Facebook friend wheels and quadratic assignment problems

Cody Bredendick Michael C. Ferris[*]

December 23, 2009

Abstract

A friend wheel is an image used on Facebook to provide a visual representation of the relationships between the friends any one person may have. We outline a tool, based on the solution of quadratic assignment problems, that generates such a wheel. The tool uses the Facebook API to gather data, the GRASP heuristic to solve the model, and GraphViz to generate the visualization.

1 Introduction

As is the case at many universities, we teach a senior level undergraduate course in optimization tools and modeling at the University of Wisconsin. The course covers a wide range of optimization topics, but concentrates on the types of models that can be easily formulated within a modeling system such as GAMS or AMPL, and attempts to provide the students with a set of techniques that can be readily adapted to a range of different application settings. As is typical for such courses, the content involves a large number of examples and modeling exercises, coupled with an overview of the perceived and observed strengths and weaknesses of different modeling formats. In order to stress the difficulties associated with data collection and processing, the course requires an individual final project that is defined by the student, utilizing the skills they have been developing during the semester and experimenting with the issues of data acquisition, refinement and usage. To ensure the project is well defined and appropriate, a single page proposal is required from the students earlier in the semester that

[*]Computer Sciences Department, University of Wisconsin, Madison, Wisconsin. The work of this author was partially supported by funding from DOE, NSF, and AFOSR.

forms the basis of an oral interview to modify the scope or content of the proposal. The final projects include a (stricly enforced) 4 page summary, but students are allowed to provide additional electronic material (including models and data) that may or may not be exercised by the instructor during evaluation. We believe this process works extremely well, and the students (and instructor) learn lots about modeling and optimization in practice. It can also lead to significant research opportunities as such projects can lead to new avenues of exploration initiated by the industry and bright ideas of interested and motivated students. Specifically, Ferris' work on radiation therapy was initiated by a project in an offering of this class carried out by David Shepard [8].

This short note describes the results of one of these projects, carried out by Cody Bredendick while he was enrolled in this class in 2009, to visualize "friend data" within Facebook. The results of other projects from this one class were also impressive, and included an investigation by Joonhoon Kim of some ideas for biological pathway models, utilizing a bilevel program to implement "Opt knock" [2]:

$$
\begin{array}{ll}
\text{max} & \text{bioengineering objective (through gene knockouts)} \\
\text{s.t.} & \text{max cellular objective (over fluxes)} \\
 & \begin{array}{ll} \text{s.t.} & \text{fixed substrate uptake} \\ & \text{network stoichiometry} \\ & \text{blocked reactions (from outer problem)} \end{array} \\
 & \text{number of knockouts} \leq \text{limit}
\end{array}
$$

The goal of the project was to extend a model that accounts for "gene" deletions and regulatory interactions, and solve the resulting problem for different biochemical productions to identify diverse metabolic engineering strategies. The specific model included about 30,000 constraints and 20,000 variables (10,000 binary).

A second project by Brandon Smith concerned video stabilization, which was motivated by the observation that videos shot with a hand-held conventional video camera often appear remarkably shaky. Such camera shake is one of the biggest components of the large quality difference between home movies and professional videos, and Smith observed that while several commercial software packages were available for digitally removing or reducing camera shake as a post processing step, the project would provide an opaque, open source version of such software. Underlying this project was a nonlinear least squares problem:

$$
\min_{\{H_f\}_{f=1}^F} \sum_{f=2}^{F-1} \sum_{j \in \phi_f} \left\| u_{f,j} - \frac{1}{2}(u_{f-1,j_{prev}} + u_{f+1,j_{next}}) \right\|^2 + \Phi(\{H_f\}_{f=1}^F)
$$

