Matroids You Have Known

DAVID L. NEEL
Seattle University
Seattle, Washington 98122
neeld@seattleu.edu

NANCY ANN NEUDAUER
Pacific University
Forest Grove, Oregon 97116
nancy@pacificu.edu

> Anyone who has worked with matroids has come away with the conviction that matroids are one of the richest and most useful ideas of our day.
>
> —Gian Carlo Rota [**10**]

Why matroids?

Have you noticed hidden connections between seemingly unrelated mathematical ideas? Strange that finding roots of polynomials can tell us important things about how to solve certain ordinary differential equations, or that computing a determinant would have anything to do with finding solutions to a linear system of equations. But this is one of the charming features of mathematics—that disparate objects share similar traits. Properties like independence appear in many contexts. Do you find independence everywhere you look? In 1933, three Harvard Junior Fellows unified this recurring theme in mathematics by defining a new mathematical object that they dubbed *matroid* [**4**]. Matroids are everywhere, if only we knew how to look.

What led those junior-fellows to matroids? The same thing that will lead us: Matroids arise from shared behaviors of vector spaces and graphs. We explore this natural motivation for the matroid through two examples and consider how properties of independence surface. We first consider the two matroids arising from these examples, and later introduce three more that are probably less familiar. Delving deeper, we can find matroids in arrangements of hyperplanes, configurations of points, and geometric lattices, if your tastes run in that direction.

While tying together similar structures is important and enlightening, matroids do not reside merely in the halls of pure mathematics; they play an essential role in combinatorial optimization, and we consider their role in two contexts, constructing minimum-weight spanning trees and determining optimal schedules.

What's that, you say? Minimum-weight what? The mathematical details will become clear later, but suppose you move your company into a new office building and your 25 employees need to connect their 25 computers to each other in a network. The cable needed to do this is expensive, so you want to connect them with the least cable possible; this will form a minimum-weight spanning tree, where by *weight* we mean the length of cable needed to connect the computers, by *spanning* we mean that we reach each computer, and by *tree* we mean we have no redundancy in the network. How do we find this minimum length? Test all possible networks for the minimum total cost? That would be $25^{23} \approx 1.4 \times 10^{32}$ networks to consider. (There are n^{n-2} possible trees on n vertices; Bogart [**2**] gives details.) A computer checking one billion configurations per second would take over a quadrillion years to complete the task. (That's 10^{15} years—a very long time.) Matroids provide a more efficient method.

Not only are matroids useful in these optimization settings, it turns out that they are the very characterizations of the problems. Recognizing that a problem involves a matroid tells us whether certain algorithms will return an optimal solution. Knowing that an algorithm effects a solution tells us whether we have a matroid.

In the undergraduate curriculum, notions of independence arise in various contexts, yet are often not tied together. Matroids surface naturally in these situations. We provide a brief, accessible introduction so that matroids can be included in undergraduate courses, and so that students (or faculty!) interested in matroids have a place to start. For further study of matroids, please see Oxley's *Matroid Theory* [**9**], especially its 61-page chapter, *Brief Definitions and Examples*. Only a cursory knowledge of linear algebra and graph theory is assumed, so take out your pencil and work along.

Declaration of (in)dependence

In everyday life, what do we mean by the terms *dependence* and *independence*? In life, we feel dependent if there is something (or someone) upon which (or whom) we must rely. On the other hand, independence is the state of self-sufficiency, and being reliant upon nothing else. Alternatively, we consider something independent if it somehow extends beyond the rest, making new territory accessible, whether that territory is physical, intellectual, or otherwise. In such a case that independent entity is necessary for access to this new territory.

But we use these terms more technically in mathematics, so let us connect the colloquial to the technical by considering two examples where we find independence.

Linear independence of vectors The first and most familiar context where we encounter independence is linear algebra, when we define the linear independence of a set of vectors within a particular vector space. Consider the following finite collection of vectors from the vector space $\mathbb{R}^3$ (or $\mathbb{C}^3$ or $(\mathbb{F}_3)^3$):

$$v_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, v_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad v_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad v_4 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix},$$

$$v_5 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad v_6 = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}, \quad v_7 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

It is not difficult to determine which subsets of this set are *linearly independent* sets of vectors over $\mathbb{R}^3$: subsets in which it is impossible to represent the zero vector as a nontrivial linear combination of the vectors of the subset. To put it another way, no vector within the subset relies upon any of the others. If some vector *were* a linear combination of the others, we would call the set of vectors *linearly dependent*. Clearly, this means v_7 must be excluded from any subset aspiring to linear independence.

Let us identify the *maximal independent sets*. By *maximal* we mean that the set in question is not properly contained within any other independent set of vectors. We know that since the vector space has dimension 3, the size of such a maximal set can be no larger than 3; in fact, we can produce a set of size 3 immediately, since $\{v_1, v_2, v_3\}$ forms the standard basis. It takes little time to find $\mathcal{B}$, the complete set of maximal independent sets. The reader should verify that $\mathcal{B}$ is

$$\{\{v_1, v_2, v_3\}, \{v_1, v_2, v_4\}, \{v_1, v_2, v_5\}, \{v_1, v_3, v_5\}, \{v_1, v_4, v_5\},$$
$$\{v_2, v_3, v_4\}, \{v_2, v_3, v_6\}, \{v_2, v_4, v_5\}, \{v_2, v_4, v_6\},$$
$$\{v_2, v_5, v_6\}, \{v_3, v_4, v_5\}, \{v_3, v_5, v_6\}, \{v_4, v_5, v_6\}\}.$$

Note that each set contains exactly three elements. This will turn out to be a robust characteristic when we expand the scope of our exploration of independence.

We know from linear algebra that every set of vectors has at least one maximal independent set. Two other properties of $\mathcal{B}$ will prove to be important:

- No maximal independent set can be properly contained in another maximal independent set.
- Given any pair of elements, $B_1, B_2 \in \mathcal{B}$, we may take away any v from B_1 and there is some element $w \in B_2$ such that $(B_1 \setminus v) \cup w$ is in $\mathcal{B}$.

