Univ. u Novom Sadu
Zb. Rad. Prirod.–Mat. Fak.
Ser. Mat. 23, 1 (1993), 245 - 257

Review of Research
Faculty of Science
Mathematics Series

A SPACE OPTIMAL ALGORITHM FOR THE ENUMERATION OF MAXIMAL DOUBLE INDEPENDENT SUBSETS OF A GRAPH

Ladislav Novak, Žarko Karadžić
Faculty of Technical Sciences, University of Novi Sad
Trg Dositeja Obradovića 6, 21000 Novi Sad, Yugoslavia

Dragan M. Acketa
Institute of Mathematics, University of Novi Sad
Trg Dositeja Obradovića 4, 21000 Novi Sad, Yugoslavia

Abstract

A space optimal algorithm is introduced for generating all the maximal simultaneously circuitless and cutsetless edge-subsets of a 2-connected graph G. The algorithm makes use of three boolean functions, which test whether a candidate for such a maximal double independent set is valid. The efficiency of the tests is due to the fact that they use local search only; one need not produce the families of all the circuits and cutsets of G.

AMS Mathematics Subject Classification (1991): 05C38
Key words and phrases: circuit, cutset, graph, matroid.

1. Preliminaries

We assume familiarity with the basic notions of graph theory such as: graph, vertex, edge, circuit and cutset.

A subset of edges of a graph G is said to be a *double independent* subset if it contains no circuits and no cutsets of the graph G. A double independent set of edges of G is said to be *maximal double independent* (basoid, [3]) if no superset of it is also double independent. These notions were introduced and some of their properities were studied in detail in the paper [1].

It turns out that it suffices (without any loss of generality) to restrict attention to 2-connected graphs G when investigating double independent sets; in more general cases, the double independent subsets belong to 2-connected components of G.

The algorithm for generating maximal double independent sets of edges of 2-connected graphs is based on the definition of double independent sets, as well as on the following observation [1]:

A double independent subset S of edges of a graph is a maximal double independent subset iff every edge in the complement of S forms a circuit or/and a cutset with the edges in S only.

It was also proved in [2] that the maximal possible size of a double independent set is equal to the so-called topological degree of freedom [3].

In what follows, we shall denote the edge-set of a graph G by $E(G)$. Also, a set of cardinality 1 will be denoted as *1-set*.

2. Algorithm

This section, the main one in the paper, includes an algorithm for enumeration of all the maximal double independent subsets of the edge-set of a 2-connected graph G. The algorithm uses directly the definition of maximal double-independent sets. It generates candidates as combinations without repetitions, consisting of d edges of G, where the number d is bounded from above by the topological degree of freedom. Three questions are asked in turn for each candidate C:

Does C contain a cutset ? Does C contain a circuit ?

Will the addition of an edge x cause the set $C \cup \{x\}$ to include either a circuit or a cutset of G ?

All the three tests are solved locally, without producing and keeping in

memory the family of all the circuits and all the cutsets of G.

```
ALGORITHM List_all_maximal_double_independent_sets ;
    Input: 2-connected graph G , with v vertices and e edges.
   Output: All the maximal double independent subsets of the edge-set of
G , without repetitions.

FUNCTION Contains_cutset (Candidate, Graph ): boolean;
    (* Tests whether the current Candidate contains a cutset of Graph *)
    BEGIN
        Remove Candidate from Graph;
                (* New_graph is obtained *)
        Contains_cutset is TRUE if and only if
                        New_graph is not connected
    END;

FUNCTION Contains_circuit ( Candidate, Graph ) : boolean;
    (* Tests whether the current Candidate contains
       a circuit of Graph *)
    BEGIN
        Temporary_graph := Graph;
        REPEAT
            IF there exists a hanging (one vertex of
            which has degree 1) edge h of Temporary_graph
            THEN
                    remove h from Temporary_graph
        UNTIL there are no hanging edges in
                Temporary_graph;
        Contains_circuit is FALSE if and only if
                Temporary_graph has no edges
    END;

FUNCTION Maximal (Candidate, Graph): boolean;
    (* Tests whether the current Candidate, which is known to be
       double - independent within Graph, is a maximal edge-set
       with this feature *)
    BEGIN
        Remove all the edges of Candidate from G ;
```

```
            Denote the obtained graph by G1 ;
            S := the set of all the 1-cutsets of G1 ;
            Contract (identify the end vertices of) all the edges
                    of Candidate from by G2 ;
            Denote the obtained graph by G2 ;
            T := the set of all the 1-circuits of G2 ;
            Maximal is TRUE if and if
                S ∪ T =E(Graph) \ E(Candidate)
    END;

BEGIN (* List_all_maximal_double_independent_sets *)
    card := Topological_degree_of_freedom;
    REPEAT
        REPEAT
            Generate the following Candidate;
            (* candidates are combinations without repetitions
               consisting of card edges among e edges *)
            IF NOT Contains_cutset ( Candidate, G ) THEN
               IF NOT Contains_circuit ( Candidate, G ) THEN
                  IF Maximal (Candidate, G ) THEN
                     output ( Candidate ) ;
               ELSE jump over the combinations which
                     contain that circuit
            ELSE jump over the combinations which
                  contain that subset
        UNTIL there are no more combinations;
        card := card - 1
    UNTIL card = 0;
END; (* List_all_maximal_double_independent_sets *)
```

