Journal of Economics and Management Sciences; Vol. 6, No. 2; 2023
ISSN 2576-3008 E-ISSN 2576-3016
https://doi.org/10.30560/jems.v6n2p11

Search Algorithms for a System of Different Representatives of Subsets of a Finite Set

Dmitri Terzi[1]

[1] State University of Moldova, Chisinau Moldova

Correspondence: Dmitri Terzi, State University of Moldova, Chisinau Moldova. E-mail: dgerterzi@yandex.ru

Received: April 25, 2023; Accepted: May 29, 2023; Published: June 7, 2023

Abstract

The paper considers various aspects and approaches regarding the search for a system of differenrt representatives (*SDR*) of subsets of a finite set. Algorithm 3 has been developed, which is an algorithmization of the search process for *SDR* on the basis of Hall's theorem on the existence of *SDR*. Algorithms 1-5 for searching for partial and complete *SDRs* based on the technology for solving problems of the transport type, in particular, the algorithm associated with the construction of cycles, are given. *SDR* search algorithms can be used as auxiliary tools for solving combinatorial problems. It is also shown, conversely, that the search for *SDR* can be considered as a combinatorial optimization problem.

Mathematics Subject Classification – MSC2020: 90-08, 90B06, 90C27

Keywords: systems of different representatives, Hall's theorem, algorithmization, combinatorial optimization, transport-type problems

1. Introduction

Let S be a non-empty finite set, $M = (S_1, S_2, ..., S_m)$ −some sequence of its non-empty subsets. A system of different representatives (*SDR)* for a collection of sets *S* is a set $a = \{a_1, a_2, ..., a_k\}$ such that ai belongs to S_i and for any *i* and *j* $a_i \neq a_j$. An *SDR* for any subsequence of the sequence *M* is said to be partial.

The need to construct a system of distinct representatives (*SDR*) arises both in purely combinatorial mathematical studies and in their applications to linear programming, mathematical economics, and combinatorial optimization (Hall, M. 1967; Wilson R. 1972; Brualdi, RA. 2009; Jukna S. 2011; Morris, J. 2023). Within the framework of combinatorial mathematics, *SRPs* are mainly used in solving problems of choice and optimization, in particular, in studying Latin rectangles, in the assignment problem, and in studying special matrices with non-negative elements. At the same time, the analysis of studies on *SDR* shows that the task of developing *SDR* search algorithms, especially algorithms that build *SDR* with certain properties, is relevant. In this regard, the algorithms 1-5 presented in the paper can be useful. Let us give examples of practical interpretation of *SDR*:

a) There are *m* commissions in some institution. It is required from the composition of each commission to appoint their chairmen so that no one person presides over more than one commission. Here the *SDRs* of the commissions will be composed by their chairmen (Lipski W. 1982; Rybnikov, K.A. 1985);

b) There are *n* vacancies in the company, each of which requires a certain qualification. There are m people who apply for *n* jobs. What is the largest number of vacancies that can be filled from the pool of applicants, if the vacancy can only be filled by a person corresponding to his qualifications. In this problem, the interesting question is the largest number of jobs that can be filled by qualified candidates and the specific assignment of the largest number of applicants to the jobs they apply for;

c) Suppose we are given a set of hours per week, a set of *n* teachers, each of whom has the opportunity to teach during some subset of hours, a set of *m* classes, each of which has only a few hours per week in the schedule, and an *nxm* matrix of non-negative integers numbers, in which the (*i,j*)-th element means the number of hours during which the *i*-th teacher must teach in the *j*-th class. It is required to determine whether it is possible to schedule lessons and distribute teachers in such a way that each class met at the required hours, each teacher could be in his classes, there would be only one teacher per class and no teacher would be assigned to two classes at the same time.

In problems of type a)-b), it is required to determine a complete system of different representatives or a partial system, but with the maximum number of representatives. When studying the problems associated with *SDR*, the practical issue of algorithmization of the process of searching for such systems is relevant.

