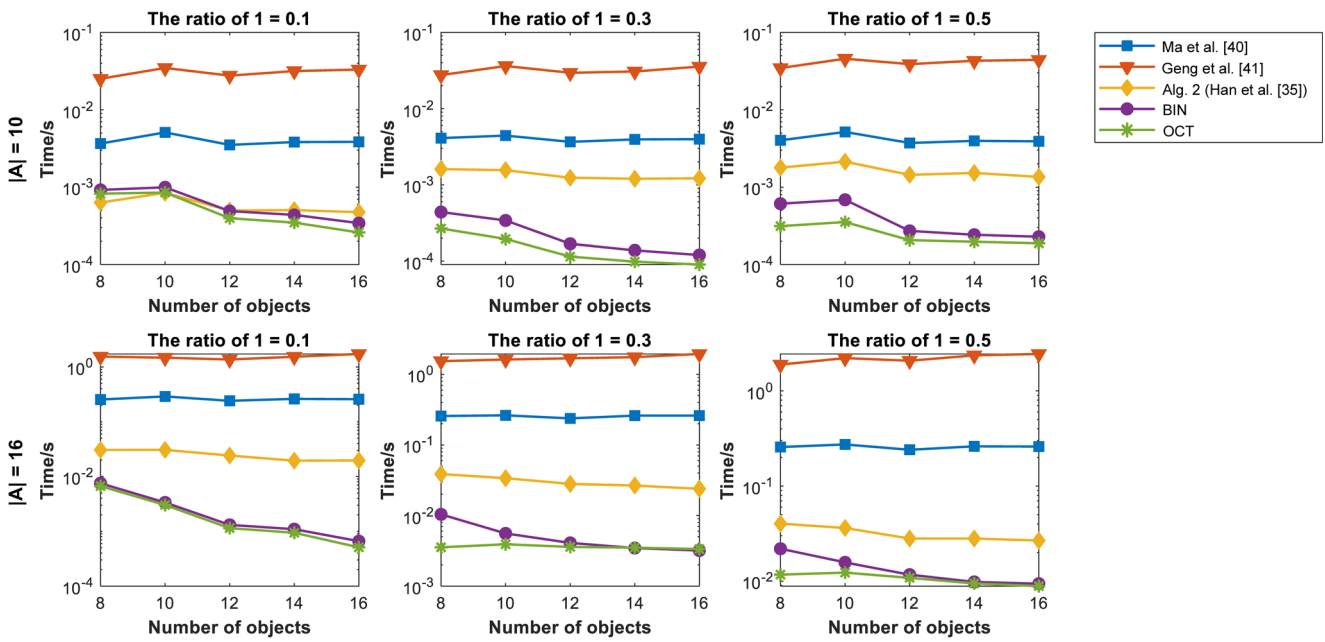


Fig. 5 The time efficiency comparison results when $|A| = 10$ and 16

N_ω . That's the principal theoretical foundation of our algorithm.

- (ii) Upon the derived necessary condition, the #NPRS problems can be solved by using integer partition technique. Different weight vectors result in different efficiency of the proposed algorithm. It is very

interesting to investigate these differences among different weight vectors. Experimental results have been achieved with respect to the time cost, success rate and the mean variance of partitions.

- (iii) All maximal solutions or optimal ones can be filtered out in sequence right after #NPRS.