The reader is encouraged to check the second property in a few cases, but also strongly encouraged not to bother checking all $\binom{10}{2} = 45$ pairs of maximal sets. (A modest challenge: Using your linear algebraic expertise, explain why this "exchange" must be possible in general.)

Notice that we used only seven vectors from the infinite set of vectors in $\mathbb{R}^3$. In general, given any vector space, we could select some finite set of vectors and then find the maximal linearly independent subsets of that set of vectors. These maximal sets necessarily have size no larger than the dimension of the vector space, but they may not even achieve that size. (Why not?) Whatever the size of these maximal sets, they will always satisfy the two properties listed above.

Graph theory and independence Though not as universally explored as linear algebra, the theory of graphs is hardly a neglected backwater. (West [**11**] and Wilson [**15**] give a general overview of basic graph theory.) We restrict our attention to connected graphs. There are two common ways to define independence in a graph, on the vertices or on the edges. We focus on the edges. What might it mean for a set of edges to be independent?

Revisiting the idea of independence being tied to necessity, and the accessibility of new territory, when would edges be necessary in a connected graph? Edges exist to connect vertices. Put another way, edges are how we move from vertex to vertex in a graph. So some set of edges should be considered independent if, for each edge, the removal of that edge makes some vertex inaccessible to a previously accessible vertex.

Consider the graph in FIGURE 1 with edge set $E = \{e_1, e_2, \ldots, e_7\}$.

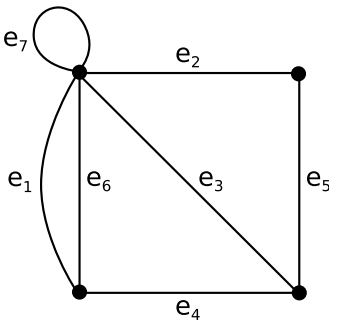

Figure 1 Connected graph G

Now, consider the subset of edges $S = \{e_1, e_3, e_4, e_5\}$. Is this an independent set of edges? No, because the same set of vertices are connected to one another even if, for example, edge e_3 were removed from S. Note that the set S contains a cycle. (A *cycle* is a closed path.) Any time some set of edges contains a cycle, it cannot be an independent set of edges. This also means $\{e_7\}$ is not an independent set, since it is itself a cycle; it doesn't get us anywhere new.

In any connected graph, a set of edges forming a tree or forest (an acyclic subgraph) is independent. This makes sense two different ways: first, a tree or forest never contains a cycle; second, the removal of any edge from a tree or forest disconnects some vertices from one another, decreasing accessibility, and so every edge is necessary. A maximal such set is a set of edges containing no cycles, which also makes all vertices accessible to one another. This is called a *spanning tree*. There must be at least one spanning tree for a connected graph. Here is the set, $\mathcal{T}$, of all spanning trees for G:

$$\begin{aligned}\mathcal{T} = \{&\{e_1, e_2, e_3\}, \{e_1, e_2, e_4\}, \{e_1, e_2, e_5\}, \{e_1, e_3, e_5\}, \{e_1, e_4, e_5\},\\ &\{e_2, e_3, e_4\}, \{e_2, e_3, e_6\}, \{e_2, e_4, e_5\}, \{e_2, e_4, e_6\},\\ &\{e_2, e_5, e_6\}, \{e_3, e_4, e_5\}, \{e_3, e_5, e_6\}, \{e_4, e_5, e_6\}\}.\end{aligned}$$

Here again we see that all maximal independent sets must have the same size. (How many edges are there in a spanning tree of a connected graph on n vertices?)

Spanning trees also have two other important traits:

- No spanning tree properly contains another spanning tree.
- Given two spanning trees, T_1 and T_2, and an edge e from T_1, we can always find some edge f from T_2 such that $(T_1 \setminus e) \cup f$ will also be a spanning tree.

To demonstrate the second condition, consider the spanning trees T_1 and T_2 shown as bold edges of the graph G in FIGURE 2.

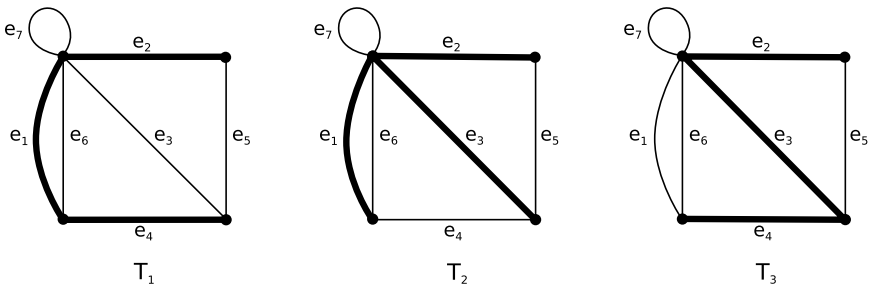

Figure 2 Three spanning trees of G

Suppose we wanted to build a third spanning tree using the edges from T_1 except e_1. Then we must be able to find some edge of T_2 that we can include with the leftover edges from T_1 to form the new spanning tree T_3. We can, indeed, include edge e_3 to produce spanning tree T_3, also shown in FIGURE 2. This exchange property would hold for any edge of T_1.

Motivated by our two examples, now is the proper time for some new terminology and definitions to formally abstract these behaviors.

Thus, matroids

As you notice these similarities between the spanning trees of a graph and the maximal independent sets of a collection of vectors, we should point out that you are not alone. In the 1930s, H. Whitney [**13**], G. Birkhoff [**1**], and S. Maclane [**8**] at Harvard and B. L. van der Waerden [**12**] in Germany were observing these same traits. They noticed these properties of independence that appeared in a graph or a collection of vectors, and wondered if other mathematical objects shared this behavior. To allow for the possibility of other objects sharing this behavior, they defined a matroid on *any* collection of elements that share these traits. We define here a matroid in terms of its maximal independent sets, or *bases*.

The bases A *matroid* M is an ordered pair, $(E, \mathcal{B})$, of a finite set E (the *elements*) and a nonempty collection $\mathcal{B}$ (the *bases*) of subsets of E satisfying the following conditions, usually called the *basis axioms*:

- No basis properly contains another basis.
- If B_1 and B_2 are in $\mathcal{B}$ and $e \in B_1$, then there is an element $f \in B_2$ such that $(B_1 \setminus e) \cup f \in \mathcal{B}$.