The system of different representatives consists of elements a_i, one from each subset S_j In case

$$S= \{S_1, S_2,S_3,S_4\} = \{\{s_1, s_4, s_5\}, \{s_1\}, \{s_2, s_3, s_4\}, \{s_2, s_4\}\}$$

one can define four *SDR*:

$$a_1 = \{s_1, s_2, s_3, s_4\}, \quad a_2 = \{s_1, s_2, s_3, s_5\},$$

$$a_3 = \{s_1, s_3, s_4, s_5\}, \quad a_4 = \{s_1, s_2, s_4, s_5\}.$$

SDR exists only if for any *k* subsets S_j their union contains at least *k* elements [1]. For example, in the family of subsets $S_1,\ S_2,\ S_3,\ S_4,\ S_5,$ sets

$$S=\{\{s_1, s_2\}, \{s_1, s_3\}, \{s_2, s_3\}, \{s_1, s_2, s_3\}, \{s_1, s_3, s_4, s_5\}\},$$

unable to find five distinct s_i elements one from each S_j, so this family has no *SDR*. Here the union of four sets $\bigcup_{i=1}^{4} S_i$ contains only three elements $\{s_1,\ s_2,\ s_3\}$.

The idea of replacing sets with their representatives turned out to be fruitful and was further developed. Systems of representatives are singled out taking into account the conditions of the tasks or the goals of theoretical generalizations. For example, task research about splitting sets, led to the concept of systems of common representatives. Such a problem, like many other problems of combinatorial optimization, reduces to the problem of finding an *SDR* for subsets of a finite set.

2. Criteria for the Existence of *SDR* a Family of Subsets

With search *SDR* the following questions are solved: is there a complete *SDR* for a given family of subsets of a finite set (a partial *SDR* always exists because subsets are non-empty); how to determine the maximum (by the number of subsets) system of representatives. Further, we assume that $S_1, S_2, ..., S_k$ are subsets of a finite sets $S=\{1,2,3,...,n\}$. If the total number of sets is finite, and the sets themselves are infinite, then we can discard all elements except *n* in each S_i, $i=1,...,k$. The existence of an *SDR* can be established by the following criteria a)-b).

a) The first criterion follows from Hall's theorem. The subsets $S_1, S_2, ..., S_n$ have *SDR* if and only if among the elements of any finite number *k* of sets s_i there are at least s_i distinct elements;

b) Another criterion for the existence of an *SDR* of a family of subsets is based on Sperner's theorem (Kotre B., Vygen J., 2015).

Definition. The set of subsets $S_1, S_2, ..., S_k$ is called Sperner collection, if none of the sets $S_1, S_2, ..., S_k$ is a part of the other.

Sperner's theorem. Let *S* be a set that consists of n elements, $S_1, S_2, ..., S_k$ be a Sperner collection. Then for the Sperner collection there exists an *SDR*.

For example, sets

$$S_1 =\{1,2,7\},\ \ S_2 =\{3,5\},\ \ S_3 =\{2,8\},\ \ S_4 =\{4,6\},\ \ S_5 =\{3,7,9\},$$

$$S_6 =\{1,2,3\},\ \ S_7 =\{1,4,5,8,9\},\ \ S_8 =1,6,7\},\ \ S_9 =\{2,5,8\} \qquad (1)$$

are such that neither is part of the other. In this case, for these sets, there exists an *SDR*.

Each of the criteria allows us to conclude that subsets have *SDRs*, but not about their definition. Therefore, such criteria are used, one might say, at the stage of preliminary information processing. The question of searching for a complete or partial *SDR* is solved by special methods of combinatorics or graph theory. Let us consider some of these algorithms for searching for partial or complete *SDR*, preferring to present them in a more natural combinatorial interpretation.

3. Algorithms for Constructing *SDR*

The subsets S_i of the set *S* are represented by the matrix *C*. Each row corresponds to one subset. As a demonstration example of the operation of algorithms, we note that for the set (1) the corresponding matrix *C* has the form

$$C=[1\ 2\ 0\ 0\ 0\ 0\ 7\ 0\ 0;\ 0\ 0\ 3\ 0\ 5\ 0\ 0\ 0\ 0;\ 0\ 2\ 0\ 0\ 0\ 0\ 0\ 8\ 0;$$
$$0\ 0\ 0\ 4\ 0\ 6\ 0\ 0\ 0;\ 0\ 0\ 3\ 0\ 0\ 0\ 7\ 0\ 9;\ 1\ 2\ 3\ 0\ 0\ 0\ 0\ 0\ 0; \tag{2}$$
$$1\ 0\ 0\ 4\ 5\ 0\ 0\ 8\ 9;\ 1\ 0\ 0\ 0\ 0\ 6\ 7\ 0\ 0;\ 0\ 2\ 0\ 0\ 5\ 0\ 0\ 8\ 0].$$

In the description of the following search algorithms for *SDR* such a matrix form of representation of subsets S_i, $i=1,\dots,m$, from a finite set $S=\{1,2,\dots n\}$ is used, which is more consistent with the formulation of criteria in combinatorial form.