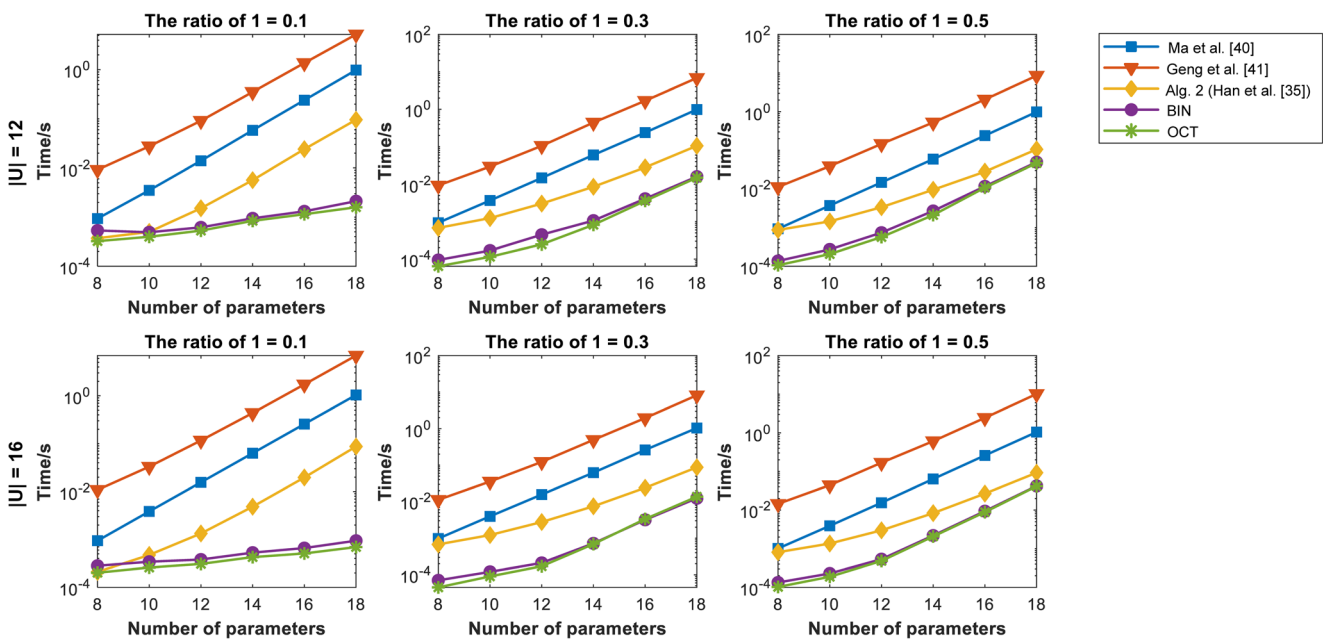


Fig. 6 The time efficiency comparison results when $|U| = 12$ and 16

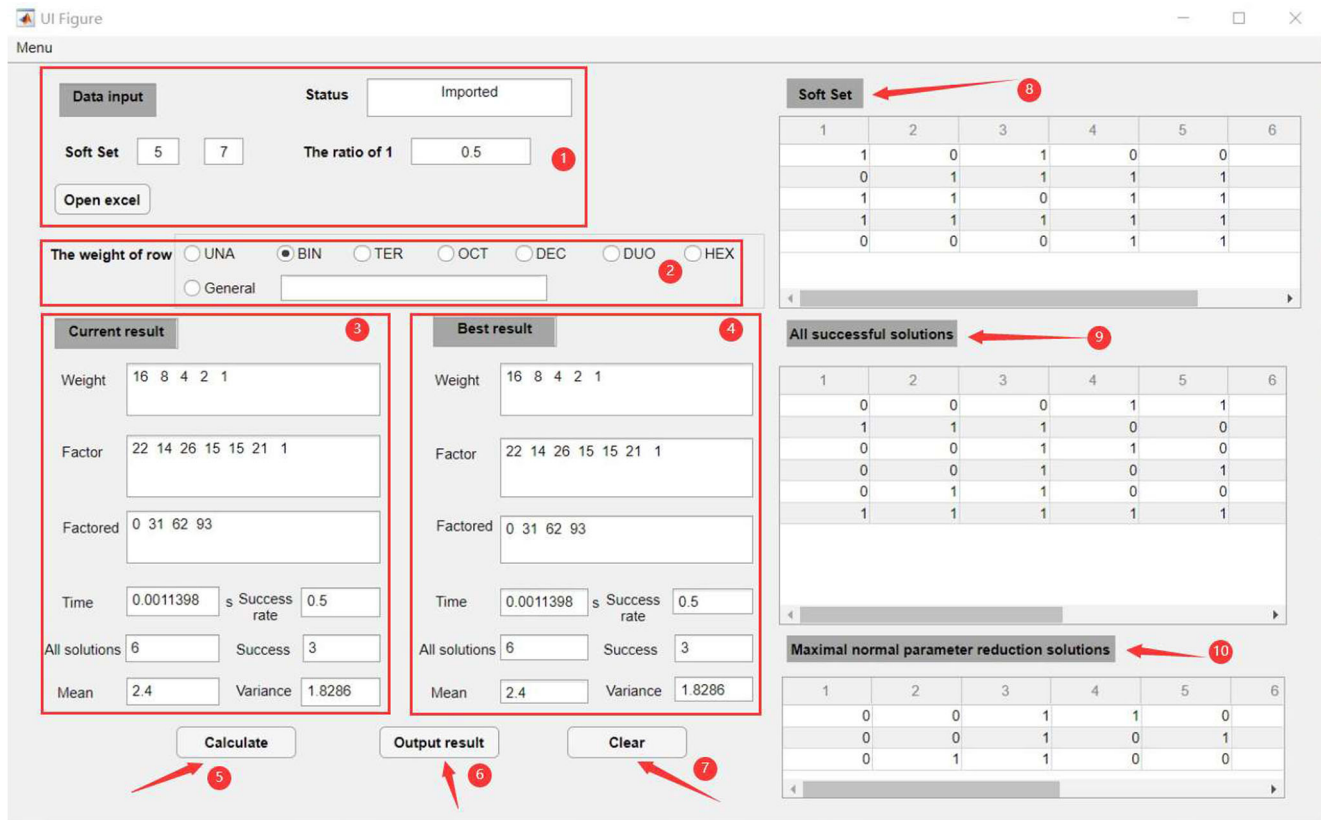


Fig. 7 The interface of our APP performing Algorithm 1 in MATLAB

7.2 Limitations of the proposed method

- (i) When $|A|$ grows, the proposed algorithm has to confront with larger workload for the integer partition. This is determined by the nature of the problem.
- (ii) Although from our experimental results we do have 100 percent success rate under some situations, this paper doesn't give the conditions under which N_ω becomes a sufficient condition for a normal parameter reduction theoretically.

7.3 A much more precise description for the idea of soft set and potential application of normal parameter reduction in data mining

7.3.1 Explanation of soft set as soft decision making scheme

Suppose $U = \{u_1, u_2, \dots, u_n\}$ is a finite set of objects. In this paper we assume that a decision making on the set U of objects means to choose a subset of U . A soft set $S = (F, A)$ over U defines a function from A to the power set of U , i.e., $\forall e \in A, F(e) \subseteq U$. Inspired by the assumption for what a decision making is, a soft set is actually a decision making strategy. That is "A decision making should be made upon different parameters or situations. Just like a soft

material which is easy to deform, a decision making should be flexible to different situations." As the saying goes, the benevolent see benevolence and the wise see the wisdom. Different people may have different opinions. So we can say that soft set is a kind of soft decision making scheme. Table 18 shows an example where different job hunters are searching for the same job. These hunters have to be judged by different interviewers or different criterions. It's normal that there may exist different decisions for who should be hired by different experts. For instance, job hunter 4 and job hunter 5 would be hired according to interviewer 1, while job hunter 3 and job hunter 4 should be hired by the opinion of interviewer 2.

Table 18 An example of soft set taken as a soft decision making scheme

	Interviewer 1	Interviewer 2	Interviewer 3	Interviewer 4
Job hunter 1	0	0	0	1
Job hunter 2	0	0	0	0
Job hunter 3	0	1	1	1
Job hunter 4	1	1	0	0
Job hunter 5	1	0	1	1