The bases of the matroid are its maximal independent sets. By repeatedly applying the second property above, we can show that all bases have the same size.

Returning to our examples, we can define a matroid on a graph. This can be done for any graph, but we will restrict our attention to connected graphs. If G is a graph with edge set E, the *cycle matroid* of G, denoted $M(G)$, is the matroid whose element set, E, is the set of edges of the graph and whose set of bases, $\mathcal{B}$, is the set of spanning trees of G. We can list the bases of the cycle matroid of G by listing all of the spanning trees of the graph.

For the graph in the FIGURE 1, the edges $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$ are the elements of $M(G)$. We have already listed all of the spanning trees of the graph above, so we already have a list of the bases of this matroid.

We can also define a matroid on a finite set of vectors. The vectors are the elements, or *ground set*, of the matroid, and $\mathcal{B}$ is the set of maximal linearly independent sets of vectors. These maximal independent sets, of course, form bases for the vector space spanned by these vectors. And we recall that all bases of a vector space have the same size.

This helps us see where some of the terminology comes from. The bases of the vector matroid are bases of a vector space. What about the word *matroid*? We can view the vectors of our example as the column vectors of a matrix, which is why Whitney [**13**] called these matroids.

$$\begin{array}{c} \begin{array}{ccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{array} \\ \left[\begin{array}{ccccccc} 1 & 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{array}\right] \end{array}$$

These (column) vectors $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ are the elements of this matroid. The bases are the maximal independent sets listed in the previous section.

Now for a quick example not (necessarily) from a matrix or graph. We said that any pair $(E, \mathcal{B})$ that satisfies the two conditions is a matroid. Suppose we take, for example, a set of four elements and let the bases be every subset of two elements. This is a matroid (check the two conditions), called a *uniform matroid*, but is it related to

a graph or a collection of vectors? We will explore this later, but first let us further develop our first two examples.

Beyond the bases You might notice something now that we've looked at our two examples again. The bases of the cycle matroid and the bases of the vector matroid are the same, if we relabel v_i as e_i. Are they the same matroid? Yes. Once we know the elements of the matroid and the bases, the matroid is fully determined, so these matroids are *isomorphic*. An isomorphism is a structure-preserving correspondence. Thus, two matroids are isomorphic if there is a one-to-one correspondence between their elements that preserves the set of bases [**15**].

Knowing the elements and the bases tells us exactly what the matroid is, but can we delve deeper into the structure of this matroid? What else might we like to know about a matroid? Well, what else do we know about a collection of vectors? We know what it means for a set of vectors to be linearly *dependent*, for instance. In a graph, we often look at the *cycles* of the graph. If we had focused on the linearly dependent sets and cycles in our examples, we would have uncovered similar properties they share.

Recall also that, if we take a subset of a linearly independent set of vectors, that subset is linearly independent. (Why? If a vector could not be written as a linear combination of the others, it cannot be written as a linear combination of a smaller set.) Also, if we take a subset of the edges of a tree in a graph, that subset is still independent: If a set of edges contains no cycle, it would be impossible for a subset of those edges to contain a cycle. So any subset of an independent set is independent, and this is true for matroids in general as well.

We can translate some of these familiar traits from linear algebra and graph theory to define some more features of a matroid. Any set of elements of the matroid that is contained in a basis is an *independent set* of the matroid. Further, any independent set can be extended to a basis. On a related note, anytime we have two independent sets of different sizes, say $|I_1| < |I_2|$, then we can always find some element of the larger set to include with the smaller so that it is also independent: There exists some $e \in I_2$ such that $I_1 \cup e$ is independent. This is an important enough fact that if we were to axiomatize matroids according to independence instead of bases—as we mention later—this would be an axiom! It also fits our intuition well, if you think about what it means for vectors.

A subset of E that is not independent is called *dependent*. A minimal dependent set is a *circuit* in the matroid; by minimal we mean that any proper subset of this set is not dependent.

What is an independent set of the cycle matroid? A set of edges is independent in the matroid if it contains no cycle in the graph because a subset of a spanning tree cannot contain a cycle. Thus, a set of edges is dependent in the matroid if it contains a cycle in the graph. A circuit in this matroid is a cycle in the graph.

Get out your pencils! Looking back at the graph in FIGURE 1, we see that $\{e_2, e_4\}$ is an independent set, but not a basis because it is not maximal. The subset $\{e_7\}$ is not independent because it is a cycle; it is a dependent set, and, since it is a minimal dependent set, it is a circuit. (A single-element circuit is called a *loop* in a matroid.) In fact, any set containing $\{e_7\}$ is dependent because it contains a cycle in the graph, or circuit in the matroid. Another dependent set is $\{e_2, e_3, e_4, e_5\}$, but it is not a circuit; $\{e_2, e_3, e_5\}$ is a circuit.

In the vector matroid, a set of elements is independent in the matroid if that collection of vectors is linearly independent; for instance, $\{v_2, v_4\}$ is an independent set. A dependent set in the matroid is a set of linearly dependent vectors, for example $\{v_2, v_3, v_4, v_5\}$. And a circuit is a dependent set, all of whose proper subsets are independent. $\{v_2, v_3, v_5\}$ is a circuit, as is $\{v_7\}$. We noted earlier that any set containing $\{v_7\}$

is a linearly dependent set; we now see that any such set contains a circuit in the vector matroid.

One way to measure the size of a matroid is the cardinality of the ground set, E, but another characteristic of a matroid is the size of the basis, which we call the *rank* of the matroid. If $A \subset E$ is a set of elements of a matroid, the rank of A is the size of a maximal independent set contained in A. In our vector matroid example, let $A = \{v_1, v_2, v_6, v_7\}$. The rank of A is two. The rank of $\{v_7\}$ is zero.

Because it arose naturally from our examples, we defined a matroid in terms of the bases. There are equivalent definitions of a matroid in terms of the independent sets, circuits, and rank; indeed most introductions of matroids will include several such equivalent axiomatizations. Often the first set of exercises is to show the equivalence of these definitions. We spare the reader these theatrics, and refer the interested reader to Oxley [**9**] or Wilson [**14, 15**].