3.1 Algorithm 1. Alternate choice of representatives of sets.

The choice of set representatives is carried out in turn, looking at the rows of the matrix *C* from left to right.

The first non-zero element s_{1j} of the first row and *j* of the column will be a representative of the set S_1. Cross out the first row and column *j*;

b) Go to the next uncrossed line *i*. The first non-zero and not crossed out element of row *i* will be a representative of the set S_i;

c) Repeat step b) as far as possible in order to determine the representatives of the sets. If there is no representative for S_i on some *i*, $i\le n$, then the search process stops and we get a partial *SDR*.

Such a method in many cases can reveal the full *SDR*. However, it may happen that in the presence of a complete *SDR* for the sets S_i, by Algorithm 1 we can determine only a partial *SDR*. The volume of such a system depends on the order of viewing the sets S_i.

3.2 Algorithm 2. Search for SDR. taking into account the number of elements in subsets

1) Find all rows *i* containing only one positive element. Each such element will be a representative of the set S_i, and the corresponding row *i* and column *j* of the matrix C are crossed out;

2) Find all columns *j* containing only one positive element. Each such element from row *i* will be a representative of the set S_i, and the corresponding row *i* and column *j* of the matrix *C* are deleted;

3) If representatives for all sets are determined, then the process ends;

4) If there are uncrossed rows or columns, then the numbers k_i and k_j are determined:

k_i is the number of uncrossed out elements in the *i*-th line, k_j is the number of non-crossed-out elements in the *j*-th column;

5) k_1 and k_2 are defined as the minimum of k_i and k_j, respectively;

6) If $k_1\le k_2$, then a row containing k_1 elements is selected, one of the elements is selected, and the row and column are crossed out;

7) If $k_1> k_2$, then a column containing k_2 elements is selected, one of the elements is highlighted, and the row and column are crossed out.

All selected elements at the end of the algorithm constitute the *SDR*, which can be full or partial. For an example with matrix (2), we get the full *SDR*:

$$s=\left\{s_{23},s_{32},s_{44},s_{61},s_{17},s_{86},s_{59},s_{75},s_{98}\right\}.$$

Algorithm 2 can be used as an auxiliary algorithm in more complex algorithms that always determine a complete *SDR*. (if such a system exists) or establish that such a system does not exist for given subsets of a finite set.

3.3 Algorithm 3. Search for *SDR* using Hall's theorem

At the beginning of Algorithm 3, Algorithm 1 is applied. The sets $S_1,\ldots,S_n$ are considered to be numbered and fixed in a certain order. The process of searching for representatives is carried out in turn, starting from the first set and up to the set S_k. If *k* is equal to *n*, then we get the full *SDR*. If *k*<*n*, then this means that all elements of S_{k+1} are representatives of other sets. In such a situation, the *SDR* search process is not stopped. The possibility of the existence of an *SDR* is clarified, in particular, the existence of a representative for the set S_{k+1}. This part of Algorithm 3 requires the algorithmization of the process of further search for *SDR*. A sequence of sets T_0, T_i, *i=1,...,t* is constructed, where T_0 is the initial set, consisting of elements S_k, T_i is a set consisting of those elements of the set S_p whose representative is an element from the set T_0 and not yet used. The elements of the set T_i are written to the set T_0 after the elements that have already been used. The process of constructing sets T_i continues until some element is found that belongs to some set from $S_1,\ldots,S_k$ with representatives $a_1,\ldots,a_k$. In parallel, a set *Q* is constructed, whose elements are the numbers of the sets from which the elements were added to T_0. With the completion of the construction of the sets T_0 and *Q,* the *SDR* is rebuilt and a transition is made to the consideration of the next subset.

For matrix (2), following Algorithm 3, we obtain representatives *a={1,3,2,4,7}* for the first five sets (represented by the first five rows). For the sixth set, S_6 *={1,2,3}*, it is not possible to find a representative, since all its elements are representatives of other sets. However, in this situation, one can also find a representative for S_6 using the following function included in Algorithm 3. Let us compose the initial set T_0 from the elements of S_6. The corresponding set is *Q={6,6,6}*. We include element *7* from the first set in T_0 and correct *Q*. We get T_0 *={1,2,3,7}* and *Q={6,6,6,1}*. After the second adjustment, T_0 and Q are T_0 *={1,2,3,7,5}* and *Q={6,6,6,1,2}*. Now we rebuild the *SDR*. As a result, a representative is determined for the set S_6, and in general, for six sets, after the restructuring, we obtain a set of representatives a and a set *P* of the corresponding numbers of the sets S_i, *i=1,2,...,6*, i.e.

a = {3,5,1, 2,4,7},

P = {6,2,1,3,4,5}.