Table 19 Basic data for 2016 US presidential election

No. of State	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
States	AK	AL	AR	AZ	CA	CO	CT	DC	DE	FL	GA	HI	IA	ID	IL	IN	KS
Electoral votes	3	9	6	11	55	9	7	3	3	29	16	4	6	4	20	11	6
TRUMP	1	1	1	1	0	0	0	0	0	1	1	0	1	1	0	1	1
Hillary	0	0	0	0	1	1	1	1	1	0	0	1	0	0	1	0	0
No. of State	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
States	KY	LA	MA	MD	ME	MI	MN	MO	MS	MT	NC	ND	NE	NH	NJ	NM	NV
Electoral votes	8	8	11	10	4	16	10	10	6	3	15	3	5	4	14	5	6
TRUMP	1	1	0	0	0	1	0	1	1	1	1	1	1	1	0	0	0
Hillary	0	0	1	1	1	0	1	0	0	0	0	0	0	0	1	1	1
No. of State	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
States	NY	OH	OK	OR	PA	RI	SC	SD	TN	TX	UT	VA	VT	WA	WI	WV	WY
Electoral votes	29	18	7	7	20	4	9	3	11	38	6	13	3	12	10	5	3
TRUMP	0	1	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1
Hillary	1	0	0	1	0	1	0	0	0	0	0	1	1	1	0	0	0

7.3.2 Potential application of normal parameter reduction in data mining

In big data era it becomes much more important to mine useful and valuable information from data generated in various ways. Our normal parameter reduction idea and skills can be applied as an useful model for data mining. To be more precise, here we take the 2016 US Presidential Election as an example. Table 19 shows basic information [46], where we can see which states TRUMP won or Hillary did. The numbers of electoral votes for each state are also listed. In this paper we make an assumption that once a candidate take an advantage in popular vote count of a state, he would win all the electoral votes. We make notations as follows: denote the set of objects as $U = \{TRUMP, Hillary\}$, parameter set $E = \{state_1, state_2, \dots, state_{51}\}$ (i.e., $E = \{AK, AL, \dots, WY\}$), the number of electoral votes for each state (the third row in Table 19) is modeled as the weight $\omega(e)$ for each parameter e . Now we get a weighted soft set $S = (F, E)$, where for each parameter e in E , $F(e)$ is equal to single-element set containing the candidate who won in state e . So now for a subset B of E , $E - B$ is called

Table 20 Subset B of states that TRUMP and Hillary are completely evenly matched over $E - B$

No. of State	1	2	3	4	10	11	13
States	AK	AL	AR	AZ	FL	GA	IA
Electoral votes	3	9	6	11	29	16	6
TRUMP	1	1	1	1	1	1	1
Hillary	0	0	0	0	0	0	0

a normal parameter reduction if and only if TRUMP and Hillary are completely evenly matched over B , i.e.,

$$\sum_{e \in B} \omega(e)F(TRUMP, e) = \sum_{e \in B} \omega(e)F(Hillary, e),$$

where $F(TRUMP, e) = 1$ means TRUMP won in the state e , similarly for $F(Hillary, e) = 1$. We figure out a normal parameter reduction $E - B$, where B is shown in Table 20 as follows. Table 20 tells us that the mainly reason for the lose of Hillary lies in the subset of states $\{1, 2, 3, 4, 10, 11, 13\}$ (i.e., $\{AK, AL, AR, AZ, FL, GA, IA\}$). Because under our assumption the two candidates would get the same number of electoral votes which is equal to 229.

7.4 Future work

- (i) A useful way for solving the problem mentioned in Section 7.2 (i) is to divide the parameter set into different parts. Then the proposed algorithm can be solved in an approximate way.
- (ii) What if we have weighted parameters, how can we extend our method for the normal parameter reduction problems in such a situation.
- (iii) It is very interesting to combine our theory of 0-1 valued information systems with other kinds of information systems, such as hypergraphs.
- (iv) There are some intelligent methods which could be useful for improving the efficiency of integer factorization. We will come into the potential combination with our algorithms.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Grant No. 61862055). And this work is also funded by the Fundamental Research Funds of the Central Universities for period 2017.01-2020.12 (Grant No. JB180712).

