Decision Tables Aggregation in Rough Sets Approximation

Salem Chakhar\textsuperscript{1}, Clara Pusceddu\textsuperscript{2}

\textsuperscript{1} CRAD, ESAD, University Laval, Québec City, Québec, Canada.
\textsuperscript{2} Faculty of Architecture of Alghero, University of Sassari, Sardinia, Italy.

Abstract

The Dominance-based Rough Set Approach (DRSA) is an extension of Rough Sets Theory to handle multicriteria classification problems by authorizing preference-ordered attributes. The DRSA assumes the existence of a single decision table while real-world decision problems imply generally several experts with different decision tables. The objective of this paper is to propose an algorithm for the aggregation of a set of decision tables, as a first step for approximating these tables. The algorithm is illustrated using real-world data.

Introduction

The Dominance-based Rough Set Approach (DRSA) (Greco, Matarazzo, and Slowiński 2001) is an extension of rough sets theory (Pawlak 1991) to handle multicriteria classification problems by authorizing preference-ordered attributes. The input data for DRSA are often structured in a decision table where rows correspond to decision objects and columns correspond to attributes. The attributes used in rough approximation in multicriteria classification problems are often divided into two disjoint subsets: a subset of condition attributes and a subset of decision attributes. The DRSA assumes the existence of a single decision table. However, multicriteria classification problems generally imply different experts having different and conflicting objectives and preferences, each with its decision table.

The approximation of several decision tables has been addressed by several authors (Bi and Chen 2007; Chakhar and Saad 2012; Chen, Kilgour, and Hipel 2012; Greco, Matarazzo, and Slowiński 2006). The first step of rough approximation of decision tables consists in the aggregation of these tables into a collective decision table with one collective decision attribute. The objective of this paper is to propose an algorithm for the aggregation of a set of decision tables as a first step to rough approximation of these tables. The algorithm is illustrated using real-world data.

The rest of paper is organized as follows. Section 2 presents the background and set decision tables aggregation problem. Section 3 presents the aggregation algorithm. Section 4 presents a numerical example. Section 5 concludes the paper.

Decision tables aggregation problem

In rough sets theory, information regarding the decision objects is often structured in a 4-tuple information table \( S = \langle U, Q, V, f \rangle \), where \( U \) is a non-empty finite set of objects and \( Q \) is a non-empty finite set of attributes such that \( q : U \rightarrow V_q \) for every \( q \in Q \). \( V_q \) is the domain of attribute \( q \). \( V = \bigcap_{q \in Q} V_q \) and \( f : U \times Q \rightarrow V \) is the information function defined such that \( f(x, q) \in V_q \) for each attribute \( q \) and object \( x \in U \). \( Q \) is often divided into a sub-set \( C \neq \emptyset \) of condition attributes and a sub-set \( D \neq \emptyset \) of decision attributes such that \( C \cup D = Q \) and \( C \cap D = \emptyset \). In this case, \( S \) is called a decision table.

In multicriteria decision-making, the domain of condition attributes are supposed to be ordered according to decreasing or increasing preference. Such attributes are called criteria. We assume that the preference is increasing with a value of \( f(\cdot, q) \) for every \( q \in C \). We also assume that the set of decision attributes \( D = \{ d \} \) is a singleton. The unique decision attribute \( d \) makes a partition of \( U \) into a finite number of preference-ordered decision classes \( Cl = \{ Cl_1, \cdots, Cl_n \} \) such that each \( x \in U \) belongs to one and only one class.

In DRSA the represented knowledge is a collection of upward union \( Cl^U_t \) and downward union \( Cl^C_t \) of classes defined as follows:

\[
Cl^U_t = \bigcup_{s \geq t} Cl_s, \quad Cl^C_t = \bigcup_{s \leq t} Cl_s.
\]

The assertion “\( x \in Cl^U_t \)” means that “\( x \) belongs to at least class \( Cl_i \)” while assertion “\( x \in Cl^C_t \)” means that “\( x \) belongs to at most class \( Cl_i \).” The basic idea of DRSA is to replace indiscernibility relation used in conventional rough sets theory with dominance relation. The dominance relation \( \Delta_P \) associated with \( P \) is defined for each pair of objects \( x \) and \( y \) as follows:

\[
x \Delta_PY \Leftrightarrow f(x, q) \geq f(y, q), \forall q \in P.
\]

To each object \( x \in U \), we associate two sets: (i) the \( P \)-dominating set \( \Delta^+_P(x) = \{ y \in U : y \Delta_P x \} \) containing objects that dominate \( x \), and (ii) the \( P \)-dominated set \( \Delta^-_P(x) = \{ y \in U : x \Delta_P y \} \) containing the objects dominated by \( x \). The \( P \)-lower and \( P \)-upper approximations of \( Cl^U_t \) with respect to \( P \subseteq C \) are defined as follows:

- \( P(Cl^U_t) = \{ x \in U : \Delta^+_P(x) \subseteq Cl^U_t \} \),
\( \bar{P}(C_i) = \{ x \in U : \Delta \bar{P}(x) \cap C_i \neq \emptyset \} \).