The mechanism of jumping over is implemented in according with the lexicographical order of candidates. To put it more precise, suppose that a combination $c_1..c_d$ contains a critical subset X (circuit or cutset) $c_{i_1}..c_{i_k}$. The lexicographically next combination is obtained by searching for the lexicographically next possibility for c_{i_k}. In this way are jumped over only those candidates containing the set X, which begin with the sequence $c_1..c_{i_k}$. Thus we may conclude that this jumping over is not complete; nevertheless, it is a useful shortcut.

3. Validity and elaboration of the algorithm

In this section we shall prove that the algorithm is correct and that it is space-optimal. In addition, we shall explain the background of the applied tests (boolean functions) and illustrate their performance by examples.

Theorem 1. *The algorithm given in Section 2. correctly solves the problem, that is, it produces all the maximal double independent sets of edges of the input connected graph G without duplications.*

Proof. The statement of the theorem can be broken into the following three statements:

(1) All the maximal double independent sets of edges of G are generated by our algorithm; that is, no maximal double independent set of edges G can be missed.

(2) The only sets which are output by our algorithm are the maximal double independent sets of edges of G .

(3) Each maximal double independent set of edges of G is generated only once by our algorithm; that is, the algorithm does not produce duplications.

Statements (1) and (3) follow from the fact that the tests are performed on the family of ALL combinations without repetitions, consisting of some d edges of G , for a given cardinality d (when the possible cardinalities d are considered, the proof of statement (1) also requires an application of the above cited results from the paper [2]). The only exceptions are those combinations, which are safely jumped over, due to the fact that they include some already recognized cutset or circuit. The exhaustive list of such combinations is generated in the lexicographical order (which eliminates the possibility that duplications occur), and it is well-known how to produce the lexicographically next combination without missing some other combinations in between. The three tests, which are performed with each candidate, guarantee that nothing but maximal double independent sets of edges of G will appear in the output (statement (2)). □

Theorem 2. *The algorithm is space-optimal.*

Proof. The applied local approach does not use any larger structure, which would oversize the input structure, which is of order $O(n^2)$, where n denotes the number of vertices of G . □

Remark. An attempt to prove the jumping over the combinations (so that ALL the combinations containing a recognized cutset (circuit) are not further visited) would probably lead to a loss of space-optimality (in such a case we would possibly have to keep in memory the recognized sets; otherwise we should give up from the lexicographical order and it is not clear how to generate the combinations properly).

We proceed with a more detailed descriptions and explanations of the three main successive tests, which are applied to each candidate. In addition, we elaborate the auxilliary connectivity test and the construction of auxilliary sets S and T, which are used for the maximality test:

3A. Connectivity test

This test is applied to check whether a subset S of edges of G contains a cutset (this is true iff the set $E(G) \setminus S$ is connected). In addition, this test is used within a loop during the construction of a Set_cutset.

The auxilliary structure used for checking connectivity is the connectivity vector. It is indexed by vertices, while its components are the labels of separate connected components. The input state of the connectivity vector are all zeros. Each new edge unifies the non-zero labels of the components which are connected by the edge (if one label is zero, then it becomes equal to the other one). If both labels are equal to zero, then a new non-zero value is introduced to replace them. An illustration is given by the following example:

Let the graph on the vertex-set $\{1,2,3,4,5,6,7,8,9\}$, with the following edges (= pairs of vertices), listed in order be given: (1,8), (9,1), (7,3), (6,9), (4,2), (1,6), (4,5).