References

- Biggs N, Lloyd E, Wilson R (1986) Graph theory. Oxford University Press, Oxford
- Molodtsov D (1999) Soft set theory-First results. *Comput Math Appl* 37(4-5):19–31. [https://doi.org/10.1016/S0898-1221\(99\)00056-5](https://doi.org/10.1016/S0898-1221(99)00056-5)
- Pei D, Miao D (2005) From soft sets to information systems. In: *Proceeding of 2005 IEEE international conference on granular computing, IEEE GrC05*. IEEE Press, Piscataway, pp 617–621
- Feng F, Cho J, Pedrycz W, Fujita H, Herawan T (2016) Soft set based association rule mining. *Knowl-Based Syst* 111(1):268–282. <https://doi.org/10.1016/j.knosys.2016.08.020>
- Sun B, Ma W, Li X (2017) Linguistic value soft set-based approach to multiple criteria group decision-making. *Appl Soft Comput* 58:285–296. <https://doi.org/10.1016/j.asoc.2017.03.033>
- Han BH, Li YM, Liu J, Geng SL, Hou HY (2014) Elicitation criterions for restricted intersection of two incomplete soft sets. *Knowl-Based Syst* 59:121–131. <https://doi.org/10.1016/j.knosys.2014.01.015>
- Muhammad A, Arooj A, Jos Carlos RA (2019) Group decision-making methods based on hesitant N-soft sets. *Expert Syst Appl* 115:95–105. <https://doi.org/10.1016/j.eswa.2018.07.060>
- Hu J, Pan L, Yang Y, Chen H (2019) A group medical diagnosis model based on intuitionistic fuzzy soft sets. *Appl Soft Comput J* 77:453–466. <https://doi.org/10.1016/j.asoc.2019.01.041>
- Gong K, Wang P, Xiao Z (2013) Bijective soft set decision system based parameters reduction under fuzzy environments. *Appl Math Model* 37(6):4474–4485. <https://doi.org/10.1016/j.apm.2012.09.067>
- Gong K, Wang Y, Xu MZ, Xiao Z (2018) BSSReduce an $O(|U|)$ incremental feature selection approach for large-scale and high-dimensional data. *IEEE Trans Fuzzy Syst* 26(6):3356–3367. <https://doi.org/10.1109/TFUZZ.2018.2825308>
- Yang J, Yao Y (2020) Semantics of soft sets and three-way decision with soft sets. *Knowl-Based Syst* :194. <https://doi.org/10.1016/j.knosys.2020.105538>
- Jun YB, Lee KJ, Khan A (2010) Soft ordered semigroups. *Math Logic Q* 56:42–50. <https://doi.org/10.1002/malq.200810030>
- Jiang YC (2010) Extending soft sets with description logics. *Comput Math Appl* 59:2087–2096. <https://doi.org/10.1016/j.camwa.2009.12.014>
- Liu WF (2013) Soft Boolean algebra. *J Shandong Univ* 48(8):56–62. <https://doi.org/10.6040/j.issn.1671-9352.0.2014.394>
- Cai XB, Liu TL (2013) The Duales of soft sets, soft groups and soft BCI/BCK-algebras. *Fuzzy Syst Math* 27(1):20–27. <https://doi.org/10.3969/j.issn.1001-7402.2013.01.004>
- Sun QM, Zhang ZL, Liu J (2008) Soft sets and soft modules. *Lecture Notes in Comput. Sci.* 5009. https://doi.org/10.1007/978-3-540-79721-0_56
- Feng F, Jun YB, Zhao X (2008) Soft semirings. *Comput Math Appl* 56:2621–2628. <https://doi.org/10.1016/j.camwa.2008.05.011>
- Qin KY, Kong ZY (2010) On soft equality. *J Comput Appl Math* 234(5):1347–1355. <https://doi.org/10.1016/j.cam.2010.02.028>
- Shabir M, Naz M (2011) On soft topological spaces. *Comput Math Appl* 61:1786–1799. <https://doi.org/10.1016/j.camwa.2011.02.006>
- Tanay B, Kandemir MB (2011) Topological structure of fuzzy soft sets. *Comput Math Appl* 61:412–418. <https://doi.org/10.1016/j.camwa.2011.03.056>
- Al-shami TM, El-Shafei ME, Abo-Elhamayel M (2019) On soft topological ordered spaces. *J King Saud Univ Sci* 31:556–566. <https://doi.org/10.1080/25765299.2019.1664101>
- Zadeh LA (1965) Fuzzy sets. *Inf Control* 8(3):338–353. [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
