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.