Matroids you may *not* have known

If a matroid can be represented by a collection of vectors in this very natural way, and can also be represented by a graph, why do we need this new notion of matroid? You may ask yourself, given some matroid, M, can we always find a graph such that M is isomorphic to the cycle matroid of that graph? Given some matroid, M, can we always find a matrix over some field such that M is isomorphic to the vector matroid? Happily, the answer to both of these questions is no. (Matroids might be a little boring if they arose only from matrices and graphs.) A graph or matrix does provide a compact way of viewing the matroid, rather than listing all the bases. But this type of representation is just not always possible. When a matroid is isomorphic to the cycle matroid of some graph we say it is *graphic*. A matroid that is isomorphic to the vector matroid of some matrix (over some field) is *representable* (or *matric*). And not every matroid is graphic, nor is every matroid representable.

To demonstrate this, it would be instructive to look at a matroid that is either not graphic or not representable. The smallest nonrepresentable matroid is the Vamos matroid with eight elements [**9**], and it requires a little more space and machinery than we currently have to show that it is not representable. However, it is fairly simple to construct a small example that is not graphic, so let us focus on finding a matroid that is not the cycle matroid of any graph.

Uniform matroids If we take a set E of n elements and let $\mathcal{B}$ be all subsets of E with exactly k elements, we can check that $\mathcal{B}$ forms the set of bases of a matroid on E. This is the *uniform matroid*, $U_{k,n}$, briefly mentioned earlier. In this matroid, any set with k elements is a maximal independent set, any set with fewer than k elements is independent, and any set with more than k elements is dependent. What are the circuits? Precisely the sets of size $k + 1$.

Let's consider an example. Let E be the set $\{a, b, c, d\}$ and let the bases be all sets with two elements. This is the uniform matroid $U_{2,4}$. Is this matroid graphic? To be graphic, $U_{2,4}$ must be isomorphic to the cycle matroid on some graph; so, there would be a graph G, with four edges, such that all of the independent sets of the cycle matroid $M(G)$ are the same as the independent sets of $U_{2,4}$. All of the dependent sets must be the same as well. Since every set with two elements is a basis of $U_{2,4}$, and every set with more than two elements is dependent, we see that each three-element set is a circuit. Is it possible to draw a graph with four edges such that each collection of three edges forms a cycle? Try it. Remember, each collection of two edges is independent, so must *not* contain a cycle.

A careful analysis of cases proves that it is not possible to construct such a graph, so $U_{2,4}$ is not isomorphic to the cycle matroid on any graph, and thus is not graphic. This matroid is, however, representable. A representation over $\mathbb{R}$ is given below. Check for yourself that the vector matroid is isomorphic to $U_{2,4}$ (by listing the bases).

$$\begin{matrix} a & b & c & d \end{matrix}$$
$$\begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 2 & 1 \end{bmatrix}$$

Notice that this representation is not unique over $\mathbb{R}$ since we could multiply the matrix by any nonzero constant without changing the independent sets. Also notice that this is *not* a representation for $U_{2,4}$ over the field with three elements $\mathbb{F}_3$ (the set $\{0, 1, 2\}$ with addition and multiplication modulo 3). Why? Because, over that field, set $\{c, d\}$ is dependent.

Harvesting a geometric example from a new field We just saw how a collection of vectors can be a representation for a particular matroid over one field but not over another. The ground set of the matroid (the vectors) is the same in each case, but the independent sets are different. Thus, the matroids are not the same. Let's further explore the role the field can play in determining the structure of a vector matroid, with an example over the field of two elements, $\mathbb{F}_2$. As above, the ground set of our matroid is the set of column vectors, and a subset is independent if the vectors form a linearly independent set when considered within the vector space $(\mathbb{F}_2)^3$.

$$\begin{matrix} a & b & c & d & e & f & g \end{matrix}$$
$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Consider the set $\{d, e, f\}$. Accustomed as we are to vectors in $\mathbb{R}^3$, our initial inclination is that this is a linearly independent set of vectors. But recall that $1 + 1 = 0$ over $\mathbb{F}_2$. This means that each vector in $\{d, e, f\}$ is the sum of the other two vectors. This is a linearly dependent set in this vector space, and thus a dependent set in the matroid, and not a basis. In fact, $\{d, e, f\}$ is a minimal dependent set, a circuit, in the matroid, since all of its subsets are independent.

The matroid generated by this matrix has a number of interesting characteristics, which you should take a few moments to explore:

1. Given any two distinct elements, there is a unique third element that completes a 3-element circuit. (That is, any two elements determine a 3-element circuit.)
2. Any two 3-element circuits will intersect in a single element.
3. There is a set of four elements no three of which form a circuit. (This might be a little harder to find, as there are $\binom{7}{4} = 35$ cases to check.)

Geometrically inclined readers might be feeling a tingle of recognition. The traits described above turn out to be precisely the axioms for a finite projective plane, once the language is adjusted accordingly.

A *finite projective plane* is an ordered pair, $(\mathcal{P}, \mathcal{L})$, of a finite set $\mathcal{P}$ *(points)* and a collection $\mathcal{L}$ *(lines)* of subsets of $\mathcal{P}$ satisfying the following [**5**]:

1. Two distinct points of $\mathcal{P}$ are on exactly one line.
2. Any two lines from $\mathcal{L}$ intersect in a unique point.
3. There are four points in $\mathcal{P}$, no three of which are collinear.

Elements of the matroid are the points of the geometry, and 3-element circuits of the matroid are lines of the geometry. Our example has seven points, and this particular projective plane is called the *Fano plane*, denoted F_7. The Fano plane is shown in FIGURE 3, with each point labeled by its associated vector over $\mathbb{F}_2$. Viewed as a matroid, any three points on a line (straight or curved) form a circuit.