- Maji PR, Biswas R, Roy AR (2001) Fuzzy soft sets. *J Fuzzy Math* 9(3):589–602
- Yang XB, Lin TY, Yang JY, Li Y, Yu DJ (2009) Combination of interval-valued fuzzy set and soft set. *Comput Math Appl* 58(3):521–527. <https://doi.org/10.1016/j.camwa.2009.04.019>
- Deli I, Cagman N (2015) Intuitionistic fuzzy parameterized soft set theory and its decision making. *Appl Soft Comput* 28(3):109–113. <https://doi.org/10.3233/IFS-151920>
- Vijayabalaji S, Ramesh A (2018) Belief interval-valued soft set. *Expert Syst Appl* 119:262–271. <https://doi.org/10.1016/j.eswa.2018.10.054>
- Pawlak Z (1982) Rough sets. *Int J Comput Inf Sci* 11(5):341–356. <https://doi.org/10.1007/BF01001956>
- Feng F, Liu XY, Leoreanu-Fotea V, Jun YB (2011) Soft sets and soft rough sets. *Inform Sci* 181(6):1125–1137. <https://doi.org/10.1016/j.ins.2010.11.004>
- Zhan J, Ali MI, Mehmood N (2017) On a novel uncertain soft set model: Z-soft fuzzy rough set model and corresponding decision making methods. *Appl Soft Comput* 56:446–457. <https://doi.org/10.1016/j.asoc.2017.03.038>
- Akram M, Ali G, Alshehri NO (2017) A new multi-attribute decision-making method based on m-polar fuzzy soft rough sets. *Symmetry* 9(271):1–18
- Chen D, Tsang ECC, Yeung DS, Wang X (2005) The parameterization reduction of soft sets and its applications. *Comput Math Appl* 49:757–763. <https://doi.org/10.1016/j.camwa.2004.10.036>
- Kong Z, Gao L, Wang L, Li S (2008) The normal parameter reduction of soft sets and its algorithm. *Comput Math Appl* 56(12):3029–3037. <https://doi.org/10.1016/j.camwa.2008.07.013>
- Ali MI (2011) A note on soft sets, rough soft sets and fuzzy soft sets. *Appl Soft Comput* 11:3329–3332. <https://doi.org/10.1016/j.asoc.2011.01.003>
- Ali MI (2012) Another view on reduction of parameters in soft sets. *Appl Soft Comput* 12:1814–1821. <https://doi.org/10.1016/j.asoc.2012.01.002>
- Kong Z, Wang L, Jia W (2015) Approximate normal parameter reduction of fuzzy soft set based on harmony search algorithm. In: *Proceeding of 2015 IEEE fifth international conference on big data and cloud computing, IEEE BDC15*. IEEE Press, pp 321–324
- Kong Z, Jia WH, Zhang GD, Wang LF (2015) Normal parameter reduction in soft set based on particle swarm optimization algorithm. *Appl Math Model* 39(16):4808–4820. <https://doi.org/10.1016/j.apm.2015.03.055>
- Han BH (2016) Comments on Normal parameter reduction in soft set based on particle swarm optimization algorithm. *Appl Math Model* 40(1):10828–10834. <https://doi.org/10.1016/j.apm.2016.06.004>
- Han BH, Li YM, Geng SL (2017) A 0-1 linear programming method for optimal normal and pseudo parameter reductions of soft sets. *Appl Soft Comput* 54(1):467–484. <https://doi.org/10.1016/j.asoc.2016.08.052>
- Kong Z, Ai J, Wang L, Li P, Ma L, Lu F (2017) New normal parameter reduction method in fuzzy soft set theory. *IEEE Access* 7:2986–2998. <https://doi.org/10.1109/ACCESS.2018.2888878>
- Ma X, Qin H (2018) A distance-based parameter reduction algorithm of fuzzy soft sets. *IEEE Access* 6:10530–10539. <https://doi.org/10.1109/ACCESS.2018.2800017>
- Khan A, Zhu TG (2019) A novel approach to parameter reduction of fuzzy soft set. *IEEE Access* 7:128956–128967. <https://doi.org/10.1109/ACCESS.2019.2940484>
- Ma XQ, Fei QH, Qin HW, Zhou XY, Li HF (2019) New improved normal parameter reduction method for fuzzy soft set. *IEEE Access* 7:154912–154921. <https://doi.org/10.1109/ACCESS.2019.2949142>
- Ma X, Sulaiman N, Qin H, Herawan T, Zain JM (2011) A new efficient normal parameter reduction algorithm of soft sets. *Comput*

