Angela Smiley
CS 6460: Educational Technology: Personal Question

Note: I've opted to break my personal question into three parts to make this writeup more digestible. The original was:

> *What makes academic advising difficult and what are the various ways that [these difficulties] have been addressed in the past?*
> *What makes the OMSCS problem different (either easier or harder) and how can you utilize available technologies to achieve the goals of the community regarding academic advising services?*

Academic advising difficulties

(Note: in this section I consider mainly "dedicated" academic advisors – that is, those who are not members of the faculty but for whom advising is the primary responsibility. Many of these issues also apply to advisors of other types, with the added complication that because a non-dedicated advisor has another role they are likely even more pressed for time.)

Several difficulties arise in traditional academic advising:

- *Subjectivity.* Advising is an art, not a science. I read many, many variations on this pithy quote while researching the first half of this course.
- *Lack of guidance.* Related to the above, advisors do not receive a detailed set of instructions for how to do their jobs. They need to figure out for themselves what methods enable them to gather information, establish a rapport with their advisees, and stay abreast of new developments.
- *Student diversity.* Students have very different backgrounds *and* goals; one may be a first-generation college student pursuing her Bachelor's degree in computer science while another may be a legacy student obtaining a dual-major in biology and nursing. What is easy to one of these students may not be to the other, and what is obvious to one may need to be taught to the other.
- *Data, data, data.* The advisor must master a diverse and *large* set of data. In addition to the graduation requirements for every degree program and prerequisites for and reputations of every course, they must find out about their own students. Some data (such as transcript and GPA) may be provided by the institution, but they themselves must figure out how to pick up other information on the fly.
- *Variability.* Because each advisor has his or her own methods of data-gathering, their advisees in turn may have wildly different sets of data available to them. A seasoned or well-connected advisor, who knows which courses are really beasts and how to prepare, may confer a big advantage on advisees. An inexperienced or indifferent advisor, one who isn't connected to the "rumor mill" of student opinions, can hurt them badly.
- *Scale.* This isn't just an OMSCS problem! Advisors at traditional institutions are increasingly expected to carry caseloads of hundreds of students per

term. That means that an advisor must not only figure out the relevant data to collect but also find ways of collecting and organizing it that allow them to retain and make use of it even when they are not on a first-name basis with many of their advisees.

Unique challenges and opportunities in OMSCS

The principal challenge of adapting advising to OMSCS is that the "human element" doesn't scale. Unfortunately, this is also one of the principal challenges currently facing the culture of on-campus advising. Setting aside the recent shift towards a higher caseload per advisor, many of the other issues I cited (most notably variability between advisors) stem from the inability to apply the skills and insights of the best advisors beyond a relatively small number of students, or to avoid requiring each advisor to duplicate the work of learning a lot of information.

This has an interesting corollary: a solution that works well for OMSCS might eventually be adapted to other programs at other institutions!

Other factors which make advising in the OMSCS either easier or harder:

- (Easier) The computer-mediated nature of the program makes data much easier to collect.
- (Easier) The popularity of tools like the UCRSS and the overwhelming OMSCS responses on CIOS (as compared with the on-campus sections) point to a spirit of voluntarism that *also* makes data easier to collect.
- (Easier) Once we determine which data are relevant to advising our students, we can apply the same insight to all students with no additional cost. Once data (especially course opinions) are collected and evaluated in a formal, computerized form, they can be made equally available to all.
- (Harder) Because our particular implementation will not be officially aligned with the institution, participation must be 100% optional. This means that many of the proactive interventions present in a traditional advising program – identifying and requiring additional advising for "at-risk" students, for instance – are not feasible for us.
- (Harder) Related to the above, data on grades, attrition, etc., is self-reported and may not be accurate. This is a challenge if we know, for example, that students who got an 'A' in Machine Learning do much better in Reinforcement Learning; the power of this insight is weakened if a student feels pressure to exaggerate their grade because, for example, they do not realize this information will not be public. We will not have access to definitive statistics about students withdrawing from the program, so exploring retention will also be challenging.

Utilizing available technologies to achieve the goals of the community

In any advising program, the goals of the individual student must be held paramount (assuming in all cases ethical behavior). For our purposes, we can divide an OMSCS student's advising goals into *education* and *graduation*.

- Some *educational* criteria: What do I expect the objective utility of this course to be? How valuable is it to my effectiveness as a computer scientist and my happiness as a human being? (For example, does my background adequately prepare me to learn from it? Does the subject matter bring me personal delight? Will the course's content help me in my current or future work?)
- Some *graduation* criteria: Does my overall combination and schedule of courses satisfy the graduation requirements for my degree program and any personal constraints on my schedule? (For example, in light of the other courses I will take, does taking this course allow me to complete my specialization? Can I complete this course even with a vacation scheduled for finals week?)