The steps of adjustment of the connectivity vector seem as follows:

new edge \ vertex	1	2	3	4	5	6	7	8	9
	0	0	0	0	0	0	0	0	0
(1,8)	1	0	0	0	0	0	0	1	0
(9,1)	1	0	0	0	0	0	0	1	1
(7,3)	1	0	2	0	0	0	2	1	1
(6,9)	1	0	2	0	0	1	2	1	1
(4,2)	1	3	2	3	0	1	2	1	1
(1,6)	1	3	2	3	0	1	2	1	1
(4,5)	1	3	2	3	3	1	2	1	1

Different labels in the final state of the connectivity vector correspond to the vertices in different connected components.

3B. Local search for an included cutset

The connectivity test is almost sufficient; it should be preceded by the removal of the edges of the tested candidate from the edge-set. The candidate contains a cutset if and only if the graph obtained after this removal is not connected (this is a direct consequence of the definition of a cutset).

3C. Local search for an included circuit

This search is based on the following observation: an edge is circuit-free if and only if it can be eliminated by iterative deletions of hanging edges. Thus the candidate contains a circuit if and only if the resulting graph has no edges, after all the possibilites for iterative deletions of hanging edges are exhausted.

3D. Testing maximality

This stage of the algorithm requires the construction of two auxilliary edge-sets, which are related to the current circuitless and cutsetless candidate $Cand$:

$S =$ the set of all edges x of $E(G) \setminus Cand$ such that the set $Cand \setminus \{x\}$ includes a cutset

$S =$ the set of all edges y of $E(G) \setminus Cand$ such that the set $Cand \cup \{y\}$

includes a circuit

These constructions will use the assumptions that the set $Cand$ is cutsetless and circuitless respectively (in other words, it is essential for these constructions that the double independence of the set $Cand$ is guaranteed). After they are done, there remains to check whether the sets S and T cover the set $E(G)$. If this is not the case, then the double independent set $Cand$ is not maximal double independent.

Remark. Note that the maximality test should be applied solely to those candidates, which are known to be double independent.

3E. A construction of the set S

This construction is based on the following obvious fact:

If the edge-set $Cand$ is cutset-free, then the set $Cand \cup \{x\}$ contains a cutset of the underlying graph G if and only if the set $\{x\}$ is a 1-cutset of the set $E(G) \setminus Cand$ (that is, of the graph which is obtained from G after the removal of edges of $Cand$).

Namely, if the removal of the edge x from the set $E(G) \setminus Cand$ does not produce two connected components, then the same will hold for the removal of the edges in $Cand \cup \{x\}$ from the set G, and conversely.

Thus the construction of the set S requires one call of the connectivity test for each edge of the set $E(G) \setminus Cand$.

Example. Let G denote the graph with vertices A,B,C,D,E,F,G and with edges 1=AB, 2=BC, 3=CD, 4=AD, 5=CE, 6=EF, 7=FG, 8=CG, and let $Cand = \{2,3\}$. After the edges 2 and 3 are removed, one obtains two circuit-free edges (=1-cutsets), 1 and 4, as well as the circuit consisting of the edges denoted by 5, 6, 7, 8. It follows that $S = \{1,4\}$.

3F. A construction of the set T

This construction is based on the following obvious fact:

If the edge-set $Cand$ is circuit-free, then the set $Cand \cup \{x\}$ contains a circuit of the underlying graph G if and only if the set $\{x\}$ is a 1-circuit of the graph obtained from G after the contraction of all the edges belonging to $Cand$ (that is, after the deletion of all these edges and identification of their end-vertices).

Namely, if the edge x completes a circuit, then by iterative contractions of other edges of that circuit one finally obtains that x is transformed to a loop; similarly, if x does not complete a circuit, then its vertices must not be equal after the contractions are done. Thus the conctraction of edges of $Cand$ is responsible for the identification of vertices of the edge x.

Remark. The two facts, which are used in this and in the previous example, are dual to each other in the matroid sense [4]; one of these facts can be obtained from the other by mutual replacement of the matroidally dual notions "circuit" and "cutset", as well as "restriction" (obtained by deleting the edges from a given subset) and "contraction" (obtained by contracting the edges from a given subset).

