Conserving the Seed Corn: Reflections on the Academic Hiring Crisis

Eric Roberts
Department of Computer Science
Stanford University
eroberts@cs.stanford.edu

Abstract

Computer science departments today face a serious staffing crisis, as faculty and graduate students abandon academia for industry while undergraduate enrollments rise. The current crisis is similar to one that occurred in the early 1980s, which gives us the opportunity to learn from that experience. This article reviews the history of the earlier crisis and proposes strategies for mitigating the effects of the current one.

1. Introduction

As those seeking to hire new faculty are already aware, there is a crisis facing academic computer science. Undergraduate enrollments are rising rapidly, while the number of Ph.D. students seeking academic positions is falling. As a result, colleges and universities often cannot hire enough faculty to meet student demand. Writing in last December's issue of the *SIGCSE Bulletin* [11], Paul Myers and Henry Walker conclude that "there is a very serious shortage of new Ph.D.s in computer science," to the point that in 1997-98 "only about half of the open tenure-track positions were filled." The crisis has also attracted increasing coverage in the popular and academic press, including an excellent report in the September 24, 1999 issue of the *Chronicle of Higher Education* entitled "Computer scientists flee academe for industry's greener pastures." [15]

The purpose of this essay is to look at the problems facing academic computer science from both a historical and a strategic perspective. Although the current hiring crisis represents a substantial change from the situation earlier in the 1990s, faculty shortages in computer science are not a new phenomenon. A similar crisis occurred in the early 1980s. By looking at the history of that period, we can learn important lessons about how such problems develop—lessons that will help us respond more effectively to the challenges we face today.

2. The "seed corn" phenomenon

In many ways, the crisis in academic computer science is a direct result of the tremendous success of the computing industry in recent years. Computing technology has acted as a primary engine for the national economy—the driving force behind the longest period of sustained economic growth in modern history. Students today see the enormous excitement in the computing industry and want to be part of it. As a result, undergraduate interest in computer science has skyrocketed. At the same time, the excitement of the industry has enticed many of the best computer science students to take jobs immediately after completing their bachelor's degrees, foregoing any thought of graduate education or academic careers. In the end, fewer people are available to teach the next generation of computer scientists, even as demand for that type of education soars.

This threat to the sustainability of computer science education has become a matter of considerable concern. In a 1999 report entitled "The supply of information technology workers in the United States," the Computing Research Association observes that

> Many educators, industrial laboratory leaders, and government science officials are concerned that the high industrial demand for information technology (IT) workers will siphon out of the educational systems many students who would otherwise pursue an advanced degree. This diminishes [the] pool of people who will join the university faculties that perform basic research and teach the next generation of students. This problem is compounded when industry also successfully recruits current faculty members, including junior faculty who would become the academic leaders of the profession in the coming decades. This is known as the "seed-corn" problem—an analogy to those who consume too much of this year's crop, reserving too little for next year's planting. [8]

The metaphor of "eating our seed corn" was first articulated by Peter Denning, who wrote an article with that title in 1981 [5]. Many of the problems computer science faces today are precisely the ones it faced in the early 1980s. A 1981 report issued with the endorsement of computer science department chairs at their biennial Snowbird conference described conditions that sound strikingly familiar today:

> Students are enrolling in record numbers in our undergraduate programs. While many are attracted to

This paper appears in the December 1999 issue of *Inroads,* the quarterly bulletin of the ACM's Special Interest Group on Computer Science Education (SIGCSE).

