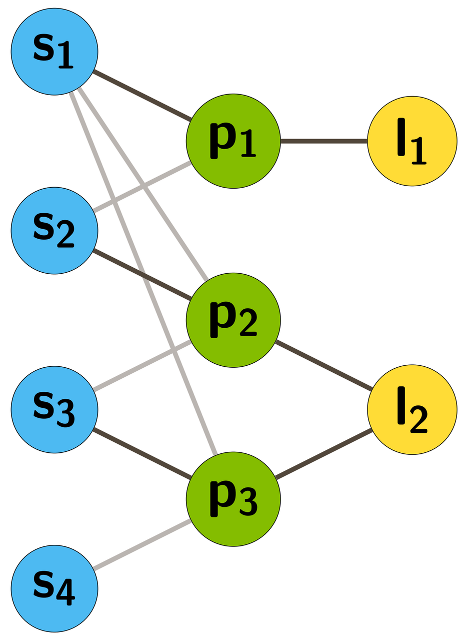
Integer Programming for Student-Project Allocation

Frances Cooper and David Manlove, University of Glasgow, Glasgow, Scotland, UK

The Student-Project Allocation Problem

The Student-Project Allocation problem with lecturer preferences over Students (SPA-S)

- A set of students $S = \{s_1, s_2, \ldots, s_{n_1}\}$, projects $P = \{p_1, p_2, \ldots, p_{n_2}\}$ and lecturers $L = \{l_1, l_2, \ldots, l_{n_3}\}$
- Each project is offered by a unique lecturer
- Students have preferences over projects, lecturers have preferences over students
- Projects and lecturers have upper quotas



Stable Matching

A **stable matching** in SPA-S is an assignment of students to projects such that capacities are respected and there is no student-project pair (s_i, p_j) where s_i and l_k, the lecturer offering p_j, have an incentive to deviate from the assignments (if any) and form a pairing.

- Every instance of SPA-S must admit a stable matching [1]
- A stable matching can be found in linear time [1]

Adding Ties and Lecturer Targets

The Student-Project Allocation Problem with lecturer preferences over Students including Ties and Lecturer targets (SPA-STL) extends SPA-S.

- Ties are allowed in lecturer (and student) preference lists
- Projects and lecturers have lower quotas
- Lecturer targets indicate a target number of lecturer allocations

Optimisations

- Similar definition of **stability** for SPA-S applies to SPA-STL
- **maximum size** - maximum number of students are assigned
- **load balancing** - variety of comparisons between the number of lecturer allocations and the lecturer targets