Different students have different heuristics for weighing these criteria relative to one another. For instance, one student might be inclined to take an "extra" course if it is particularly applicable to her work or research area, while another might be strictly tied to completing the program at the minimum possible cost (financial or temporal). However, both areas are relevant to every student in the program.[1]

Graduation requirements are generally objective, either met or unmet.[2] It is possible to formulate them in a way that can be manipulated directly by the computer, and it is generally simple to solicit a user's feedback on them. Many other scheduling problems (e.g., the job shop scheduling problem and related tasks, such as constrained critical path project scheduling) share enough common features with this subproblem that the techniques used to address them can also be applied here.

In particular, helping a student meet their graduation requirements can be modeled as a constraint-satisfaction problem with prioritization (also known as a weighted constraint-satisfaction problem). In English, this means that we can do an excellent job of capturing the graduation goal by:

- Intelligently selecting constraint classes to model (e.g., "desired specialization" or "desired graduation date")
- Deciding the relative importance of each
- Allowing the student to tell us what their values for those constraints are (e.g., "Interactive Intelligence" and "2016")
- Coming up with individual methods for mapping courses into each of these constraints (e.g., metadata listing the required courses for a given specialization or a procedure for estimating how many hours a week of study is required for a given class)
- Presenting a list of courses that satisfies as many of the most-important constraints as possible.

[1] Ignoring extreme edge cases, such as students who enroll with no intention of graduating.

[2] Graduate or graduate not, there is no try.

The good news is that CSPs are a research area of great vitality and there are many well-developed solving heuristics available for us to implement. The bad news is that most general formulations of CSPs are (at least) NP-complete. The consolatory news is that an optimal solution is probably not required – after all, a human advisor would not offer an optimal solution, either. This issue will need to be explored through user testing – how good does a plan need to be for it to be of benefit to the student?

In contrast to graduation requirements, *educational* requirements are open-ended and usually difficult to communicate or evaluate. Questions like "how much will I enjoy this course?" and "how much will it challenge me?" don't have consistent answers; they depend on attributes of the student that can never be fully captured by our system. Furthermore, the system itself has no way of validating whether its plan contains courses that are "too hard" or "too boring."

Clearly, this subproblem requires a very different analogy. We can find it in the domain of online shopping systems, where a user must evaluate products and make decisions like "Will this saw meet my needs?" and "How much will I enjoy this book?" In 2015, almost any e-commerce system (and quite a number of free distribution platforms as well) includes a recommendation system to suggest other appropriate purchases. The techniques employed by these recommender systems – which revolve around collecting user reviews and matching up similar users – can be borrowed to form the basis of our approach to supporting the student's educational goals.

A word on other goals

All of the foregoing assumes that the community's goals are simply the aggregate of the goals of the individual students within it. However, it is instructive to consider alternatives, too.

There are other stakeholders in the OMSCS community beyond students, including prospective applicants, instructors, administrators, advisors, and researchers. Let's speculate a bit about what each of these types might want:

- Applicants might like to know what students say about their courses once they are enrolled in the program, either to decide whether the program is right for them or to help tailor their application and statement of purpose to the areas they deem most relevant to OMSCS admissions.
- Instructors might like to know what factors lead to a well-loved course versus an unpopular one, or an educationally effective course (one which improves a student's grades in subsequent courses) versus an unimportant one. Insights in this vein could potentially be extracted through data mining; while the small (in terms of different courses offered) sample size makes this challenging, results that are not statistically significant enough to be

conclusive might still offer good starting points for the instructor's own investigations.

- Administrators might like to know what factors influence student retention and student satisfaction. If we can design a good proxy metric for leaving the program, we can perform statistical analysis to see what factors often precede a student's departure. Correlation is not causation, but again, these would be promising starting points for an investigation.
- Advisors might like to stop getting X,000 e-mails in their inbox every day during registration. In addition to the obvious, they might appreciate a tool that allows them to predict enrollment in different courses, so they can work with Ms. Wilson to project the required number of TAs and with the registrar's office to set the course enrollment caps appropriately. Assuming that student plans are saved, this could be accomplished simply by skimming that data (anonymized, in aggregate).

Researchers are a bit of a special case; the system's ability to support research will be largely a function of non-technological factors. (For example, do we have an established system in place for asking students to 'opt-in' to data collection for research purposes? How well and how systematically can we process data requests?) For that reason I won't devote time to discussing them here.

Finally, can we make the argument that the community *itself* has goals, above and beyond those of its members? This might be a bit of a stretch, but the continued evolution of the OMSCS Google+ presence (and the creation of alternative communication channels, such as Slack) suggests that one of the community's current goals is *connectedness*. The very decentralized nature of the program means that students do not meet and socialize without a conscious effort to do so; nonetheless, we seem to be growing into a more and more interconnected community.

How could an OMSCS advisor program serve *this* goal? It's possible that adding certain social networking features (e.g., sharing one's own plan and following other students to find out what classes they add to theirs) could facilitate the development of community, allowing students to conveniently form study groups and encouraging them to connect with classmates both before the class begins and after it ends.