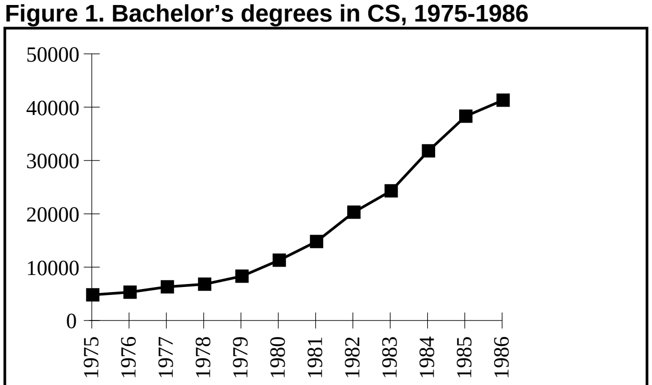
> Computer Science by its excellent careers, a growing number is being encouraged by the “computer revolution” to take introductory and intermediate programming courses in order to become familiar with the computer as a tool in other disciplines. The result? Existing terminal facilities and computing centers cannot handle the load. Class sizes balloon. Lab facilities are insufficient. Faculty consider industrial positions. [6]

The situation today, however, is different from that of the 1980s in one crucial respect: we have lived through this situation before. When universities and colleges first faced these problems, they had to respond without the benefit of hindsight. Today, we have the advantage of being able to look back on that earlier experience, which allows us to learn from both the mistakes that were made and the strategies that succeeded.

3. The academic crisis of the early 1980s

For most of the 1970s, undergraduate interest in computer science grew steadily but at a relatively modest pace. Toward the end of the decade, however, there was a fantastic explosion in the level of undergraduate interest in computer science, as the graph in Figure 1 shows.

Figure 1. Bachelor’s degrees in CS, 1975-1986



Sources: National Science Foundation, National Center for Education Statistics

From 1975 to 1981, the number of BS degrees in computer science increased by a factor of three. Over that same period, the number of computer science Ph.D.s remained relatively constant, hovering at an average level of 230 doctorates per year. Over the next five years, undergraduate degree production again showed a nearly threefold increase. Although Ph.D. production also rose during this period, reaching a level of 412 in 1986, the rate of growth was substantially lower and did not keep pace with the demand for new faculty, particularly since Ph.D. computer scientists increasingly chose industrial positions over academic ones.

The imbalance between the modest increase in Ph.D. production and the rapid growth in the number of undergraduates led to a severe faculty shortage at the end of the 1970s. Within academia, the problem was quickly recognized as departments sought unsuccessfully to hire the faculty they needed to teach a burgeoning undergraduate population. The problem was sufficiently severe that ACM president Peter Denning identified computer science as “a discipline in crisis” [6].

In part because of the importance of computing to the national economy, the U.S. federal government was also quick to respond. In a report entitled *Science and Engineering Education for the 1980’s and Beyond*, the National Science Foundation and Department of Education issued the following warning:

> Computer professional fields represent the single broad area in which there are clear shortages of personnel at all degree levels at the present time. [12]

The report also addressed the effect of such shortages on academic programs in computer science, along with other engineering disciplines in which academia and industry compete for a limited pool of talent.

> Declining Ph.D. production and the availability of more attractive employment opportunities in industry for persons with doctorates in engineering and the computer professions have led to a shortage of both junior and senior faculty in these fields. . . . There are approximately 200 vacancies in departments that specialize in the computer professions. These vacancies are straining the capacities of these schools and departments.

In 1983, Kent Curtis of the National Science Foundation issued a much more detailed report [4], which focused on the problems facing computer science as distinct from other engineering disciplines. He cited the following as evidence that a crisis did in fact exist:

- Students are not entering graduate school but are being lured by attractive salaries and professional opportunities at the bachelor’s level.
- Graduate students are leaving graduate school without completing their Ph.D.s.
- Faculty are leaving academia for industry.

Taking this evidence together with an analysis of the economic forces affecting the labor supply, Curtis wrote

> We must conclude that the educational institutions of the country cannot obtain the labor they need and have poor prospects of finding it in the near future. They face a real crisis. The migration of student interest is working against them, not in their favor, and the job mobility which allows so many people to enter computer professions in business, industry and government is not effective for educational institutions because of their highly specialized job requirements.