Analogously, the \( P \)-lower and \( P \)-upper approximations of \( C_i \) with respect to \( P \subseteq C \) are defined as follows:

\( P(C_i) = \{ x \in U : \Delta P(x) \subseteq C_i \} \),  
\( \bar{P}(C_i) = \{ x \in U : \Delta \bar{P}(x) \supseteq C_i \} \).

The \( B_n \)-boundaries of \( C_i \) and \( \bar{P} \) are defined as:

\( B_n(C_i) = \bar{P}(C_i) - P(C_i) \),  
\( B_n(\bar{P}) = \bar{P}(\bar{P}) - P(\bar{P}) \).

The quality of classification (or approximation) of a partition \( C_l \) by means of a set of criteria \( P \) is measured by the ratio \( \gamma_P \), which expresses the ratio of all \( P \)-correctly classified objects to all objects in the system.

Let \( H = \{ 1, \ldots, h \} \) and let \( S_i = \langle U, C \cup \{ E_i \}, V, f_i \rangle \) (\( \forall i \in H \)) be \( n \) decision tables where \( E_i \) and \( f_i \) are respectively the decision attribute and the information function relative to the \( i \)th decision table. We assume that a preference order for \( U \) represented by a finite set of preference-ordered classes \( C_l \) is \( \{ C_{i,t} \}, t \in T, i \leq n \), \( \forall r, t \in T, r \neq t, \) and if \( x \in C_{i,r}, y \in C_{i,s}, \) and \( r > s, \) then \( x \) is better than \( y \) for the \( i \)th decision table. The \( n_i \) is the number of decision classes for the \( i \)th decision table.

The approximation of the \( i \)th decision table \( S_i \) is characterized, among others, by: (i) \( P \)-lower approximation and \( P \)-boundary of \( C_{i,t} \) and \( \bar{C}_{i,t} \), for each \( t \in T, i \) and (ii) the \( \gamma_P \) of classification \( \gamma_P \).

The first step of rough approximation of decision tables consists in the aggregation of these tables into a collective decision table with one collective decision attribute. The problem of decision tables aggregation can be stated as follows: Let \( S_i = \langle U, C \cup \{ E_i \}, V, f_i \rangle \) (\( \forall i \in H \)). Then, construct a collective decision table \( S = \langle U, C \cup \{ E \}, V, g \rangle \) where \( E \) is a decision attribute and \( g \) is an information function defined for each \( x \in U \) as follows:

\[
g(x,q) = \begin{cases} f(x,q), & \text{if } q \in C, \\ g(x,E), & \text{if } q = E. \end{cases}
\]

The decision attribute \( E \) induces a partition of \( U \) into \( h \) sets of decision classes \( C_l = \{ C_{l_1}, \ldots, C_{l_h} \} \) such that each \( x \in U \) belongs to one and only one class \( C_{l_i} \in C_l \). To define \( S \) it suffices to specify the values of \( g(x,E) \) for all \( x \in U \).

### Decision tables aggregation algorithm

As stated above, the objective of the aggregation algorithm is to construct a decision table \( S \) by aggregating the decision tables \( S_1, \ldots, S_h \). The idea of the aggregation algorithm is to use the upward and downward approximation of unions of classes in order to identify the possible assignments classes for each decision object. Two sets will be constructed: (i) set \( N_1 \) contains the possible assignments obtained based on the upward approximation of unions of classes; and (ii) set \( N_2 \) contains the possible assignments obtained based on the downward approximation of unions of classes. These sets will then be used to associate to each object \( x \in U \) an assignment interval \( I(x) = [l(x), u(x)] \) where \( l(x) \) and \( u(x) \) are respectively the lower and upper classes to which object \( x \) can be assigned. Finally, some simple rules are used to reduce the assignment interval \( I(x) \) into a single element representing the value of the collective decision attribute \( E \).