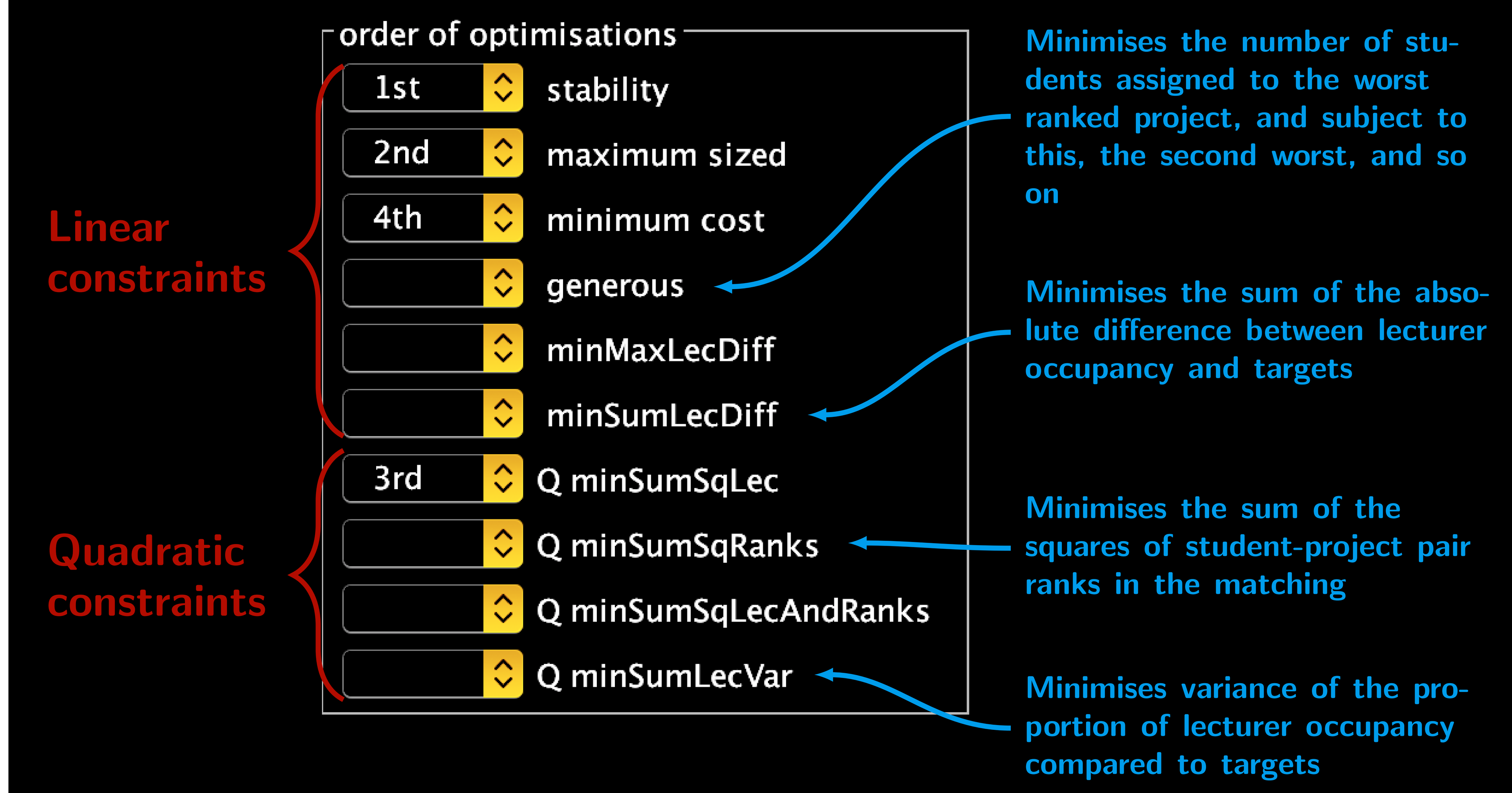
New Integer Programming Model

Integer Programming (IP) is a computational technique which can deal with hard problems. Finding a maximum stable matching in an instance of SPA-STL is NP-hard and so an **IP model** was developed for instances of SPA-STL with the aim of investigating the scalability of the IP model with changes in instance complexity and size, and also investigating changes in matching characteristics when altering instance parameters such as preference list length and probability of ties.

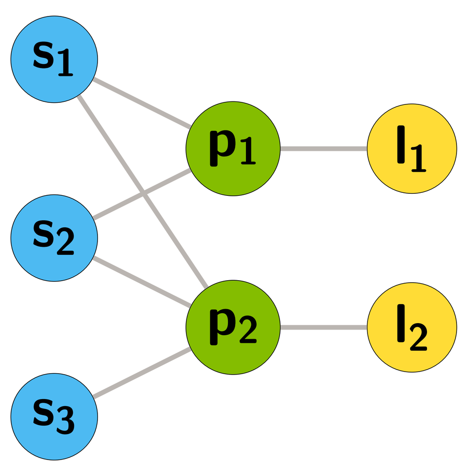
- New integer inequalities and objective functions created for stability constraints and load balancing optimisations

Java Application

- Integer Program accessed by Java application
- Optimisations can be performed in any order



Conflicting Objectives Example



s_1: $(p_1\ p_2)$
s_2: $p_2\ p_1$
s_3: p_2

p_1: LQ: 0, UQ: 2
p_2: LQ: 0, UQ: 2

l_1: $(s_1\ s_2)$ LQ: 0, UQ: 2
l_2: $(s_3\ s_1)\ s_2$ LQ: 0, UQ: 2
all lecturer targets 1

Objectives A:
Opt 1: stable
Opt 2: maximum size
$M=\{(s_1,p_1)(s_2,p_2)(s_3,p_2)\}$

Objectives B:
Opt 1: minimise the sum of lecturer differences
Opt 2: maximum size
$M=\{(s_1,p_1)(s_2,p_2)\}$

Generated Results

Input datasets were generated randomly and vary parameters such as the prevalence of ties in preference lists (Figure 1).