Math Appl 62(2):588–598. <https://doi.org/10.1016/j.camwa.2011.05.038>

44. Geng SL, Han BH, Wu RZ, Xu RQ (2020) An offline and online algorithm for all minimal $k|U|$ parameter subsets of a soft set based on integer partition. IEEE Access 8:2169–3536. <https://doi.org/10.1109/ACCESS.2020.3032578>
45. Cai MK, Geng SL, Lin LH, Han BH (2019) Minimum normal parameter reduction of soft set based on local search. J Qinghai Normal Univ (Natural Science Edition) 2:19–25
46. Electoral College (2016) 2016 Electoral College Results. National Archives Web. <https://archives.gov/electoralcollege/2016?ga=2.154881784.1961345302.1632059232-447802178.1632059232>. Accessed 28 September 2021

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Banghe Han received the B.S. degree in mathematics and applied mathematics from Shaanxi Normal University, Xi'an, Shaan Xi Province, in 2004, the M.S. degree in uncertainty reasoning from Shaanxi Normal University in 2007 and the Ph.D. degree in computational intelligence from Shaanxi Normal University in 2011.

From 2009 to 2015 he is a lecturer with the School of Mathematics and Statistics, Xidian University, Xi'an,

China. From 2015 to now, he is an associate professor with the School of Mathematics and Statistics, Xidian University. His research interest includes uncertainty reasoning theories such as fuzzy sets, fuzzy logic, soft sets, rough sets and so on. Dr Han's awards includes the first prize of excellent paper award for young people of the Shaanxi Mathematics Association in 2014 and the second prize of Xi'an science and technology progress award in 2017.



Ruize Wu is now a junior at the School of Physics and Optoelectronic Engineering in Xidian University. His research interest includes electromagnetic wave propagation and antenna design.

He received scientific research training by participating in three undergraduate mathematical contests in Modeling during his two years in college, and he is participating in his fourth mathematical modeling contest as the team leader. He also participated

in the China Undergraduate Physics Tournament (CUPT) and was promoted to the regional competition. He has been working on programming problems related with parameter reduction problem soft set for one year and has a strong interest in mathematical research and mathematical modeling. He likes to discover hidden patterns in mathematical problems.



Shengling Geng was born in Qinghai, China. She received her B.S. degree and M.S. degree from Qinghai Normal University and Ph.D degree from ShaanXi Normal University. She is currently a professor with the School of Computer Science, Qinghai Normal University. And she also works in Academy of Plateau Science and Sustainability in Qinghai Province, China.

Her research interest includes soft computing, data mining, soft set theory in decision making and intelligent control. She has presided over and completed one national natural fund project, three provincial scientific research projects, and one preliminary research project of national major basic research (973 Program). She has received two times of the third prize of provincial Science and Technology Progress Award in Qinghai Province.



Xinyu Nie is now a junior at the School of Electronic Engineering in Xidian University. Her research interest includes electronic circuit design and electromagnetic field and microwave technology.

During the two years in university, she received systematic training in mathematical modeling, and participated in four modeling college mathematics competitions. At the same time, as a member of the thesis, she is participating in her fifth mathematical modeling competition. She also took part in the Internet + and Challenge Cup, and has achieved preliminary results. She has been engaged in research and writing problems related to soft sets of parameter reduction problems for one year, and has a strong interest in mathematical research and mathematical modeling.