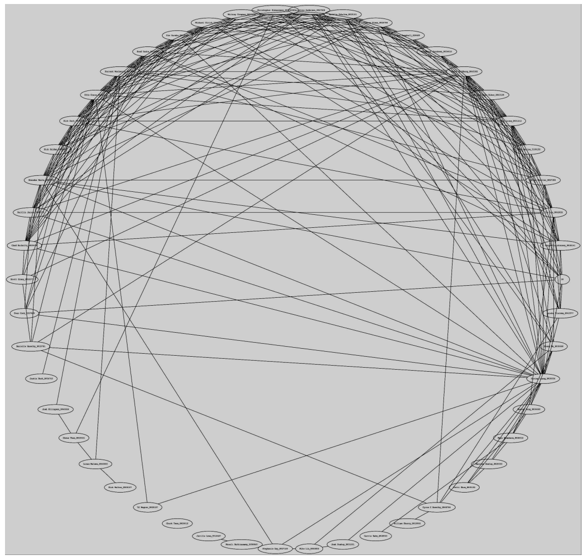

Figure 1: A friend wheel

where

$$\Phi(\{H_f\}_{f=1}^{F}) = \sum_{f=2}^{F-1} \lambda_1\theta_f^2 + \lambda_2 t_{x,f}^2 + +\lambda_3 t_{y,f}^2 + \lambda_4 s_{x,f}^2 XS + \lambda_5 s_{y,f}^2$$

H is an unknown transform, parameterized by θ, t and s. Smith was able to solve this problem and demonstrated the results using actual video samples.

2 The Friend Wheel Project

By now, Facebook needs no real introduction. It is one of the largest and most visited websites in existence and has practically defined the modern definition of "social networking". Within this domain, many new features are constantly being developed. Facebook released a developers' API that allows apps to be developed by anyone. A friend wheel is one such app. A specific example is given as Figure 1. The typical size of such wheels involves from 50 to 1000 friends.

A friend wheel is an image used to provide a visual representation of the relationships between the friends any given person may have. It is constructed by gathering each of the target individual's friends and placing

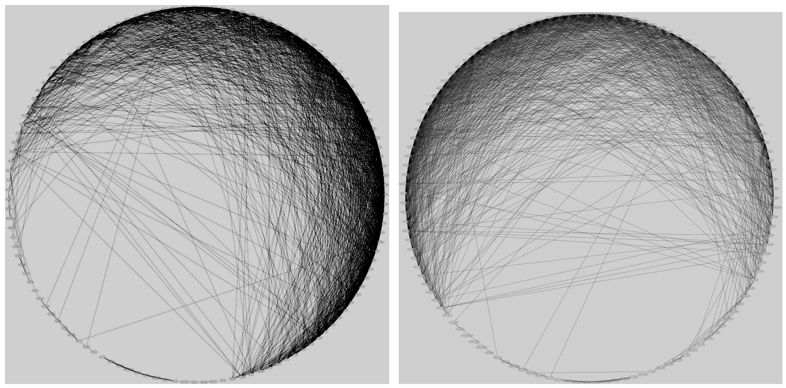

Figure 2: Two representations of a friend wheel using different orderings

them at uniform spacing on the circumference of a circle. Having done that, line segments are drawn between each point (each friend) if the people that those two connecting points represent are (announced) friends with each other. It is clear that ordering determines the resulting image, and that with a random ordering, one can expect a very dense mesh of lines uniformly distributed across the interior of the circle. Such friend wheels existed before this project was conceived: the inspiration of the particular project was to order the friends around the circumference of the circle in order to reduce amount of ink used in displaying the wheel.

In order to complete this project, three main steps were needed. Firstly, the required data needed to be gathered. This was accomplished by building a small *web service* to retrieve friend data using the Facebook API. The data is then manipulated into an appropriate format for the optimization solver using php scripts within Facebook.

The second step is to determine the ordering of friends around the circumference of the circle. We describe that below in Section 2.1.

The final step is to manipulate the optimization output for an image package to allow visualization of the image. The approach used here involved the (publically available) *graphviz* package, with two passes, one to fix the locations on the circle and draw the connecting arcs, and the second to generate the image in a nice format (for example png, jpg, pdf). Specific details on this are available from the authors on request.

2.1 Formulation and Solution via Quadratic Assignment

In order to minimize the ink that is needed to create a facebook wheel, we model the problem as a quadratic assignment problem (QAP). The QAP was first proposed in 1957 by Koopmans and Beckmann [3] as a mathematical model for facility location. Specifically, given n facilities $\{f_1, \ldots, f_n\}$, and n locations $\{l_1, \ldots, l_n\}$, the problem is to determine to which location each facility must be assigned.