4. Responses to the crisis

In the face of rapidly growing demand for computer science education in the midst of a severe faculty shortage, how did academic institutions respond? The most common strategies undertaken at that time are discussed in the sections that follow, along with an analysis of the advantages and disadvantages that have become clear through hindsight.

Increasing workloads

In what was probably the most common and least successful response, many colleges and universities tried to meet the demand with their existing faculty resources by increasing the workload. The risks of such a strategy are apparent in

Kent Curtis's report, in which he discusses the results of an NSF survey that sought to assess institutional reactions to the growing undergraduate demand:

> There is a consistent pattern in their responses and the results can be applied without exception to the computer field whether the departments are located in engineering schools or elsewhere. 80% of the universities are responding by increasing teaching loads, 50% by decreasing course offerings and concentrating their available faculty on larger but fewer courses, and 66% are using more graduate-student teaching assistants or part-time faculty. 35% report reduced research opportunities for faculty as a result. In brief, they are using a combination of rational management measures to adjust as well as they can to the severe manpower constraints under which they must operate.

As rational as this approach seems, increasing faculty workload is generally shortsighted. Curtis observes that

> These measures make the universities' environments less attractive for employment and are exactly counter-productive to their need to maintain and expand their labor supply. They are also counterproductive to producing more new faculty since the image graduate students get of academic careers is one of harassment, frustration, and too few rewards.

As Curtis makes clear, increasing workloads in an attempt to meet a growing demand is extremely risky. The long-term health of academia is enhanced by attracting more students into academic positions, thereby increasing the number of faculty available to offer the necessary instruction. The short-term response of most institutions in the early 1980s—increasing faculty loads to meet the level of demand—undermines this long-term goal.

That teaching loads cannot be increased arbitrarily without risking faculty attrition is further demonstrated by the following message posted to the **`SIGCSE-MEMBERS`** list by a former faculty member who had left academia:

> I would like to offer my reasons for leaving, because I think my reasons are common reasons. In my five-person CS department, three of us left in the last two years. . . . For all of us, salary was certainly one big issue. But there was another one—working conditions and workload. In my case, I knew the time to leave had come when I realized, not only were my industry friends making far more money than I, but they were working fewer hours with less stress. This was what drove my colleagues to leave as well. [10]

To avoid faculty burnout, academic institutions must adopt other measures in their attempts to satisfy increasing student demand.

Faculty retraining

In the early 1980s, one of the most common strategies for developing a computer science program was to recruit faculty from other disciplines. At the time, many colleges and universities did not have a department of computer science or any significant academic program in the field. If—as was generally the case given the shortage of faculty applicants—those institutions were unable to hire computer science Ph.D.s, the easiest way to staff popular introductory computer science courses was to convince faculty members from other fields, such as physics or mathematics, to take on those courses. Most of those faculty members had programming experience that enabled them to offer excellent courses at the introductory level. Students, however, soon began to demand more advanced courses, for which the faculty were not adequately prepared. As a result, many of the faculty members who volunteered to teach the introductory courses sought advanced retraining, usually in the form of a master's degree in computer science. A few universities began to offer programs specializing in faculty retraining. According to informal reports, the retraining course at the University of Massachusetts at Amherst was swamped with applicants, including several from professors who were already *chairing* departments of computer science.

I am convinced that academic computer science could not have survived were it not for the willingness of some faculty to move to a new field. For the most part, those who migrated to computer science were extremely conscientious about acquiring the expertise they needed to teach in their adopted discipline. Their efforts sustained computer science education at many institutions and helped reduce the impact of the earlier crisis.

At the same time, the strategy of retraining faculty has its dangers. For one thing, faculty who have taken one or two courses as part of a retraining program will not have the depth and breadth of understanding they would get from pursuing a Ph.D. in computer science. Although many aspects of academic tradition are certainly open to question, thorough knowledge of one's own discipline seems critical to faculty effectiveness and represents a central foundation of higher education as a whole.

