Re: Problem on an nxn grid

Source: http://sci.tech−archive.net/Archive/sci.math/2007−05/msg03287.html

---

- *From*: Chip Eastham <hardmath@xxxxxxxxx>

- *Date*: 20 May 2007 07:18:10 −0700

On May 17, 1:05 pm, Jonathan Berry <jbe...@xxxxxxxxxxxxx> wrote:

> It has taken me days to understand the equivalent of "See Spot Run" in the vocabulary of Graph Theory. Sparing everyone the litany of that journey...
>
> Thanks very much to every poster, especially Chip, Asinops and Robert Israel, for their help.
>
> The problem is a "maximum−weight matching" with n up to around 300. Does there exist code (or better, pseudo−code) to solve this? It could be argued that Chapter 5.3 of Gibbons's "Algorithmic Graph Theory" gives a pseudo−code for the Edmonds algorithm, but it is couched in, er, Graphic, language. As, for the purposes of this discussion, a lukewarm mathematician, and a lukewarm coder, it looks daunting. Indeed, I have mined google, and although there seems to be something approaching pseudo− code for related problems (such as the Stable Roommate Problem) and code for parts of the Edmonds algorithm (such as the Hungarian algorithm), I could not find code or pseudo−code for the whole kahuna, an algorithm which solves the maximum−weight matching problem in polynomial time. Maybe I haven't looked in the right places. But an email from one of you confirms: no readily− available code.
>
> I'm guessing that the reason will be that it is "too easy". Too easy a subject for a degree thesis, that is.
>
> Anyway, if somebody can point me to pseudo−code, I'd appreciate it.
>
> In a separate post, I will put forward an idea that might not reduce the big−O Order of working this through by brute force, but might reduce the real−world CPU time in cases where the values (weights) are not overly close to each other.
>
> −−

Jonathan Berry

Hi, Jonathan:

My last post to this thread seems to have gone missing, an all too frequent event with Google Groups. Below I will refer to the weight of edge from node i to node j as A(i,j), consistent with the earlier grid/matrix notation in this thread. All weights are assumed to be nonnegative. We seek a minimum (total) weight perfect matching of the (underlying) complete graph on n nodes.

This paper:

http://www.dcg.ethz.ch/publications/ctw04.pdf

by Wattenhofer and Wattenhofer gives pseudo−code for two approximation algorithms assuming that the weights of the graph satisfy the triangle inequality, i.e. A(i,j) <= A(i,k) + A(k,j) for all nodes i,j,k. This restriction is in one sense inessential as adding max(A) to every weight forces that condition to be true but at the expense of messing up the approximation property (because it now applies to the total inflated by max(A) * n/2).

It suggests the question of whether exact solutions for weighted graphs satisfying triangle inequalities can be found just as efficiently, e.g. with nearly quadratic algorithms rather than the improvement of Edmonds original O(n^4) to O(n^3) by Gabow (and in an earlier paper by Lawler).

Although most implementation details are omitted, a helpful discussion of the history and tests is here:

http://www2.isye.gatech.edu/~wcook/papers/match_ijoc.pdf

in a paper by Cook and Rohe. If the weights can be realized as the Euclidean distances between nodes assigned coordinates, then the problem is said to be "geometric", and of course the weights must then satisfy the triangle inequality condition.

It is known that an exact solution can be obtained in near quadratic time if the coordinates are in a plane:

www.cs.brown.edu/cgc/cgc98/final/final21.ps by divide and conquer techniques.

regards, chip

.