Snapshot 2: Matching and the New York Sanitation Department

Margaret H. Wright
Computer Science Department
Courant Institute of Mathematical Sciences
New York University

Simons Society of Fellows Retreat
February 24, 2018
How it began (October 24, 2017)

From: Russel E Caflisch <caflisch@courant.nyu.edu>

I received a request for help from Joel Binn with an optimization problem.

NYC has $10^4$ employees, who can make requests for job changes within the city government. The priority of these choices is supposed to be ordered by seniority. But they have the issue that a job change by a lower priority employee could open up a position that is desired by a higher priority employee. So this requires iteration, which they now perform by hand. They are looking for a researcher who could work with them and write an algorithm that would solve this problem.
I was (very) interested—Bell Labs nostalgia—and began talking to the NYSD people.

Initially I was misled by the term “optimization” in the NYSD’s description of the problem. I pictured formulating it as an integer programming problem with nonlinear constraints.

But it soon became clear that this is a matching problem, involving discrete objects (in this case, locations to which workers can request to transfer).

Some well known examples of matching problems are: locations for medical residencies; organization of donors/recipients of kidney transplants; and allocation of housing by universities.

I knew almost nothing about matching! (But I was happy to learn.)
A key feature of the NYSD problem is that it does not involve “big data”. There are approximately 1000 different locations in the NYSD system, and two or three times each year several hundred workers are allowed to request transfers.

Possibly the most important point about the NYSD problem is that seniority is an absolute priority. A second crucial feature is that someone who wants to keep his/her location cannot be forced to move.

Using a so-called “serial dictatorship”, meaning going down the list in priority order, giving each person the location he/she wants if it is available, does not solve the problem. (It would, if all positions were considered to be vacant at the beginning of each round of transfer requests.)
Here’s an example of why this is unsatisfactory.

Suppose that there are 3 agents, $a_1$, $a_2$, $a_3$, with seniority in that order. Initially, $a_1$ is in position $h_{10}$, $a_2$ is in position $h_8$, and $a_3$ is in position $h_7$.

Suppose that $a_1$’s top choice is $h_8$, with second preference $h_9$; $a_2$’s top preference is $h_{10}$; and $a_3$’s top choice is $h_8$.

At the beginning, $a_1$ cannot have $h_8$ because it is occupied by $a_2$, so $a_1$ is moved to $h_9$ (his/her second choice) leaving $h_{10}$ available. Now it is the turn of $a_2$, who is able to have his/her top choice of $h_{10}$, thereby making $h_8$ available. So when we reach $a_3$, he/she can move to $h_8$.

This is unacceptable because $a_3$, who has lower seniority than $a_1$, is getting $h_8$, which was the first choice of $a_1$.

This leads to a union grievance!
There are many papers, mostly in the economics literature, about housing problems. In 1999, Abdulkadiroğlu and Sönmez published a paper about “housing allocation with existing tenants”, which, it turns out, is the same problem as NYSD’s.

A&S proposed an algorithm with an incredibly evocative name: 

*You Request My House; I Get Your Turn*

called YRMH-IGYT.
This algorithm does exactly what its name says: the list of transfer requests is traversed in order of seniority, assigning locations as long as there is no conflict. But when $a_k$, who has priority $k$, requests a house that is occupied by $a_m$, where $m > k$ (so that $a_k$ has higher priority than $a_m$), $a_m$ “gets the turn of” $a_k$, and the process continues from there.

In the example above, the solution is simple: $a_1$ requests the house of $a_2$, so $a_2$ is inserted before $a_1$. Since $a_2$'s top choice is actually the house of $a_1$, a trivial cycle has been created: $a_1$ gets $h_8$ and $a_2$ gets $h_{10}$.

It can be shown that, by completing such cycles and then removing them from the transfer process, the method must terminate. In addition, it has many splendid properties.
A very satisfying outcome for all parties!