The main part of the *SDR* reforming algorithm 3 is represented using MatLab (Attaway Stormy, 2019). The function is applied with the following data: the set T_0 consists of *t* elements S_{k+1} that have been used as representatives of other sets; *rs=k+1*; *Q* – consists of the numbers of the sets whose elements are included in T_0, $a=\left(a_1,a_2,\ldots a_k\right)$ и $P=\left(P_1,P_2,\ldots P_k\right)$ are such that a_i is a representative for the set with number P_i, *i=1,2,...,k*; the S=*{S(i1,j1)}* matrix consists of elements, the rows of which represent the sets $S_1,S_2,\ldots S_n$; *prvd* – a sign of the possibility of extending the set of representatives *a* by the representative S_{k+1}.

For the FileSDR.m script, the input is:

rs=6;

k=5; a=[1 3 2 4 7]; P=[1 2 3 4 5]; % Representatives *a* for *k* sets *P*

t=3; T0=[1 2 3]; Q=[6 6 6].

Transformations of the input data and the results obtained are commented:

FileSDR.m

```
for i1=1:rs-1
   prvd=0;

   for j1=1:n s=S(i1,j1);
       if s==0 continue; end
       % Checking whether an element s belongs to the set
       h1=0;
       for k1=1:t
          if s~=T0(k1) h1=h1+1; end
       end

       if h1<t continue; end
       if h1==t h2=0;    % Element s does not belong to
          for k2=1:k
            % Checking if element s matches one from the
            % representatives of the set a.
            if s~=a(k2) h2=h2+1; end
          end
       end

       if h2<k t=t+1;
         % Coincidence of the element s with one of the
         % representatives of a and the inclusion of s in
           T0=[T0 s]; Q=[Q i1];
       end

       if h2==k
           % s does not match with none of the representatives a
          t=t+1;
          T0=[T0 s]; Q=[Q i1]; prvd=1;
          break;
       end
   end   %   j1

   if prvd==1
      % For   sets, there is an SDR in the form (a,P)
      % The vectors ap and Tp are introduced, which are used
      % in the formation of the SDR
      ap(1:k)=1; Tp(1:t)=0;
      xQ=Q(t); yT=T0(t); Tp(t)=1;
      PR=t+k;

      for h=1:PR
```

```
          pr=0;
           for k1=k:-1:1
               if xQ==P(k1)
                   xp=P(k1); ya=a(k1); ap(k1)=0; pr=1; break;
               end
           end
           if pr==0 break; end

           if pr==1
               for t1=t:-1:1
                   if T0(t1)==ya
                       xQ=Q(t1); yT=T0(t1); Tp(t1)=1;
                   end
               end
           end
        end     % h

        % Sets ap and Tp are defined
        % and an extended SDR of the form (an, Pn) is formed
        r=0;
        for k1=1:t
           if Tp(k1)==1
               r=r+1; an(r)=T0(k1); Pn(r)=Q(k1);
           end
        end
        for k1=1:k
           if ap(k1)==1
               r=r+1; an(r)=a(k1); Pn(r)=P(k1);
           end
        end
        k=r; P=[Pn]; a=[an];
        fprintf('\n SDR is defined for the sets S1,...,S6:');
        fprintf('\n a='); for ic=1:k fprintf(' %d',a(ic)); end
        fprintf('\n P='); for ic=1:k fprintf(' %d',P(ic)); end
        fprintf('\n');
        break;
    end % prvd==1
end   % i1
```

As a result of the execution of this script:

>> FileSDR

SDR is defined for the sets $S_1,...,S_6$:

a= 3 5 1 2 4 7

P= 6 2 1 3 4 5
>>

If the set T_0, after the completion of the construction of the sets T_i, will consist of representatives of the sets $S_1,\ldots,S_k$, and *k<n*, we conclude that there is no *SDR* for the sets $S_1,\ldots,S_{k+1}$ due to the absence of a representative for a separate set S_{k+1}. Thus, using Algorithm 3, one can construct a complete *SDR* or establish non-existence of such for given sets S_i, *i=1,2,…n*.