Before introducing the aggregation algorithm we need to introduce new concepts. More specifically, the definition of sets \( N_1 \) and \( N_2 \) requires the introduction of three concepts: concordance power, discordance power and the credibility indexes. Let first standardize the quality of classifications \( \gamma_P (\forall k \in H) \) as follows:

\[
\gamma_k = \frac{\gamma_P}{\sum h \gamma_P}
\]

We assume that \( o \in \{ \geq, \leq \} \) and \( C_l = \{ C_{l_1}, \ldots, C_{l_n} \} \).

#### Concordance power

For each \( x \in U \) and \( C_{l_i} \in C_l \) we define the set: \( L(x, C_{l_i}) = \{ i : i \in H \land x \in P(C_{l_i}) \} \) where \( P(C_{l_i}) \) is the \( P \)-lower approximation of \( C_{l_i} \) in respect to the \( i \)th decision table. Then, the concordance powers for the assignment of \( x \) to \( C_{l_i} \) are defined as follows.

**Definition 1** The concordance power for the assignment of \( x \) to \( C_{l_i} \) is computed as follows:

\[
S(x, C_{l_i}) = \sum_{k=1}^{h} S_k(x, C_{l_i})
\]

where:

\[
S_k(x, C_{l_i}) = \begin{cases} \gamma_k, & \text{if } k \in L(x, C_{l_i}), \\ 0, & \text{otherwise.} \end{cases}
\]

#### Discordance power

For each \( x \in U \) and \( C_{l_i} \in C_l \) we define the set: \( B(x, C_{l_i}) = \{ i : i \in H \land x \in B_n(C_{l_i}) \} \) where \( B_n(C_{l_i}) \) is the boundary of \( C_{l_i} \) in respect to the \( i \)th decision table. Then, the discordance powers for the assignment of \( x \) to the boundary of \( C_{l_i} \) is defined as follows.

**Definition 2** The discordance power for the assignment of \( x \) to \( C_{l_i} \) is computed as follows:

\[
Z(x, C_{l_i}) = \prod_{k=1}^{n} Z_k(x, C_{l_i})
\]

where

\[
Z_k(x, C_{l_i}) = \begin{cases} \frac{1 - \gamma_k^i}{S(x, C_{l_i})}, & \text{if } \gamma_k^i > S(x, C_{l_i}) \land k \in L(x, C_{l_i}), \\ 1, & \text{otherwise.} \end{cases}
\]

We may distinguish two cases in the definition of the discordance power. The first case holds when \( \gamma_k^i \leq S(x, C_{l_i}) \), which leads to \( Z_k(x, C_{l_i}) = 1 \). In this case, there is no veto effect for decision maker \( k \) and \( Z_k(x, C_{l_i}) \) will have no effect on the definition of overall discordance power \( Z(x, C_{l_i}) \) and on the value of the credibility indexes as explained in the next paragraph. The second case holds when \( \gamma_k^i > S(x, C_{l_i}) \), which leads to \( 0 < Z_k(x, C_{l_i}) < 1 \). Here, decision maker \( k \) do have a veto effect and \( Z_k(x, C_{l_i}) \) will have an effect on the value of overall discordance power \( Z(x, C_{l_i}) \) and on the value of the credibility indexes as explained later.
Credibility indexes  Using the concordance and discordance powers, we may define the credibility index for assigning \( x \) to \( C_{I_{k}^{x}} \) as follows.

**Definition 3** Let \( x \in U \) and \( \circ \in \{ \geq, \leq \} \). The credibility indexes for the assignment of \( x \) to \( C_{I_{k}^{x}} \) is computed as follows:

\[
\sigma(x, C_{I_{k}^{x}}) = S(x, C_{I_{k}^{x}}) \cdot Z(x, C_{I_{k}^{x}}) \quad (7)
\]

