COMBINATORIAL GENERATION OF MATROID REPRESENTATIONS: THEORY AND PRACTICE

PETR HLINĚNÝ*

Faculty of Informatics,
Masaryk University in Brno,
Botanická 68a, 602 00 Brno, Czech Rep.
E-mail: hlineny@fi.muni.cz

Matroids (also called combinatorial geometries) present a strong combinatorial generalization of graphs and matrices. Unlike isomorph-free generation of graphs, which has been extensively studied both from theoretical and practical points of view, not much research has been done so far about matroid generation. Perhaps the main problem with matroid generation lies in a very complex internal structure of a matroid. That is why we focus on generation of suitable matroid representations, and we outline a way how to exhaustively generate matroid representations over finite fields in reasonable computing time. In particular, we extend here some enumeration results on binary (over the binary field) combinatorial geometries by Kingan et al. We use the matroid generation algorithm of [P. Hliněný, *Equivalence-Free Exhaustive Generation of Matroid Representations*] and its implementation in MACEK; see http://www.cs.vsb.cz/hlineny/MACEK.

1. Introduction

Matroids, introduced by Whitney in 1935, present a common generalization of the concept of independency in graphs and matrices. We follow Oxley [10] in our matroid terminology, and we refer the reader to the full version of this paper for explanations.

In graph theory, one often uses pictures to visualize particular graphs. On contrary, such visualization is very difficult in matroid theory – it is almost impossible to give a "nice drawing" of a general matroid in rank higher than 3. That fact makes matroid research more difficult, and brings a strong need for an automated matroid generation routine. (It is often such that proving a theorem in structural matroid theory requires one to check many small cases by hand. As matroid researchers know themselves,

*Supported by the Institute of Theoretical Computer Science ITI, project no. 1M0545.

checking the "small cases" can be quite long and painful, and prone to errors.) However, great complexity and enormous numbers of (even small) matroids makes the task much harder than generation of, say, graphs.

A promising tractable approach is to generate matroid representations as matrices. Matroids represented by matrices over finite fields play an important role in structural matroid theory, similar to the role that graphs embedded on a surface play in structural graph theory. Hence exhaustive generation routines for matroid representations over finite fields could have important applications in matroid research. However, the history of matroid enumeration is rather short. We refer to [8] for a nice overview and bibliography. In particular, we mention here the old work of Acketa [1] on enumeration and construction of small binary matroids, which is closely related to further mentioned work [8] and to our contribution in Section 3.

Perhaps the main problem with matroid generation lies in a very complex internal structure of a matroid – a single matroid on n elements carries an amount of information exponential in n. That is why we are looking for generation of suitable matroid representations which would be easier to handle. Incidently, such are matrix representations of matroids over finite fields, which happen to be interesting and useful in matroid structure theory. We refer the works of Dharmatilake [2] and Kingan [7] as two examples of tasks of matroid-representations generation for the purpose of obtaining theoretical structural results.

Dharmatilake [2] used computer generation of binary matroid representations in order to find the binary excluded minors for matroids of branch-width three. He, in fact, found all 10 (3 new) of them, but his search was not finished. We have completed [4] the search exhaustively using our (faster) generation routine; and moreover, we have found [5] all the ternary excluded minors for branch-width three and generated many more such excluded minors over larger fields.

Among some recent works we cite Kingan et al. [8], and (unpublished) Pendavingh [11]. Kingan et al. present a generation algorithm for matroid representations, which we compare with our generation routine in Section 2, and we significantly extend their enumeration results in Section 3. On the other hand, Pendavingh systematically generates matroids in general (as set systems), which is computationally very hard. The point of our interest in his work is that some nontrivial outcomes of his search (e.g. the small excluded minors for representability over $GF(5)$ and $GF(7)$) agree with results we can obtain with our tools.

2. Matroid Extension Generation Algorithm

A simple approach to combinatorial generation of certain objects is the following: Exhaustively construct all possible "presentations" of the objects of given size (called [9] also labeled objects or labeled representations), and then select one representative of each isomorphism class by means of an isomorphism tester. Since there are typically many presentations for each one resulting object, and the procedure requires to isomorph-test each pair of the constructed presentations, that approach quickly becomes infeasible. A better solution is provided by the technique of a "canonical construction path" which has been explicitly formulated by McKay [9], and used (explicitly or implicitly) in many combinatorial searches.

The idea of canonical construction path is briefly described here: Select a small *base* object. Then, out of all ways how to construct our big object by single-element steps from the base object (*construction paths*), define the lexicographically smallest one (the *canonical* construction path). Adapt the construction process so that all non-canonical extensions at each step are thrown away immediately (assuming that the canonical lexicographic order is hereditary). In this way every object is generated only once, and no explicit isomorphism tests are necessary. (Though the isomorphism problem may be implicitly contained in the definition of the canonical order.)