3.4 Algorithm 4. Search for *SDR* reduces to solving a specially constructed optimization problem

To do this, the matrix *C*, representing the sets S_i with elements from $J=\{1,2,...,n\}$, is replaced by the matrix S,

$$S=\{s_{ij}\},$$

where $s_{ij}=1$, if the *i*-th set S_i contains an element *j* from *J*, otherwise, $s_{ij}=M$, and *M* is a sufficiently large positive number.

We solve the following function minimization problem

$$f=\sum_{i=1}^{n}\sum_{j=1}^{n}s_{ij}x_{ij}$$

provided that only one element x_{ij} is *1*, the remaining elements of the row or column are *0*.

Based on the obtained optimal solution, *SDR* is constructed for sets S_i, *i=1,…,n*. Let us assume that the found optimal solution has the form

$$x=\{x_{i_1j_1},x_{i_2j_2},...,x_{i_{1n}j_n}\}$$

Then the coordinate $x_{i_kj_k}$ is 1 and the corresponding value $s_{i_kj_k}$ is 1 or M. If $s_{i_kj_k}=1$, then the element of the row i_k in the column j_k of the matrix C is considered to be a representative of the set. $s_{i_kj_k}=1$. All such representatives constitute the maximum possible SDR. If $s_{i_kj_k}=M$, then the set S_{i_k} has no representative. The optimal value is f=k+(n-k)*M in accordance with the fact that k sets will have representatives and n-k sets will not have representatives.

3.5 Algorithm 5. SDR search associated with the construction of cycles

1) We build the matrix *S* based on the sets $S_1,S_2,\ldots,S_n$;

2) As a representative of the set S_i from the group of sets S_{i_1j}, $S_{i_2j},\ldots,S_{i_kj}$ with one element in the row is and in the same column *j*, we take an element from row *i*; delete all lines;

S_{i_1j}, $S_{i_2j},\ldots,S_{i_kj}$.

3) Similarly, we look through columns with one element and as a representative of the set S_j from the group of sets S_{ij_1}, $S_{ij_2},\ldots,S_{ij_s}$, we take an element from column *j* ; cross out all columns S_{ij_1}, $S_{ij_2},\ldots,S_{ij_s}$;

4) For the remaining elements of the matrix *S* after deleting all rows and all columns with one element , we build a cycle with pairs *(r, s)* of the following form

$$(i_1,j_1),\ (j_1,i_2),\ (i_2,j_2),\ (j_2,i_3),\ldots,(j_r,i_s) \quad (3)$$

5) From the odd pairs of the constructed cycle, we take for the sets S_r the representative s_{rs}. Cross out all rows of the matrix *S* corresponding to the sets S_r;

6) From the remaining elements of the matrices S, we build another cycle (if any) and select representatives for the remaining rows;

7) Repeat steps 4)-5) until all rows of the matrix *S* are crossed out.

As a result of Algorithm 5, we obtain a system of sets with representatives, as well as sets that do not have representatives, since they do not correspond to the Sperner criterion. In this algorithm, it is possible to select representatives from the cycle and, according to another option, i.e. choice of representatives of sets from even pairs of cycle (3). Thus we get another system of different representatives.

4. Conclusions

The presented algorithms can be used in solving problems from various fields of research related to the choice of their solution from several alternative options. The algorithms are presented without taking into account the features of the objects that form the initial subsets of the finite set, which ensures their universality. They can be used in transport-type problems, in planning, in the formation of independent expert groups, in resource allocation, and in other studies.

References

Attaway Stormy. (2019). MATLAB A Practical Introduction to Programming and Problem Solving, Elsevier, Butterworth-Heinemann, Oxford-Camdridge. https://doi.org/10.1016/B978-0-12-815479-3.00003-9

Brualdi, R.A. (2009). Introductory Combinatorics, Pearson India Education Service.

Hall, M. (1967). Combinatorial Theory, Blaisdell Publishing Company, Waltham -Toronto-London.

Jukna, S. (2011). Extremal Combinatorics. Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/978-3-642-17364-6

Korte, B., & Vygen, J. (2015). Combinatorial optimization Theory and Algorithms, Springer-Verlag Berlin Heidelberg.

Lipski, W. (1982). Kombinatoryka dla programistow, Wydawnictwa naukowo-techiczne Warszawa.

Morris, J. (2023). Combinatorics from https://www.cs.uleth.ca/~morris/Combinatorics.

Rybnikov, K. A. (1985). Introduction to combinatorial analysis, 2nd ed, Moscow, Moscow University Press, (russ.).

Wilson, R. (1972). Introduction to graph theory, Oliver and Boyd Edinburgh.