This formula can be explained as follows. If there is no support for the assignment of \( x \) to \( C_{I_{k}^{x}} \), i.e., \( S(x, C_{I_{k}^{x}}) = 0 \), then the credibility indexes will be \( \sigma(x, C_{I_{k}^{x}}) = 0 \). In turn, if there is a full support, i.e., \( S(x, C_{I_{k}^{x}}) = 1 \) (which imposes that \( Z(x, C_{I_{k}^{x}}) = 1 \)), then credibility indexes will be \( \sigma(x, C_{I_{k}^{x}}) = 1 \). Finally, if \( x \) is a partial support, i.e., \( 0 < S(x, C_{I_{k}^{x}}) < 1 \) (which imposes that \( 0 < Z(x, C_{I_{k}^{x}}) \leq 1 \)), then \( 0 < \sigma(x, C_{I_{k}^{x}}) < 1 \). In the last case, we may distinguish two subcases, according to the verification or not of the condition \( \gamma_{k}^{x} > S(x, C_{I_{k}^{x}}) \). The first subcase holds when the condition \( \gamma_{k}^{x} > S(x, C_{I_{k}^{x}}) \) is not verified. This leads to \( Z(x, C_{I_{k}^{x}}) = 1 \) and then \( \sigma(x, C_{I_{k}^{x}}) = S(x, C_{I_{k}^{x}}) < 1 \). In this subcase, the credibility index is simply equal to the concordance power; hence the discordance power will have no effect on the value of the credibility indexes \( \sigma(x, C_{I_{k}^{x}}) \). The second subcase holds when condition \( \gamma_{k}^{x} > S(x, C_{I_{k}^{x}}) \) is verified. This leads to \( Z(x, C_{I_{k}^{x}}) < 1 \) and consequently \( \sigma(x, C_{I_{k}^{x}}) = S(x, C_{I_{k}^{x}}) \cdot Z(x, C_{I_{k}^{x}}) < 1 \). In this subcase, the credibility index is obtained by decreasing the concordance power \( S(x, C_{I_{k}^{x}}) \) proportionally to the value of the discordance power \( Z(x, C_{I_{k}^{x}}) \).

**Definition of assignment interval** Let \( \lambda \in [0.5, 1] \) be a credibility threshold. Then, based on the credibility indexes, we may define the sets \( N_{1} \) and \( N_{2} \) as follows.

**Definition 4** The credibility indexes, we may define the sets \( N_{1} \) and \( N_{2} \) as follows:

- \( N_{1}(x) = \{ C_{I_{k}^{x}} : \ x \in U \land \sigma(x, C_{I_{k}^{x}}) \geq \lambda \} \),
- \( N_{2}(x) = \{ C_{I_{k}^{x}} : \ x \in U \land \sigma(x, C_{I_{k}^{x}}) \leq \lambda \} \).

Then, the idea for the definition of assignment intervals is to constrain possible assignment classes by the content of sets \( N_{1}(x) \) and \( N_{2}(x) \). Indeed, the set \( N_{1}(x) \) is defined based on the upward union of classes \( C_{I_{k}^{x}} \); it should be used to define the lower limit \( l(x) \) of the assignment interval of \( x \). In turn, the set \( N_{2}(x) \) is defined based on downward union of classes \( C_{I_{k}^{x}} \); it should be used to define the upper limit \( u(x) \) of the assignment interval of \( x \).

**Definition 5** Let \( x \in U \). Then, we associate to each object \( x \) an assignment interval \( I(x) = [l(x), u(x)] \) where:

\[
l(x) = \begin{cases} \arg\max_{C_{I_{k}^{x}} N_{1}(x)}, & \text{if } N_{1}(x) \neq \emptyset, \\ C_{I_{0}}, & \text{otherwise}. \end{cases} \quad (8)
\]

\[
u(x) = \begin{cases} \arg\min_{C_{I_{k}^{x}} N_{2}(x)}, & \text{if } N_{2}(x) \neq \emptyset, \\ C_{I_{n}}, & \text{otherwise}. \end{cases} \quad (9)
\]

**Reduction of the assignment interval** Let \( I(x) = [l(x), u(x)] \) be the assignment interval for object \( x \in U \) defined as previously. Two cases hold for the reduction of \( I(x) \). The first case holds when \( l(x) = u(x) \). Here, object \( x \) is assigned to a single class and consequently we can set \( g(x, E) = l(x) \) (or \( g(x, E) = u(x) \)). The second case holds when \( l(x) < u(x) \). This corresponds to the situation where object \( x \) can be assigned to more than one class. To specify the value of \( g(x, E) \) when the second case holds we may apply one of the following rules to reduce the collective assignment interval \( I(x) \) to a single class: the minimum value, the maximum value, the floor of the median value, and the ceiling of the median value.

**Aggregation algorithm** The aggregation procedure is formalized in Algorithm 1. This algorithm works as follows. It loops on the set of decision objects and for each object: (i) computes the credibly indexes for upward unions of classes (the first inner for loop); (ii) computes the credibly indexes for downward unions of classes (the second inner for loop); (iii) computes the assignment interval \( I(x) = [l(x), u(x)] \); and (vi) computes the values of the collective decision attribute \( E \).

Functions SigmaUpward and SigmaDownward permit to compute the credibility indexes and function IntervalReduction permits to compute the assignment interval.