The more serious problems that accompany a strategy of faculty retraining, however, appear only over time. In the early 1990s, after computer science had emerged from the depths of its hiring crisis, I served as an evaluator for several computer science programs. In those institutions, the department was typically composed of junior faculty with computer science Ph.D.s working to satisfy tenure requirements imposed by a senior faculty trained in other disciplines. This situation created considerable tension among the junior faculty, which reduced their level of satisfaction with the academic experience and may have contributed to the flight of junior faculty from academia to industry.

Adjunct faculty

Another common strategy, particularly for institutions located near industrial centers, was to recruit adjunct faculty from industry to teach individual courses. After all, given that people with the necessary expertise were moving into the industry, it was reasonable to look to industry when departments found faculty in short supply.

Adjunct faculty often teach excellent courses that are extremely popular with students. They have first-hand experience integrating the theoretical underpinnings of computer science with the demands of professional practice.

Moreover, because so much cutting-edge research in computer science occurs in industry, adjunct faculty can bring new ideas into academic institutions that help maintain their intellectual vitality.

Despite these advantages, relying too heavily on adjunct faculty—assuming they are available—carries some risks. First, most such faculty are not integrated into the life of the institution. Most adjunct faculty come to campus for lecture, with perhaps a few office hours on the side. Such activities, while important, do not encompass much of what is important in education. One of the central advantages of the academic environment is that it constitutes a community in which students and teachers can collaborate in the pursuit of knowledge. Adjunct faculty are only tangentially part of that community and are unable to participate in much of the learning process that goes on outside the classroom.

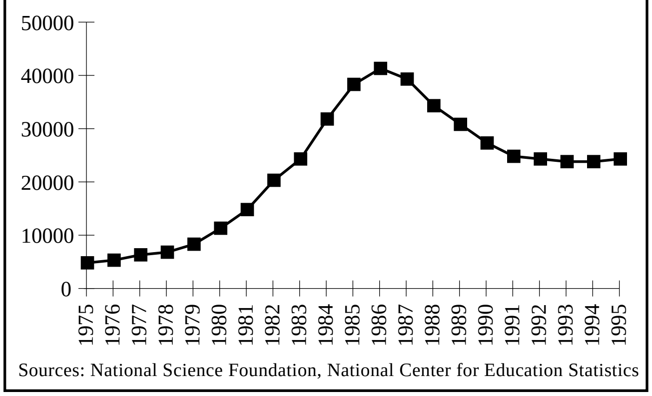
The second—and perhaps more serious—concern about the use of adjunct faculty lies in the potential for a conflict of interest. While I do not believe most faculty members visiting from industry intend to recruit students for their own companies, it is certainly true that students who take such courses make connections that pull them toward industry and away from academic careers.

Limiting enrollments

The two previous strategies—retraining faculty from other disciplines and hiring industrial experts as adjuncts—both address the supply side of the imbalance between the availability of teaching expertise and the expanding demand from students for education. At some point in the 1980s, these strategies proved insufficient, forcing departments to restrict demand by imposing limits on enrollment. Some institutions attained these limits by setting strict quotas on the number of students who could major in computer science or by requiring extraordinarily high GPAs to declare computer science as a major. Others achieved the same effect without formal limitations, simply by making the introductory courses so difficult that relatively few students would continue in the field.

These measures certainly had their effect. During the late 1980s, enrollments in computer science shrank precipitously, as shown in Figure 2. The number of bachelor's degrees peaked in 1986 and then declined steadily through 1994. Although most colleges and universities report significant growth in undergraduate enrollment since 1995, these increases have not yet appeared in the national statistics, which are not yet available for years beyond 1995.

Figure 2. Bachelor's degrees in CS, 1975-1995



Sources: National Science Foundation, National Center for Education Statistics

It is easy to draw incorrect conclusions from the data in Figure 2. The graph seems to indicate that students lost interest in computer science during the second half of the 1980s. In seeking to explain this phenomenon, some educational policy analysts have argued that undergraduate enrollment dropped as a result of a downturn in the industry. Computing, after all, is cyclical. When the industry is hot, students flock to computer science; when the industry cools off, students abandon the field.