Formally, if we let $W = (w_{i,j}) \in \mathbb{R}_+^{n\times n}$ where $w_{i,j}$ represents the flow between f_i and f_j, let $D = (d_{i,j}) \in \mathbb{R}_+^{n\times n}$ where $d_{i,j}$ represents the distance between l_i and l_j, and let $p : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$ be an assignment of facility f_i to location $l_{p(i)}$ whose cost is

$$c(p) = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j} d_{p(i),p(j)}$$

then the problem of interest is:

$$\text{QAP} : \min c(p) \text{ subject to } p \in \Pi_n.$$

Here we use the notation Π_n to denote the set of all permutations of $\{1, \ldots, n\}$. It is known that QAP is strongly NP-hard [7]. A survey is found in [6], and more recent results are detailed in [4] and the references contained therein.

To formulate our minimizing ink problem as a QAP we let n represent the number of friends of a given individual, assign each friend a location on the circumference of the circle and define $w_{i,j} = 1$ if i is a friend of j, and 0 otherwise. The distance from location r on the circle circumference to location s (Euclidean or otherwise) is denoted by $d_{r,s}$. Thus, the only required input to define this problem is n and the links $w_{i,j}$ that represent that i is a friend of j. In this way, the data acquisition is very compact.

Given the instruction from the class, the first attempt was to model the problem in GAMS using binary variables $p_{i,r} = 1$ to indicate that facility f_i is located at l_r. The problem is a mixed integer quadratic program

$$\begin{array}{ll} \min & \sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j} \sum_{r} \sum_{s} d_{r,s} * p_{i,r} * p_{j,s} \\ \text{subject to} & \sum_{i} p_{i,r} = 1, \forall r \\ & \sum_{r} p_{i,r} = 1, \forall i \end{array}$$

Unfortunately, as is well known, even state-of-the-art commercial codes are unable to solve these formulations: the relaxed problem is not necessarily convex, and even tricks such as fixing one location to remove symmetry are ineffective. Similar results are found if additionally binary variables $q_{i,r,j,s} = (p_{i,r}$ and $p_{j,s})$ are defined using

$$\begin{aligned} q_{i,r,j,s} &\leq p_{i,r} \\ q_{i,r,j,s} &\leq p_{j,s} \\ q_{i,r,j,s} &\geq p_{i,r} + p_{j,s} - 1 \end{aligned}$$

to make the problem a mixed integer linear program. This is expected since it is well known that exact solution is very hard, even for some small ($n = 40$) problems [1]. More recent work, some using semidefinite programming is also referenced in [4]. While there are sophisticated approaches that may outperform the simple models we implemented, it is clear that the time frame and size of our models precluded exact solution.

Consequently, there are many heuristic solution stategies suggested for these problems. The survey [4] gives a more complete picture. For this project, we utilized a greedy randomized adaptive search procedure (GRASP) as given in [5]. A critical reason for this choice was the availability of the source code for download. While more details are available in the cited reference, the main idea is to generate starting solutions using a greedy heuristic, apply local search until some termination criteria satisfied, and optionally apply "path-relinking" procedures (a heuristic to combine to good solutions) to improve the overall quality of solutions generated.

Some very limited results are provided below to show the use of this heuristic compared to some exact methods.

		25		50		125	
solver	options	time	soln	time	soln	time	soln
dicopt		1.3	195	27.0	1752		
sbb		4.6	189	98.9	1742		
grasp		0.2	186	1.3	1660	16.7	25107
	-npr					7.0	25143
	-npr -n 2					14.0	25143
	-npr -i 64					14.0	25143
	-npr -i 10					2.2	25143
	-npr -i 5					1.1	25398
	-npr -s 250					7.1	25144

It is clear that the grasp code is much more effective than the exact codes shown here, and that specific options to that code facilitate quick solution of reasonable sized models. However, for models of larger size (including the example given in Figure 3, it is clear that there is need for application of other heuristic methods and these kinds of instances undoubtedly provide a rich class of test models for such approaches.

2.2 Details of the procedure

To try these approaches, the following steps are needed:

- Login to your facebook account
- Navigate to *http://apps.facebook.com/cscbfriend/*
- This page calls index.php on a remote "execution" websever
- Initiates a non-blocking call to start the data gathering process
- php preparedata.php [userID] [sessionID]
  - Communicates with facebook API
  - Generate a list of friends, and then a list of connections (using the Facebook Query Language)
  - Loops over data in order to generate names.txt, arcs.txt, qinput.txt
  - Executes a shell script to run "grasp" and "graphviz"

Unfortunately, a timer program was needed to allow the page more than the default time to load. Details on this again are available from the authors.