It appears that the canonical construction path technique has not been fully implemented before in matroid generation. The works mentioned in Section 1 usually use variations of the above simple generation method. Although Kingan et al. [8] claim to use the canonical construction path method, they implement it only partially — there is no definition of a canonical order, but an isomorphism tester is used to select the representatives of one-element matroid extensions.

The main theoretical contribution [6] of our work is in a complete adaptation of the canonical construction path technique for exhaustive generation of matroid representations over finite fields. (We can, moreover, generate matroid representations over so called [12] "partial fields".) The adaptation is not at all obvious due to specific difficulties with matroid representations. Our base object is a common minor of all generated matroid representations, and one-element extensions and coextensions play the role of single-element steps. Altogether, the generation steps form what we call an *elimination sequence* of a generated matroid representation.

3. Practical Computations

We have used an implementation of our algorithm in MACEK [3] for carrying out several successful matroid generation projects, e.g. [4,5]. (MACEK implements also numerous other matroid structural computations, like minor, connectivity, or isomorphism tests, generation of representations, etc.)

Our results concerning enumeration of binary combinatorial geometries (i.e. of simple binary matroids) are summarized in Table 1. We have independently verified all the enumeration results of Kingan et al. [8] on up to 11 elements, and moreover we have added significant new results for matroids on 12 and 13 elements. For instance, the number of 8-element rank-5 simple binary matroids can be obtained by running MACEK with `'GF(2)' '@ext-simple;!extendsize 5 3' '1;1'`.

```
MACEK 1.2.09 (26/08/05) starting...
      ......
~652~ In total 15 (co-)extensions of 1 matrix-sequence generated over GF(2).
```

(One needs the latest version $\geq$1.2.09 of MACEK to run this.)

Table 1.

An enumeration of small simple binary matroids (* new entries).

*rk.**el.*	2	3	4	5	6	7	8	9	10	11	12	13
2	1	1	0	0	0	0	0	0	0	0	0	0
3		1	2	1	1	1	0	0	0	0	0	0
4			1	3	4	5	6	5	4	3	2	1
5				1	4	8	15	29	46	64	89	* **112**
6					1	5	14	38	105	273	* **700**	* **1794**
7						1	6	22	80	312	* **1285**	* **5632**
8							1	7	32	151	* **821**	* **5098**
9								1	8	44	266	* **1948**
10									1	9	59	* **440**
11										1	10	* **76**
12											1	11
13												1

For interested readers, we add a short summary of real running times of our enumerations (after some tuning-up). All the results on at most 11 elements are quite easy, they take just seconds or at most minutes to finish. For 12 elements, the computation took approximately 3 hours (rank 6), 13 hours (rank 7), and 16 hours (rank 9); for 13 elements it took 11 hours (rank 6), 110 hours (rank 7), 260 hours (rank 8), 300 hours (rank 9), and

140 hours (rank 10). (Computing times are normalized to 1GHz CPU.)

4. Conclusions

We have presented a matroid extension generation scheme which fully implements the idea of a generation via a canonical construction path by McKay. Compared with other matroid generation programs, the implementation of our algorithm appears to be remarkably faster and more powerful in generation of matroids representable over finite fields. We have already used it to prove some results of theoretical interest, and significantly extended known enumeration results of small matroids. In current work we are extending the enumeration in Table 1 to ternary and other representable matroids. (For instance, we have so far verified the ternary results of [8].)

Acknowledgments

The large-scale computations summarized in Table 1 have been run on the `minos` computing cluster at West Bohemia University (ITI center).

References

1. D.M. Acketa, *A construction of non-simple matroids on at most 8 elements*, J. Combin. Inform. System Sci. 9, 121–132.
2. J.S. Dharmatilake, *Binary Matroids of Branch-Width* 3, PhD. dissertation, Ohio State University, 1994.
3. P. Hliněný, *The* MACEK *Program*, `http://www.cs.vsb.cz/hlineny/MACEK`, 2002–2005, version 1.2.09. (With an online trial interface.)
4. P. Hliněný, *On the Excluded Minors for Matroids of Branch-Width Three*, Electronic Journal of Combinatorics 9 (2002), #R32.
5. P. Hliněný, *Using a Computer in Matroid Theory Research*, Acta Math. Univ. M. Belii 11 (2004), 27–44.
6. P. Hliněný, *Equivalence-Free Exhaustive Generation of Matroid Representations*, submitted.
7. S.R. Kingan, Matroid Structure, Ph.D. dissertation, 1994, Louisiana State University.
8. R.J. Kingan, S.R. Kingan, W. Myrvold, *On matroid generation*, Proceedings of the 34th Southeastern Int. Conf. on Combinatorics, Graph Theory, and Computing, Congressus Numerantium 164 (2003), 95–109.
9. B.D. McKay, *Isomorph-free exhaustive generation*, J. Algorithms 26 (1998), 306–324.
10. J.G. Oxley, Matroid Theory, Oxford University Press 1992.
11. R. Pendavingh, personal communication, 2004.
12. C. Semple, G.P. Whittle, *Partial Fields and Matroid Representation*, Advances in Appl. Math. 17 (1996), 184–208.