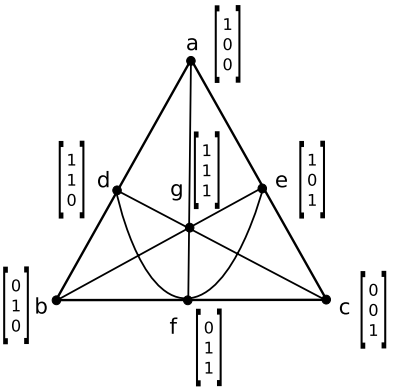

Figure 3 The Fano plane, F_7

We have already seen a variety of structures related to matroids, with still more to come. Ezra Brown wrote in *The many names of* (7, 3, 1) [**3**] in the pages of this MAGAZINE: "In the world of discrete mathematics, we encounter a bewildering variety of topics with no apparent connection between them. But appearances are deceptive." In fact, now that we've recognized the Fano plane as the Fano matroid, we may add this matroid to the list of the "many names of (7, 3, 1)". (For more names of F_7, the interested reader is referred, not surprisingly, to Brown [**3**].)

The Fano plane exemplifies the interesting fact that any projective geometry is also a matroid, though the specific definition of that matroid becomes more complicated once the dimension of the finite geometry grows beyond two. (Although the Fano plane has rank 3 as a matroid it has dimension 2 as a finite geometry, which is, incidentally, why it is called a plane. Oxley [**9**] gives further information.)

We started with a vector matroid and discovered the Fano plane, so we already know that the Fano matroid is representable. The question remains, is it graphic? We attempt to construct a graph, considering the circuits $C_1 = \{a, b, d\}$, $C_2 = \{a, c, e\}$, and $C_3 = \{b, c, f\}$. These would have to correspond to cycles in a graph representation of the Fano matroid. There are two possible configurations for cycles associated with C_1 and C_2, shown in FIGURE 4. In the first we cannot add edge g so that $\{a, f, g\}$ forms a cycle. In the second, we cannot even add f to form a cycle for C_3. (Since the

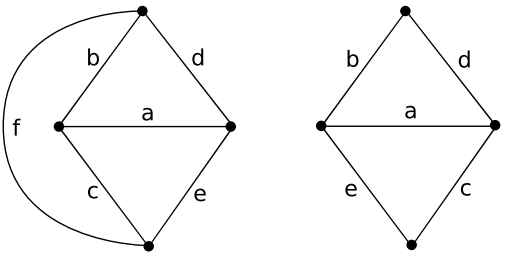

Figure 4 Two possible configurations

matroid has rank 3, the spanning tree must have three edges, so the graph would have 4 vertices and 7 edges.) Thus, the Fano matroid is not a graphic matroid.

One last fact about the Fano plane [**9**]: Viewed as a matroid, the Fano plane is *only* representable over $\mathbb{F}_2$.

Matroids—what are they good for?

Now that we have seen four different types of matroids, we consider their applications. Beyond unifying distinct areas of discrete mathematics, matroids are essential in combinatorial optimization. The greedy algorithm, a powerful optimization technique, can be recognized as a matroid optimization technique. In fact, the greedy algorithm guarantees an optimal solution only if the fundamental structure is a matroid. Once we've familiarized ourselves with the algorithm, we explore how to adapt it to a different style of problem. Finally, we will explore the ramifications, with respect to matroids, of the greedy algorithm's success in finding a solution to this different style of problem.

Walking, ever uphill, and arriving atop Everest Suppose each edge of a graph has been assigned a weight. How would you find a spanning tree of minimum total weight? You could start with an edge of minimal weight, then continue to add the next smallest weight edge available, unless that edge would introduce a cycle. Does this simple and intuitive idea work? Yes, but only because the operative structure is a matroid.

An algorithm that, at each stage, chooses the best option (cheapest, shortest, highest profit) is called *greedy*. The greedy algorithm allows us to construct a minimum-weight spanning tree. (This particular incarnation of the greedy algorithm is called Kruskal's algorithm.) Here are the steps:

In graph G with weight function w on the edges, initialize our set B: $B = \emptyset$.

1. Choose edge e_i of minimal weight. In case of ties, choose any of the tied edges.
2. If $B \cup \{e_i\}$ contains no cycle, then set $B := B \cup \{e_i\}$, else remove e_i from consideration and repeat previous step.

The greedy algorithm concludes, returning a minimum-weight spanning tree B.

We will later see that, perhaps surprisingly, this approach will always construct a minimum-weight spanning tree. The surprise is that a sequence of locally best choices results in a globally optimal solution. In other situations, opting for a locally best choice may, in fact, lead you astray. For example, the person who decides she will always walk in the steepest uphill direction need not end up atop Mount Everest, and, indeed, most of the time such a walk would end instead atop some hill (that is, a local maximum) near her starting point. Or, back to thinking about graphs, suppose a traveling salesperson has to visit several cities and return back home. We can think of the cities as the vertices of a graph, the edges as connecting each pair of cities, and the weight of an edge as the distance he must drive between those cities. What we seek here is a minimum-weight spanning cycle. It turns out that the greedy algorithm will not usually lead you to an optimal solution to the *Traveling Salesperson Problem*. Right now, the only way to guarantee an optimal solution is to check all possible routes. For only 10 cities this is $9! = 362{,}880$ possible routes. But for the minimum-weight spanning tree problem, the greedy algorithm guarantees success.

What does this have to do with matroids? The greedy algorithm constructs a minimum-weight spanning tree, and we know what role a spanning tree plays in a graph's associated cycle matroid. Thus, the greedy algorithm finds a minimum-weight basis of the cycle matroid. (Once weights have been assigned to the edges of G, they have also been assigned to the elements of $M(G)$.) Further, for *any* matroid, graphic or otherwise, the greedy algorithm finds a minimum-weight basis.

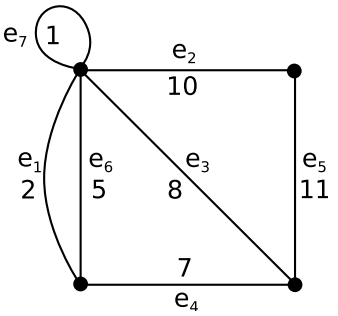

Figure 5 Graph G with weights assigned to each of its edges

Let's work through an example, based on the cycle matroid of the weighted graph shown in FIGURE 5. The greedy algorithm will identify a minimum-weight basis from the set of bases, $\mathcal{B}$. It will build up this basis, element by element; thus, in the algorithm below, the set B will not actually be a basis for the matroid until the algorithm has concluded. (It will be an independent set throughout, but only maximal when the algorithm concludes.) We will use matroid terminology to emphasize the matroidal nature of the algorithm:

Initialize our set B as $B = \emptyset$.