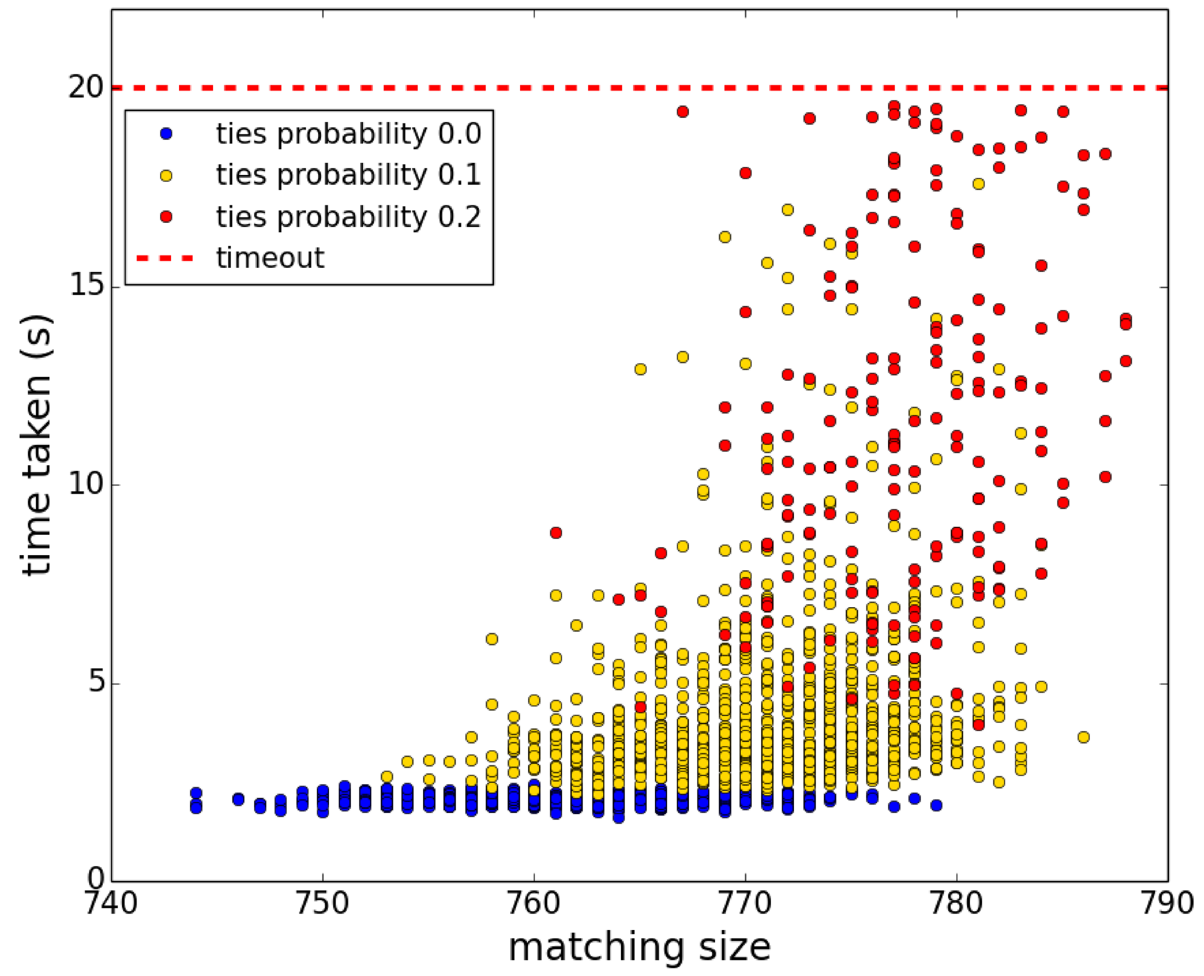


Figure 1: Preliminary results. Changes in time taken to solve instances versus matching size when varying preference list ties probability. In all cases there are 800 students, 350 projects, 200 lecturers, all lower quotas 0, all upper quotas 1000.

- 0%, 2.1% and 82% instances timeout for 0.0, 0.1 and 0.2 ties probability respectively
- As tie probability increased, matching size and time taken to solve also increased

Real World Results

In addition to generated data, the IP model has been used on **several real world scenarios** including student project allocations for the University of Glasgow, the University of Edinburgh and the University of Leeds, and teacher-region allocations for TeachFirst. Each scenario had varying requirements but in several instances the IP model replaced a manual allocation process which was both time-consuming and unlikely to result in an optimal outcome.

University of Glasgow UNIVERSITY OF LEEDS TeachFirst

References

[1] David J. Abraham, Robert W. Irving, and David F. Manlove. Two algorithms for the student-project allocation problem. Journal of Discrete Algorithms, 5:73-90, 2007.