The problem with this explanation is that the falloff in student enrollment *precedes* the economic downtown of the 1980s. The computing industry began to have trouble toward the end of the decade, after the decline in the number of students had already begun. Moreover, graduation rates are a lagging indicator with respect to the choices students make for their undergraduate majors. If the number of bachelor's degrees peaked in 1986, then students started to choose other majors a few years earlier, sometime around 1984. As those who saw the famous Superbowl commercial [14] will remember, 1984 was the year that Apple introduced the Macintosh—hardly a down year for the industry.

Having watched what was happening at the time, I feel quite certain that the decline in students shown in Figure 2 has little if anything to do with student choice. Students in the mid 1980s did not *decide* not to major in computer science but were instead *prohibited* from doing so by departments that lacked the resources to accommodate them. Given the pressures departments faced at the time, these restrictions may well have been necessary. Moreover, they did, in the end, mitigate the crisis. They did so, however, at an enormous cost. At a time when industry needed more people to sustain its momentum, universities were forced to cut back. The flow of students collapsed, and industry was faced with a shrinking labor pool. Given the complexity of any economic system, it is usually impossible to prove causality, but I have believed for some time that the crisis in academic computer science during the 1980s contributed significantly to the industrial decline at the end of the decade.

Quite apart from the cost to the economy as a whole, restricting enrollment creates other problems for computer science departments. Not surprisingly, enrollment limitations are unpopular with students. Implementing restrictions creates considerable tension that leads in turn to an erosion in quality of life for faculty, as the relationship with students becomes more adversarial. When I was forced in 1983 to restrict the number of students who could major in computer science, the ambience of the community changed dramatically. Students became more competitive and, in many cases, angry. The last straw for me was getting a call in my office from a lawyer retained by a parent, who insisted that the tuition his clients paid to my college obligated us to take all students into the major. The fact that we were *unable* to hire additional faculty had no effect on his perception of the case.

The most insidious effect of enrollment limitations, however, is likely to be the fact that the students most likely to continue in the face of such limitations may not be the ones faculty enjoy seeing in their classrooms. At first, it might seem as if enrollment limitations based on such nominally objective criteria as GPAs would select for the brightest and most highly motivated students. This is not necessarily the case. More often—as is certainly the stereotype in premedical programs—tight admissions standards select for the most highly competitive students. Faced with limitations and hurdles that make studying computer science less fun, those who are motivated by a love of the subject matter can easily lose their interest. Those who instead see computer science—and indeed their education—merely as a means to an end are much more likely to stick it out. Through this process of self-selection, students who might be interested in graduate careers often give way to those who are bound and determined to become the next multibillionaires in the industry.

Enrollment limitation will almost certainly have a disastrous effect on the diversity of the undergraduate computer science population. Students from weaker school systems and those who have not had the opportunity to work with computers at home will have much more trouble with introductory courses designed to act as filters for a limited-admission major. Similarly, studies have documented the fact that women are likely to underrate their own abilities with respect to their male counterparts [16]. Faced with a highly competitive admissions process, women are more likely to choose other options in selecting a major. From 1986 to 1991, the number of men graduating with bachelor's degrees in computer science dropped by 34 percent, while the number of women declined by 51 percent [2].

5. Lessons for the current crisis

In order to survive the current crisis, academic computer science must draw on the lessons from history. On the one hand, we cannot seek to satisfy the growing demand from undergraduates with a shrinking pool of faculty. Doing so increases the pressure on faculty who remain in academia and ultimately accelerates the exodus to industry. On the other hand, it is equally important to avoid the opposite extreme of shutting off student demand by imposing draconian limits on enrollment. While forcing a reduction in the supply of trained computer professionals may help academia cope with its crisis, it does so at the expense of society that needs their expertise. As described in the preceding section, enrollment limitations also threaten the cohesion of the academic community and may end up selecting for a more industry-oriented, less culturally diverse student population.