1. The minimum weight element is e_7, but it is rejected since its inclusion would create a circuit. (It is a loop.) $B = \emptyset$.
2. Consider next smallest weight element e_1. It creates no circuits with the edges in B, so set $B = \{e_1\}$.
3. Consider e_6: It creates a circuit with e_1, so do not add it to B. $B = \{e_1\}$.
4. Consider e_4: It creates no circuits with e_1, so set $B = \{e_1, e_4\}$.
5. Consider e_3: It creates a circuit with the current elements of B, so do not add it to B. $B = \{e_1, e_4\}$.
6. Consider e_2: It creates no circuits with the elements of B, so set $B = \{e_1, e_2, e_4\}$.
7. Consider the remaining element, e_5. It creates a circuit with the elements of B. $B = \{e_1, e_2, e_4\}$.

The greedy algorithm concludes, returning a minimum-weight basis $B = \{e_1, e_2, e_4\}$.

None of these steps was actually specific to the graph—they all involve avoiding circuits in the matroid. This is a matroid algorithm for constructing a minimum-weight basis, whether the matroid is graphic or not.

Let us sketch a proof of why this algorithm will always produce a minimum-weight basis. Suppose the greedy algorithm generates some basis $B = \{e_1, e_2, \ldots, e_n\}$, yet

there exists some other basis $B' = \{f_1, f_2, \ldots, f_n\}$ with smaller total weight. Further, without loss of generality, let the elements of each basis be arranged in order of ascending weight. Then $w(e_1) = w(f_1)$, necessarily. Let k be the smallest integer such that $w(f_k) < w(e_k)$. Consider the two independent sets $I_1 = \{e_1, \ldots, e_{k-1}\}$ and $I_2 = \{f_1, \ldots, f_k\}$, and recall the observation we made earlier about two independent sets of different size. Since $|I_1| < |I_2|$ we know there must be some $f_l, l \leq k$ such that $I_1 \cup f_l$ is independent. But this means f_l is both not dependent on $e_1, \ldots, e_{k-1}$ and has weight smaller than e_k. So this is a contradiction, because the greedy algorithm would have selected f_l over e_k in constructing B. This contradiction proves that the greedy algorithm will find a minimum-weight basis. (Oxley [**9**] gives more details.)

What is fascinating and quite stunning is that one may go further and *define* matroids using the greedy algorithm. That is, it turns out that any time the greedy algorithm, in any of its guises, guarantees an optimal solution for all weight functions, we may be sure that the operative mathematical structure *must* be a matroid. Stated another way, only when the structure is a matroid is the greedy algorithm guaranteed to return an optimal solution. (See Oxley [**9**] or Lawler [**7**].) We may, however, have to dig deep to find out what that particular matroid might be.

$$\left(\begin{array}{c}\text{The greedy algorithm}\\ \text{guarantees an optimal solution.}\end{array}\right) \Longleftrightarrow \left(\begin{array}{c}\text{The underlying structure}\\ \text{is actually a matroid.}\end{array}\right)$$

Figure 6 A stunning truth

Finally, one other observation on the nature of matroids is in order. Once a particular matroid is defined, another matroid on the same ground set naturally arises, the *dual matroid*. The set of bases of this new matroid are precisely the set of all complements of bases of the original matroid. That is, given a matroid M on ground set E, with set of bases $\mathcal{B}$, we may always construct the dual matroid with the same ground set and the set of bases $\{B' \subseteq E \mid B' = E \setminus B, B \in \mathcal{B}\}$. What is surprising is that this new collection of sets does in fact satisfy the basis axioms, and this fact has kept many matroid theorists employed for many years. In our current context, the reason this is particularly interesting is that any time the greedy algorithm is used to find a minimum-weight basis for a matroid, it has simultaneously found a maximum-weight basis for the dual matroid. Pause for a moment to grasp, and then savor, that fact. (In fact, the greedy algorithm is sometimes presented first as a method of finding a maximum-weight set of bases, in which case the adjective "greedy" makes a little more sense.)

Is a schedule(d) digression really a digression? Lest we forget how important mathematics can be in the so-called "real world," let us imagine a student with a constrained schedule. This student, call her Imogen, can only take classes at 1 PM, 2 PM, 3 PM, and 4 PM. She's found seven classes that she must take sooner or later, but at the moment she has prioritized them as follows in descending order of importance: Geometry (g), English (e), Chemistry (c), Art (a), Biology (b), Drama (d), French (f). The classes offered at i PM, denoted H_i, are

$$H_1 = \{c, e, f, g\}, \quad H_2 = \{a, b, d\}, \quad H_3 = \{c, e, g\}, \quad H_4 = \{d, f\}.$$

Now the question is perhaps an obvious one: Which classes should Imogen take to best satisfy the prioritization she has set up for herself? Granted, it can be tempting in a small example to stumble our way through some process of trial and error, but let's demonstrate ourselves a trifle more evolved. Casting ourselves in the role of Imogen's advisor, we will attempt something akin to the greedy approach we saw above. Though

it would be a rather busy schedule, we will allow Imogen to take four courses, if there is indeed a way to fill her hours.

Since Geometry is Imogen's top priority, any schedule leaving it out must be considered less than optimal, so we make sure she signs up for g. (This class is offered at two times, but for the moment we must suppress our desire to specify which hour we choose.) What should our next step be? If we can add English, e, without blocking out Geometry, then we should do so. Is it possible to be signed up for both g and e? Yes, each is offered at 1 PM and 3 PM. (We still need not commit her to a time for either class.) Can she take her third priority, Chemistry, c, without dislodging either of those two classes? No, because Chemistry is only offered at 1 PM and 3 PM. There are only two possible times for her top three priorities. What about her fourth priority, Art, a? Yes, she could take Art at 2 PM, the only time it is offered. Imogen's next priority is Biology, b, but it is only offered at 2 PM, where it conflicts with Art. Finally we may fill one more slot in her schedule by signing her up for Drama, d, at 4 PM.