**Application**

We consider a real-world data relative to the management of post-accident nuclear risk in the PRIME project (Chakhar and Saad 2012). The problem involves 18 decision objects and 7 attributes (radioecological vulnerability of agricultural area (\( A_{1} \)), radioecological vulnerability of forest area (\( A_{2} \)), radioecological vulnerability of urban area (\( A_{3} \)), real estate vulnerability (\( A_{4} \)), tourism vulnerability (\( A_{5} \)), economic vulnerability of companies (\( A_{6} \)), and employment vulnerability (\( A_{7} \)).

The main input is three decision tables summarized in Table 1. Each object is described in terms of seven condition attributes (\( A_{1}, A_{2}, \cdots, A_{7} \)) and three decision attributes (\( E_{1}, E_{2}, \) and \( E_{3} \)). The values of condition attributes correspond to vulnerability levels. The values of decision attributes correspond to the global vulnerability levels as specified by three experts. All condition and decision attributes are evaluated on a six-level ordinal scale (from normal situation (0) to major and long-lasting negative impact (5)).

The software 4eMka2 (which implements the DRSA) is used to approximate the individual decision tables. The outputs of individual approximations are then compiled in a single .txt file and provided as input to a prototype implementing the aggregation algorithm.

The credibility indexes values computed using Equation (7) are given in Table 2. The assignment intervals along with the application of interval reduction rules are given in Table 3.
Algorithm 1: Aggregation

Input: $\lambda$ where $\lambda = (U, C, V, f)$ is the common decision table.
$N_1, \ldots, N_h$ where $N_i = (U, C \cup \{E\}, V, f_{\lambda})$.
\(\lambda\) // where $\lambda = [0.5, \ldots, 1]$ is the credibility threshold.
\(ir\text{-}rule\) // where \(ir\text{-}rule\) is the interval reduction rule.
\(def\text{-}fault\text{-}rule\) // where \(def\text{-}fault\text{-}rule\) is the interval reduction rule.
\(de\text{-}fault\text{-}rule\) // default rule to use when "median" rule do not apply.

Output: $S$ where $S = (U, C \cap E, V, g)$ is the aggregated decision table.
$E \leftarrow$ decision attribute,
$Q \leftarrow C \cup \{E\},$
$H \leftarrow \{1, 2, \ldots, h\}$

for $(a(x) \in U)$ do
  $l(x) \leftarrow [l(x)]$;
  $u(x) \leftarrow [u(x)]$;

end

if $\sigma(x) \geq \lambda$ then
  $N_1 \leftarrow N_1 \cup C_{1t}$;
end

if $\sigma(x) \geq \lambda$ then
  $N_2 \leftarrow N_2 \cup C_{1t}$;
end

end

$N \leftarrow \argmax_{C_{1t}} N_1(x)$;

for $(all \ (q \in C))$ do
  $g(x, q) \leftarrow f(x, q)$;
end

if $\{l(x) \leq u(x)\}$ then
  $g(x, E) \leftarrow l(x)$;
else
  $g(x, E) \leftarrow \text{IntervalReduction}(l(x), u(x), ir\text{-}rule, def\text{-}fault\text{-}rule)$;
end

$S \leftarrow S \cup U, Q, V, g >$;
return $S$

Conclusion

We proposed an algorithm for the approximation of a set of decision tables. The algorithm is illustrated using real-world data. In the future, we intend to study the mathematical properties of the introduced concepts. We also intend to conceive and to develop a full-featured decision support system supporting the aggregation algorithm.

References


Table 1: Information table with assignment examples

<table>
<thead>
<tr>
<th>Object</th>
<th>$\sigma(x_1, C_{1t}^x)$</th>
<th>$\sigma(x_1, C_{1t}^x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_4$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_5$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_6$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_7$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_8$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_9$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_10$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_11$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_12$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_13$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_14$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_15$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_16$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_17$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_18$</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: Credibility indexes values

| Object | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | $x_{10}$ | $x_{11}$ | $x_{12}$ | $x_{13}$ | $x_{14}$ | $x_{15}$ | $x_{16}$ | $x_{17}$ | $x_{18}$ |
| $\mu(x)$ | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |
| $\max(x)$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| $\min(x)$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\text{floor}(x)$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\text{ceil}(x)$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Table 3: Collective decision attribute $E$ values for $\lambda = 0.70$

<table>
<thead>
<tr>
<th>Object</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_3$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_4$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_5$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_6$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_7$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_8$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_9$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_{11}$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_{12}$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_{13}$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_{14}$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_{15}$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_{16}$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_{17}$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_{18}$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