Example. Let G denote the graph with vertices A,B,C,D,E,F,G,H and with edges 1=AB, 2=BC, 3=CD, 4=DE, 5=EF, 6=FG, 7=GH, 8=AH, 9=BE, 10=BG. In addition, let $Cand = \{1,2,3,7,9,10\}$. The path for the construction of the set T associated to $Cand$ seems as follows:

	List of edges 1,...,10, represented by end-vertices									
Edge	1	2	3	4	5	6	7	8	9	10
	AB	BC	CD	DE	EF	FG	GH	AH	BE	BG
Contract 1	–	AC	CD	DE	EF	FG	GH	AH	AE	AG
Contract 2	–	–	AD	DE	EF	FG	GH	AH	AE	AG
Contract 3	–	–	–	AE	EF	FG	GH	AH	AE	AG
Contract 7	–	–	–	AE	EF	FG	–	AG	AE	AG
Contract 9	–	–	–	AA	AF	FG	–	AG	–	AG
Contract 10	–	–	–	AA	AF	FA	–	AA	–	–

The set T contains exactly all those edges of $E(G) \setminus Cand$, which are finally contracted to loops. In this case, this happens with the edges 4 and 8, so $T = \{4,8\}$. Note that both of the edges 5 and 6 should be added to $Cand$ in order to close a circuit; none of them could do it alone.

Remark. It seems more appropriate to perform one contraction after another and to consider the generated effects, as in the table above. On the contrary, when the construction of the set S is considered, it seems more appropriate to remove all the edges of $Cand$ at once and to treat the edges from $E(G) \setminus Cand$ independently from each other afterwards.

4. Some illustrative examples

We elaborate two complete examples, which were listed in [1].

Example 1.

Let G_1 denote the graph with vertices A,B,C,D and edges 1=AB, 2=AD, 3=BD, 4=BC, 5=BC, 6=CD.

Maximal double independent sets of G_1 are 36, 134, 135, 146, 156, 234, 235, 246 and 256. It is easy to check that none of them includes a cutset of G_1 (some of the sets 12, 136, 236, 456, 1345 and 2345) or a circuit of G_1 (some of the sets 45, 123, 346, 356, 1246 and 1256).

In order to justify that the mentioned double independent sets are maximal, we show the circuits and cutsets which arise with some of them, after one new element is added:

Double independent set 36 is maximal since:
the supersets 136 and 236 are cutsets,
while the supersets 346 and 356 are circuits.

Sets S and T are equal to 12 and 45 respectively in this case.

Double independent set 134 is maximal since:
the supersets 1234 contains circuit 123 and cutset 12;
the supersets 1345 contains circuit 45 and is itself a cutset ;
the supersets 1346 contains circuit 346 and cutset 136.

Both sets S and T are equal to 256 in this case.

In a similar fashion we may proceed with the remaining seven maximal double independent sets.

It can be shown that the topological degree of freedom of graph G_1 is equal to 3.

Trace of the algorithm:

Candidate			
123	contains	cutset 12	(jump over 124 and 125)
134	is double	independent	and maximal
135	is double	independent	and maximal
136	contains	cutset 136	
145	contains	circuit 45	
146	is double	independent	and maximal
156	is double	independent	and maximal
234	is double	independent	and maximal
235	is double	independent	and maximal
236	contains	circuit 236	
245	contains	circuit 45	
246	is double	independent	and maximal
256	is double	independent	and maximal
345	contains	circuit 45	
346	contains	circuit 346	
356	contains	circuit 356	
456	contains	circuit 45	

In accordance with the remark given in the algorithm, note that the combinations 124, 125 and 126 were not visited, due to the former stretch of the cutset 12. On the contrary, four candidates were rejected due to the same circuit 45. This circuit does not belong to lexicographically the first part of the candidate and therefore it could not be used for a safe shotcut of generating combinations.

Example 2. Let G_2 denote the graph on 8 vertices and 14 edges, obtained by double joining two copies of the graph G_1 considered in the previous example, in the following manner:

The vertices of G_2 are denoted A, B, C, D, E, F, G, H.

The edges of G_2 are

1=AB, 2= AH, 3=BH, 4=BC, 5=BC, 6=CH, 7=GH,
8=CD, 9=DG, 10=FG, 11=DF, 12=DE, 13=DE, 14=EF.

