

Družinkátor math documentation

Petr Brož

September 16, 2024



1 Intro (What even is this?)

Družinkátor is a tool for finding optimal placement of the KOČJ summer camp (TOČJ from now on) attendees into companies. This is achieved by solving a mixed integer program (MIP) with PySCIPOpt¹.

For now, the purpose of this file is mainly to describe the formulation of this MIP, not to provide documentation of Družinkátor's functions, since the author (foolishly) believes the comments within the code to be sufficient.

2 Problem statement

Each attendee of TOČJ must be placed into exactly one of four companies. This assignment is fixed for the whole duration of the camp. Not all attendees are present for the full duration – they might for example only partake in the first or second week of the camp.

Three main objectives are pursued when deciding the placement of people:

- **Respecting incompatibilities:**

Some people should not be placed together in the same company, e.g. siblings or romantically involved couples.

- **Balancing companies:**

The companies should be balanced in their manpowers, so as to make competition between the companies more practical. It is also desirable to spread other characteristics or skills evenly between the companies – e.g., if 4 students of Matfyz attend the camp, it is better to spread them 1-1-1-1 than to jam them all into one company².

- **Interannual mixing:**

For veterans attending for multiple years, an attempt should be made to vary whom they share companies with. E.g., if person A and person B were placed in the same company last year and the year before that, we should try not to place them in the same company this year.

3 Mixed integer program formulation

SCIPopt expects a MIP formulated as

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0 \quad \forall i \in I \\ & l_j \leq x_j \leq u_j \quad \forall j \in J \\ & x_k \in \mathbb{Z} \quad \forall k \in K \end{aligned} \tag{MIP formulation}$$

where $f(\mathbf{x})$ and $g_i(\mathbf{x})$ are (nonlinear) functions of the vector of all optimization variables \mathbf{x} and J, K are subsets of the set containing all indices of \mathbf{x} .

¹<https://github.com/scipopt/PySCIPOpt>

²This balancing act should be viewed on a day by day basis, taking into account the different attendances of individual people. If we want balanced competition on monday, then the companies should be balanced on monday, etc.

3.1 Membership variables

Let's begin by modelling the assignment of people into companies. The *membership* optimization variables, arranged into the membership matrix M , take care of that.

M is a $4 \times p$ matrix where p is the total number of people attending TOČJ. Each column then represents one person. Membership variable $M_{i,j}$ represents the truth value of the statement "Person j is assigned to company i ." For example, if M took the following form:

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix},$$

then that would represent the situation where person 0 is placed in company 0, persons 1 and 2 are placed in company 3 and the rest in company 2, with company 1 having no members.

To ensure that each person is placed into exactly one company, following constraints are placed on the variables:

$$M_{i,j} \in \mathbb{Z} \quad \forall i, j. \quad (1)$$

$$0 \leq M_{i,j} \leq 1 \quad \forall i, j. \quad (2)$$

$$\mathbf{1}_4^T \cdot M = \mathbf{1}_p^T, \quad (3)$$

where $\mathbf{1}_n$ is a column vector of ones of length n .

The $M_{i,j}$ variables are the crucial ones – although more optimization variables are introduced later, the solution of the optimization problem is fully captured in the values of $M_{i,j}$.

Note that constraint 1 is what turns this problem from what would otherwise be a tame quadratic program into a MIP.

3.2 Cost function

The function f from MIP formulation, which will be minimized by the solver, consists of 3 terms:

$$f = AAE_{sum} + CCP_{sum} + SP_{sum}. \quad (4)$$

How the terms come about will be elaborated later. *AAE* stands for Absolute Attribute Error – this term penalizes imbalance between the companies. *CCP* stands for Co-Company Penalty, and the second term penalizes lack of interannual mixing. *SP* stands for Soft Penalty and the last term penalises violations of soft constraints.

3.3 Shared Company Matrix

To aid in formulating further parts of the problem, I introduce the Shared Company Matrix, or *SCM* for short. *SCM* is a $p \times p$ matrix, where $SCM_{i,j}$ is equal to 1 if persons i and j are placed in the same company and 0 if they are not. This is achieved by setting

$$SCM_{i,j} = \mathbf{1}_4^T \cdot \begin{pmatrix} M_{0,i} \\ M_{1,i} \\ M_{2,i} \\ M_{3,i} \end{pmatrix} \odot \begin{pmatrix} M_{0,j} \\ M_{1,j} \\ M_{2,j} \\ M_{3,j} \end{pmatrix}, \quad (5)$$

where \odot denotes elementwise multiplication.

3.4 Handling incompatibilities

Introducing a constraint in the form $SCM_{i,j} = 0$ ensures that i th and j th person will not be placed together in the same company. Setting the right side to 1 would instead require that they be placed in the same company.

Alternatively, those limitations can be formulated as soft constraints – in that case, the penalty for violating such soft constraint is added to SC_{sum} .

3.5 Balancing companies