In order to strike the correct balance between these competing demands, we need to find more creative ways to increase the number of people available for teaching by drawing on more resources. From my experience in several very different academic environments, I suggest that institutions take the steps outlined in the following sections.

Adopt an aggressive plan for faculty recruitment

There is no reason to abandon the search for faculty, just because they are scarce. In many cases, qualified candidates may be looking for opportunities in your geographic area or have personal ties to your institution. Canvas your alumni, pursue any leads, and be persistent. And, if you are lucky enough to find faculty, make sure that you treat them well enough to have them stay. Academia will not be able to match industrial salaries. Even so, some qualified faculty candidates will prefer academic life to the corporate world. To attract those candidates, your salaries must be competitive with those at comparable institutions.

Create academic positions that focus on teaching

When faced with the crisis of the 1980s, most colleges and universities were more easily persuaded to retrain faculty from other disciplines than they were to hire qualified computer scientists who lacked a Ph.D. The imprimatur of the degree was regarded as more important than knowledge of the field. In the current crisis, it is essential to reverse that priority and look for people with an interest in and aptitude for teaching, whether or not those individuals have the training or the credentials for research. In disciplines where one can find scholars who are skilled in both teaching and research, you should certainly look for that combination. If you hold out for that combination in computer science, you will likely be left in the cold. Not every institution today needs to do research in computer science, but it is hard to imagine that a college or university that failed to offer basic instruction in this area—so vital to society and to effective research in most other fields—could claim that its students were receiving a modern education. There are people out there who have the necessary skills and love teaching, but who would prefer to teach without the encumbrance of competing expectations for research. At Stanford, we have a team of six lecturers—half with Ph.D.s and half with master's degrees—who do a fantastic job teaching the large undergraduate courses. Many other universities do the same, and it is certainly time that more colleges found the courage to follow that lead.

Get undergraduates involved in teaching

The crisis in computer science education arises from the fact that there are too few teachers to meet the need. At the same time, there are, in some sense, too many undergraduates. One of the best ways to meet student demand is to get those undergraduates involved in the teaching process. Using undergraduates as teaching assistants not only helps to alleviate the teaching shortfall but also provides a valuable educational experience to the student assistants. Many schools—including Brown, Georgia Tech, Harvard, Stanford, the University of Arizona, and the University of Virginia—have major programs to recruit and train undergraduates as teaching assistants who play a central role in the educational process. Our program at Stanford is described in detail in a paper in the 1995 SIGCSE symposium [13], and I strongly encourage you to adopt similar programs in your own institution.

6. The need for public policy initiatives

Although the strategies described in the preceding section can reduce the impact of the current shortfall in computer science faculty, the crisis facing computer science education is a difficult one that will require a coordinated effort to address. In examining the situation in 1983, Kent Curtis concluded that the educational system faced "a staffing problem which seems to have no solution within the context of normal supply/demand forces" [4]. I believe that this conclusion is equally valid today. Academia lacks the resources to address the faculty shortfall on its own and will need help from other sources to address the problem.

Unfortunately, industry is not really in a position to provide the right sort of assistance. Industry is subject to market forces that optimize for short-term individual gain over long-term public good. That industry would seek to hire all available talent in computer science, thereby depleting the future supply, is a classic example of a problem that goes by many names. In 1968, Garrett Hardin identified the problem as *the tragedy of the commons* [9], which he introduces by asking his readers to picture a sheep-grazing pasture open to all. What happens when that pasture, or commons, reaches its capacity? Each individual shepherd has an incentive to introduce more sheep into the commons, even though it cannot support the increase. After all, the profit from the extra sheep remains with the individual shepherd, while the cost of overgrazing the pasture is, in the long run, shared by all. In game theory, this problem is often known as a *prisoner's dilemma,* characterized by a situation in which the optimal choice for each individual is not the best choice for the community as a whole. In economics, such situations represent instances of *market failure*—situations in which we cannot expect normal forces of supply and demand to reach an optimal overall condition.