Now her schedule is full, and she is signed up for her first, second, fourth, and sixth most important classes, and filled all her time slots. Better yet, she even still has some flexibility, and can choose whether she'd like to take Geometry at 1 PM and English at 3 PM or vice versa. As her advisor, we leave our office feeling satisfied with our performance, as we should, for if we were to search all her possible schedules, we would find that this is the best we could do.

Why do we need powerful concepts like matroids and the greedy algorithm to tackle this problem? In this example, the problem and constraints are simple enough that trial-and-error may have allowed us to find the solution. But in more complicated situations the number of possibilities grows massive. (This is affectionately referred to as the "combinatorial explosion.") If Imogen had eight possible times to take a class and a prioritized list of 17 classes, trial-and-error would likely be a fool's errand. Similarly, in our earlier example with 25 computers in a network, constructing a minimum-weight spanning tree without the algorithm would be miserable: we would need a quadrillion years to compare all possible spanning trees. Imagine an actual company with hundreds of computers! Knowing that our structure is a matroid tells us that the algorithm will work, and the algorithm is an efficient way to tackle a problem where an exhaustive search might take the fastest computer longer than a human lifetime to compute.

The hidden matroid For Imogen's schedule, at each stage we chose the best option available that would not conflict with previous choices we had made. This is another incarnation of the greedy algorithm, and in this type of scheduling problem it will always produce an optimal solution. (We omit the proof here for brevity's sake. See Bogart [**2**] or Lawler [**7**] for details.) Since the greedy algorithm is inextricably connected to matroids, it must also be true that a matroid lurks in this scheduling example. Let's ferret out that matroid!

To identify the matroid, we need to identify the two sets in $(E, \mathcal{B})$. The first is fairly simple: E is the set of seven courses. Now which subsets of E are bases?

The solution to the scheduling problem is actually an example of a *system of distinct representatives* (or SDR) [**7**]. We have four possible class periods available, and a certain number of courses offered during each period. A desirable feature of a course schedule for Imogen would be that she actually *takes* a class during each hour when she is available. We have a set of seven courses, $\mathcal{C} = \{a, b, c, d, e, f, g\}$, and a family of four subsets of $\mathcal{C}$ representing the time slots available, which we've denoted H_1, H_2, H_3, and H_4. We seek a set of four courses from $\mathcal{C}$ so that each course (element of the set) is taken at some distinct time (H_i). The classes we helped Imogen choose, $\{a, d, e, g\}$, form just such a set; a distinct course can represent each time. Formally, a

set S is a *system of distinct representatives* for a family of sets $A_1, \ldots, A_n$ if there exists a one-to-one correspondence $f : S \to \{A_1, \ldots, A_n\}$ such that for all $s \in S$, $s \in f(s)$. These problems are often modeled using bipartite graphs and matchings. Bogart [**2**] and Oxley [**9**] give details.

The greedy algorithm returns a minimum-weight basis; in our example that basis was an SDR. It turns out that any system of distinct representatives for the family of sets H_1, H_2, H_3, H_4 is a basis for the matroid; that is, $\mathcal{B} = \{S \subseteq \mathcal{C} \mid S \text{ is an SDR for family } H_1, H_2, H_3, H_4\}$. The SDR we found was minimum-weight. (Imogen defined a weight function when she prioritized the classes.)

We now know the matroid, but which sets are independent in this matroid? Gone are the familiar characterizations like "Are the vectors linearly independent?" or "Do the edges form any cycles?" Thinking back to the definition of independence, a set is independent if it is a subset of a basis. In our example, a set $S \subseteq \mathcal{C}$ will be independent when it can be extended into a system of distinct representatives for H_1, H_2, H_3, H_4. This is a more unwieldy definition for independence. But there is a simpler way to characterize it. As long as a subset of size k (from $\mathcal{C}$) can represent k members of the family of sets (the H_is), it will be possible for that set to be extended to a full SDR (assuming that a full SDR is indeed possible, as it was in our example). Naturally, this preserves the property that any subset of an independent set is independent; if some set S can represent $|S|$ members of the family of sets, then clearly any $S' \subseteq S$ can also represent $|S'|$ members of the family of sets.

So it turns out that $(E, \mathcal{B})$ forms a matroid. By definition, SDRs must have the same number of elements, and thus no SDR can properly contain another, satisfying the first condition for the bases of a matroid.

A full proof of the basis exchange property for bases would be rather involved, so let's examine one example to see how it works in this matroid. Consider two bases, $B_1 = \{b, c, d, f\}$ and $B_2 = \{a, d, e, g\}$. Again, each of these is an SDR for the family of sets H_1, H_2, H_3, H_4. In a full proof, we would show that for any element x of B_1, there exists some element y of B_2 such that $(B_1 \setminus x) \cup y$ is a basis, which in this case is an SDR. For this example, consider element d of B_1. We must find an element of B_2 to replace d, and if we do so with e we find that, indeed, the resulting set $B_3 = \{b, c, e, f\}$ is an SDR, as shown in TABLE 1.

TABLE 1: Three SDRs and the sets they represent

Time	Set	B_1	B_2	$B_3 = (B_1 \setminus d) \cup e$
1 PM	H_1	French	Engl. or Geom.	Chemistry
2 PM	H_2	Biology	Art	Biology
3 PM	H_3	Chemistry	Engl. or Geom.	English
4 PM	H_4	Drama	Drama	French

Notice that in B_2, there are options for which class will represent H_1 and H_3. In defining the SDR, we need not pick a certain one-to-one correspondence, we just need to know that at least one such correspondence exists. Note also that the sets represented by f and c changed from B_1 to B_3. Finding a replacement for d from B_2 forced the other classes to shuffle around. This is a more subtle matroid than we've yet seen. (You may also have noticed that we could have just replaced d from B_2 when building B_3. But, wouldn't that have been boring?)