There are two cardinalities (5 and 6) of maximal double independent sets of G_2 , with 8, respectively with 80, double independent sets.

We shall give the complete list of maximal double independent sets of edges, corresponding to the graph G_2 :

Maximal double independent sets of cardinality 5:

1	3	6	11	14	2	3	6	11	14	3	6	7	11	14	
3	6	8	11	14	3	6	9	10	12	3	6	9	10	13	
3	6	9	10	14	3	6	9	11	14						

Maximal double independent sets of cardinality 6:

1	3	4	7	11	14	1	3	4	8	11	14	1	3	4	9	10	12
1	3	4	9	10	13	1	3	4	9	10	14	1	3	4	9	11	14
1	3	5	7	11	14	1	3	5	8	11	14	1	3	5	9	10	12
1	3	5	9	10	13	1	3	5	9	10	14	1	3	5	9	11	14
1	3	6	9	11	12	1	3	6	9	11	13	1	3	6	9	12	14
1	3	6	9	13	14	1	3	6	10	11	12	1	3	6	10	11	13
1	3	6	10	12	14	1	3	6	10	13	14	1	3	6	7	11	14
1	4	6	8	11	14	1	4	6	9	10	12	1	4	6	9	10	13
1	4	6	9	10	14	1	4	6	9	11	14	1	5	6	7	11	14
1	5	6	8	11	14	1	5	6	9	10	12	1	5	6	9	10	13
1	5	6	9	10	14	1	5	6	9	11	14	2	3	4	7	11	14
2	3	4	8	11	14	2	3	4	9	10	12	2	3	4	9	10	13
2	3	4	9	10	14	2	3	4	9	11	14	2	3	5	7	11	14
2	3	5	8	11	14	2	3	5	9	10	12	2	3	5	9	10	13
2	3	5	9	10	14	2	3	5	9	11	14	2	3	6	9	11	12
2	3	6	9	11	13	2	3	6	9	12	14	2	3	6	9	13	14
2	3	6	10	11	12	2	3	6	10	11	13	2	3	6	10	12	14
2	3	6	10	13	14	2	4	6	7	11	14	2	4	6	8	11	14
2	4	6	9	10	12	2	4	6	9	10	13	2	4	6	9	10	14
2	4	6	9	11	14	2	5	6	7	11	14	2	5	6	8	11	14
2	5	6	9	10	12	2	5	6	9	10	13	2	5	6	9	10	14
2	5	6	9	11	14	3	6	7	9	11	12	3	6	7	9	11	13
3	6	7	9	12	14	3	6	7	9	13	14	3	6	7	10	11	12
3	6	7	10	11	13	3	6	7	10	12	14	3	6	7	10	13	14
3	6	8	9	11	12	3	6	8	9	11	13	3	6	8	9	12	14
3	6	8	9	13	14	3	6	8	10	11	12	3	6	8	10	11	13
3	6	8	10	12	14	3	6	8	10	13	14						

References

[1] Novak, L., Gibbons, A., Basoids and Double Independence in Graphs, Research Report of the Department of Computer Science, University of Warwick, 1990.

[2] Sengoku, M., Hybrid trees and hybrid tree graphs, IEEE Trans. on Circuit and Systems, vol. CAS-22, pp. 786-790, 1975.

[3] Ohtsuki, T., Ishiziaki, Y., Watanabe, H., Topological degrees of freedom and mixed analysis of electric networks, IEEE Trans. on Circuit Theory, CT-17, 491-499, 1970.

[4] Welsh, D.J.A., Matroid Theory, London Math. Soc. Monographs, no. 8., Academic Press, London, 1976.

REZIME

MEMORIJSKI OPTIMALAN ALGORITAM ZA NABRAJANJE MAKSIMALNIH DUPLO NEZAVISNIH SKUPOVA GRANA U GRAFU

Predložen je jedan memorijski optimalan algoritam za generisanje svih maksimalnih podskupova skupa grana nekog 2-povezanog grafa, koji (istovremeno) ne sadrže ni konture, ni snopove (preseke). Algoritam uključuje tri bulove funkcije, pomoću kojih se testira ispravnost kandidata za takve maksimalne duplo nezavisne skupove. Efikasnost testova sledi iz činjenice da oni koriste samo lokalna pretraživanja; ne moraju se generisati familije svih kontura i snopova grafa.

Received by the editors April 3, 1990