When faced with a tragedy of the commons, the only solution is to impose constraints that restrict the individual's ability to take actions that harm the community. Academia and industry cannot, by themselves, bring such constraints into existence. Ensuring that the public good is not sacrificed to individual gain—whether the costs be polluting the environment, exhausting natural resources, or hiring every available person with computing expertise—is the responsibility of government, which must take action to ensure that no individual or company can monopolize scarce resources to the detriment of society as a whole.

7. Conclusions

In this paper, I have used the lessons of the faculty-hiring crisis of the early 1980s to suggest three survival strategies for colleges and universities facing similar problems today:

1. Adopt an aggressive plan for faculty recruitment
2. Create academic positions that focus on teaching
3. Get undergraduates involved in teaching

These techniques will make it easier to meet student demand for computer science courses, but do not, in themselves, constitute a long-term solution to the problem. To maintain our ability to produce well-trained computing professionals, academia, industry, and government must work together to develop public policies that support faculty development and retention. By doing so, we will be able to sustain the computer revolution that has been such a major force in our economy and our lives.

References

1. Ethan Bronner. Voracious computers are siphoning talent from academia. *The New York Times,* June 25, 1998, page A1.
2. Tracy Camp. The incredible shrinking pipeline. *Communications of the ACM,* October 1997.
3. Computing Research Association. Taulbee surveys of CS departments, **`http://www.cra.org/statistics`**.
4. Kent K. Curtis. Computer manpower: Is there a crisis? Washington, DC: National Science Foundation, 1983, **`http://www.acm.org/sigcse/papers/curtis83/`**.
5. Peter J. Denning. Eating our seed corn. *Communications of the ACM,* June 1981.
6. Peter J. Denning, Edward Feigenbaum, Paul Gilmore, Anthony Hearn, Robert Ritchie, Joseph Traub. A discipline in crisis. *Communications of the ACM,* June 1981.
7. Peter J. Denning. Our seed corn is growing in the commons. *Information Impacts Magazine,* March 1999, **`http://www.cisp.org/imp/march_99/denning/03_99denning.htm`**.
8. Peter Freeman and William Aspray. The supply of information technology workers in the United States. Washington, DC: Computing Research Association, July 1999, **`http://www.cra.org/reports/wits/it_worker_shortage_book.pdf`**.
9. Garrett Hardin. The tragedy of the commons. *Science Magazine,* December 13, 1968.
10. Bonnie McKellar. Reasons for leaving academia. Message posted to the **`SIGCSE-MEMBERS`** discussion list, July 28, 1998.
11. J. Paul Myers, Jr. and Henry M. Walker. The state of academic hiring in computer science: An interim review. *SIGCSE Bulletin,* December 1998.
12. National Science Foundation and the Department of Education. *Science and Engineering Education for the 1980's and Beyond*. Washington, DC: U.S. Government Printing Office, October 1980.
13. Eric Roberts, John Lilly, and Bryan Rollins. Using undergraduates as teaching assistants in introductory programming courses: An update on the Stanford experience. *SIGCSE Bulletin,* March 1995.
14. Ridley Scott. *1984*. Commercial for the Apple Macintosh broadcast on January 24, 1984, **`http://www.apple-history.com/movies/1984.mov`**.
15. Robin Wilson. Computer scientists flee academe for industry's greener pastures. *Chronicle of Higher Education,* September 24, 1999.
16. Laraine Zappert and Kendyll Stansbury. In the pipeline: a comparative analysis of men and women in graduate programs in science, engineering, and medicine at Stanford University. Stanford University: Institute for Research on Women and Gender, 1985.