The approach is one way to generate such friend wheels. As is typical with "apps" there are a number of other competitive versions of these wheels. One such application generated the output in Figure 3. While the output shown was slow to generate, and the solution represented there is not as good as the QAP solution that methods such as the grasp code provide, there is significant other functionality within the displayed output that make it more appealing to facebook users. Specifically, the output is colored and has dynamic facilities within it to allow a user to manipulate the wheel and investigate features of the representation and the social groupings that are clustered along the circumference. Future work aims to collaborate with the developer of this application to switch out the solver engine to use a more sophisticated optimization tool under the hood.

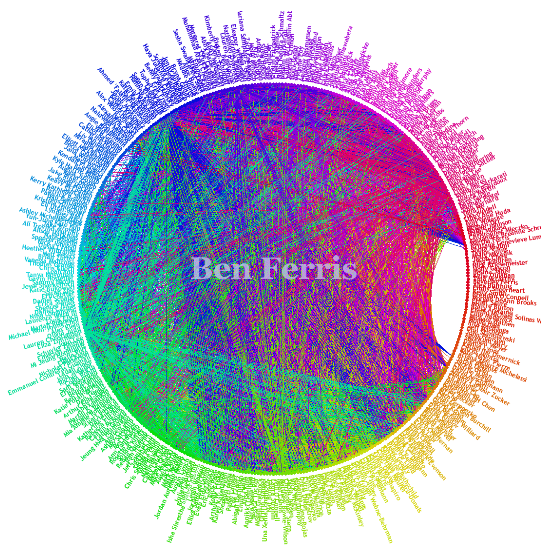

Figure 3: The friend wheel of Ben Ferris

3 Conclusions

Optimization is a powerful tool whose applications are becoming more and more widespread due to increased complexity of systems and enormous growth in data collection and provision. This note detailed a specific application of one optimization formulation that can be used as a tool for visualization of specific data and is available at: *http://apps.facebook.com/cscbfriend/*

It is clear that optimization is critical in this application, but the underlying process of building such models teaches students that development time is frequently dominated by data handling issues. This note also provides an interesting source of new test programs for QAP, where the overall running time of the application is still dominated by the time to perform optimization.

Finally, it is clear that modern optimizers must have a range of skills including the ability to do rigorous modeling in multiple formats, the ability and flexibility to merge model solutions to perform hard underlying tasks, and computational skills to extract and manipulate data and display the results of optimization in clear and concise ways. It remains an exciting time to be an optimizer!

References

[1] K. Anstreicher, N. Brixius, J.-P. Goux, and J. Linderoth. Solving large quadratic assignment problems on computational grids. *Mathematical Programming*, 91:563–588, 2002.

[2] A. P. Burgard, P. Pharkya, and C. D. Maranas. Optknock: A bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnology and Bioengineering*, 84(6):647–657, 2003.

[3] T. C. Koopmans and M. J. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25:53–76, 1957.

[4] E. M. Loiola, N. M. M. Abreu, P. O. Boaventura-Netto, P. Hanh, and T. Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176:657–690, 2007.

[5] C.A. Oliveira, P.M. Pardalos, and M.G.C. Resende. GRASP with path-relinking for the QAP. In T. Ibaraki and Y. Yoshitomi, editors, *Proceedings of the Fifth Metaheuristics International Conference (MIC2003)*, pages 57–1 – 57–6, Kyoto, Japan, 2003.

[6] P. M. Pardalos, F. Rendl, and H. Wolkowicz. The quadratic assignment problem : A survey of recent developments. In P. M. Pardalos and H. Wolkowicz, editors, *Quadratic Assignment and Related Problems*, volume 16 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 1–42, Rhode Island, 1994. AMS.

[7] S. Sahni and T. Gonzales. P-complete approximation problems. *Journal of the Association for Computing Machinery*, 23:555–565, 1976.

[8] D. M. Shepard, M. C. Ferris, G. Olivera, and T. R. Mackie. Optimizing the delivery of radiation to cancer patients. *SIAM Review*, 41:721–744, 1999.