What if Imogen had chosen a list of classes and times such that it was only possible for her to take at most two or three classes? Even in situations where no full system

of distinct representatives is possible, there exists a matroid such that the bases are the partial SDRs of maximum size. Any matroid that can be realized in such a way is called a *transversal matroid*, since SDRs are usually called transversals by matroid theorists. Such a matroid need not be graphic (but certainly could be). The relationship between the types of matroids we have discussed is summarized in the Venn diagram in FIGURE 7.

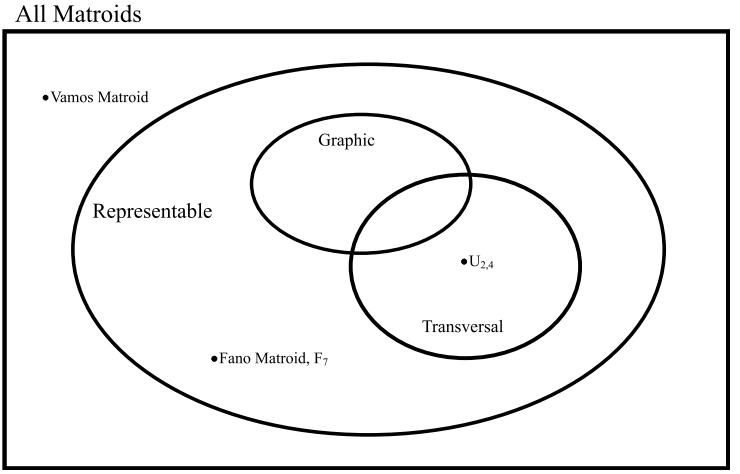

Figure 7 Matroids you have seen

Matroids you have now seen

Where you previously saw independence you might now see matroids. We have encountered five matroids: the cycle matroid, the vector matroid, the uniform matroid, the Fano matroid, and the transversal matroid. Some of these matroids you have known, some are new. With the matroid, we travel to the worlds of linear algebra, graph theory, finite geometry, and combinatorial optimization. The matroid is also tied to endless other discrete structures that we have not yet seen. We have learned that the greedy algorithm is a characterization of a matroid: when we have a matroid, a greedy algorithm will find an optimal solution, but, even more surprisingly, when a greedy approach finds an optimal solution (for all weight functions), we must have a matroid lurking. Once, we have even found that lurking matroid.

Do you now see matroids everywhere you look?

REFERENCES

1. Garrett Birkhoff, Abstract linear dependence and lattices, *Amer. J. Math.* **57** (1935) 800–804.
2. Kenneth P. Bogart, *Introductory Combinatorics*, 3rd ed., Harcourt Academic Press, San Diego, 2000.
3. Ezra Brown, *The Many Names of* (7, 3, 1), this MAGAZINE **75** (2002) 83–94.
4. Tom Brylawski, A partially-anecdotal history of matroids, talk given at "Matroids in Montana" workshop, November, 2006.
5. Ralph P. Grimaldi, *Discrete and Combinatorial Mathematics: An Applied Introduction*, 5th ed., Pearson/Addison-Wesley, Boston, 2004.
6. Frank Harary, *Graph Theory*, Addison-Wesley, Reading, MA, 1972.
7. Eugene Lawler, *Combinatorial Optimization: Networks and Matroids*, Dover Publications, Mineola, NY, 2001.

8. Saunders MacLane, Some interpretations of abstract linear dependence in terms of projective geometry, *Amer. J. Math.* **58** (1936) 236–240.
9. James G. Oxley, *Matroid Theory*, Oxford University Press, Oxford, 1992.
10. Gian-Carlo Rota and Fabrizio Palombi, *Indiscrete Thoughts*, Birkhäuser, Boston, 1997.
11. Douglas B. West, *Introduction to Graph Theory*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2000.
12. B. L. van der Waerden, *Moderne Algebra*, 2nd ed., Springer, Berlin, 1937.
13. Hassler Whitney, On the abstract properties of linear dependence, *Amer. J. Math.* **57** (1935) 509–533.
14. Robin J. Wilson, An introduction to matroid theory, *Amer. Math. Monthly* **80** (1973) 500–525.
15. Robin J. Wilson, *Graph Theory*, 4th ed., Addison Wesley Longman, Harlow, Essex, UK, 1996.

Letter to the Editor: Archimedes, Taylor, and Richardson

The enjoyable article "What if Archimedes Had Met Taylor?" (this MAGAZINE, October 2008, pp. 285–290) can be understood in terms of eliminating error terms. This leads to a different concluding approximation that is more in the spirit of the note by combining previous estimates for improvement. We denote the paper's weighted-average estimates for π based on an n-gon by A_n, using area, and P_n, using perimeter. The last section shows two formulas,

$$\text{Error(perim)} = P_n - \pi = \frac{\pi^5}{20n^4} + \frac{\pi^7}{56n^6} + \cdots \quad \text{and}$$

$$\text{Error(area)} = A_n - \pi = \frac{2\pi^5}{15n^4} + \frac{2\pi^7}{63n^6} + \cdots,$$

where we have corrected the first term in the latter. A combination of $8/5$ of the first and $-3/5$ of the second will leave $O(1/n^6)$ error. So, the last table could show

$$\frac{8}{5}P_{96} - \frac{3}{5}A_{96} = 3.14159265363.$$

This approach could be used alternatively to justify

$$A_n = \frac{1}{3}AI_n + \frac{2}{3}AC_n,$$

where AI_n and AC_n are inscribed and circumscribed areas respectively, by subtracting out the $1/n^2$ error terms and leaving the corrected error formula above. For general integration, a similar derivation motivates Simpson's rule as the combination of $1/3$ trapezoidal rule plus $2/3$ midpoint rule. This is more than a coincidence since the inscribed area connects arc endpoints as in trapezoidal rule and circumscribed area uses the arc midpoint.

The technique of combining estimates to eliminate error terms is known as Richardson's Extrapolation in most numerical analysis textbooks. It is usually applied to halving step-size in the same approximation formula. For example, Archimedes could have computed

$$\frac{16}{15}P_{96} - \frac{1}{15}P_{48} = 3.14159265337,$$

if Taylor could have whispered these magical combinations.

—Richard D. Neidinger
Davidson College
Davidson, NC 28035