algorithms

MDPI

Article

Allocating Students to Industry Placements Using Integer Programming and Ant Colony Optimisation

Dhananjay Thiruvady [1,*], Kerri Morgan [1], Susan Bedingfield [2] and Asef Nazari [1]

[1] School of Information Technology, Deakin University, Geelong, VIC 3217, Australia; kerri.morgan@deakin.edu.au (K.M.); asef.nazari@deakin.edu.au (A.N.)
[2] Clayton School of Information Technology, Monash University, Clayton, VIC 3800, Australia; sue.bedingfield@gmail.com
* Correspondence: dhananjay.thiruvady@deakin.edu.au

Abstract: The increasing demand for work-ready students has heightened the need for universities to provide work integrated learning programs to enhance and reinforce students' learning experiences. Students benefit most when placements meet their academic requirements and graduate aspirations. Businesses and community partners are more engaged when they are allocated students that meet their industry requirements. In this paper, both an integer programming model and an ant colony optimisation heuristic are proposed, with the aim of automating the allocation of students to industry placements. The emphasis is on maximising student engagement and industry partner satisfaction. As part of the objectives, these methods incorporate diversity in industry sectors for students undertaking multiple placements, gender equity across placement providers, and the provision for partners to rank student selections. The experimental analysis is in two parts: (a) we investigate how the integer programming model performs against manual allocations and (b) the scalability of the IP model is examined. The results show that the IP model easily outperforms the previous manual allocations. Additionally, an artificial dataset is generated which has similar properties to the original data but also includes greater numbers of students and placements to test the scalability of the algorithms. The results show that integer programming is the best option for problem instances consisting of less than 3000 students. When the problem becomes larger, significantly increasing the time required for an IP solution, ant colony optimisation provides a useful alternative as it is always able to find good feasible solutions within short time-frames.

Keywords: allocation; matching; industry placements; integer programming; ant colony optimisation

Citation: Thiruvady, D.; Morgan, K.; Bedingfield, S.; Nazari, A. Allocating Students to Industry Placements Using Integer Programming and Ant Colony Optimisation. *Algorithms* **2021**, *14*, 219. https://doi.org/10.3390/a14080219

Academic Editors: Yun-Chia Liang, Mehmet Fatih Tasgetiren, Quan-Ke Pan and Hsiang-Ling Chen

Received: 5 July 2021
Accepted: 20 July 2021
Published: 21 July 2021

1. Introduction

There is an increasing emphasis on *Work Integrated Learning* (WIL) in universities and other tertiary education providers [1] As a result, the number of students undertaking WIL is growing rapidly, and the need to effectively manage the processes involved is increasingly important. The Australian National Strategy of Work Integrated Learnin (http://cdn1.acen.edu.au/wp-content/uploads/2015/03/National-WIL-Strategy-in-university-education-032015.pdf, accessed 21 July 2021) proposes actions in eight key areas, including: (1) developing university resources, processes and systems to grow WIL and engage business and community partners and (2) addressing equity and access issues to enable students to participate in WIL.

The problem of allocating students to industry placements is of significant interest. Several factors need to be considered in order to achieve an acceptable allocation. In particular, we consider the Industry Based Learning (IBL) allocation problem where students are allocated to industry placements. The aim is to provide an excellent educational experience to students whilst maximising industry partner satisfaction (and thus engagement). In addition, the allocation aims to provide gender diversity within placements and ensures that students allocated consecutive placements experience a diversity of sectors. Constraints

arising from inclusive practices are increasingly important. For example, ref. [2] discusses this in the context of resettlement of refugees and comments that 'the remaining question is how to design mechanisms that cater to diversity objectives while still satisfying desirable properties'.

We have considered real problem instances, but it is notable that the number of student placements have been increasing overtime. Moreover, in certain university departments (e.g., nursing), the number of placements to be filled can involve a very large number of students and placements (>2000). A recent survey [3] of the placement of students in nurse practitioner programs in the United States found that only 12% of facilities used electronic means for placing students. They comment that 'an investment in these platforms could reduce faculty time for securing clinical placements'. Discussing a related problem of assigning students to medical schools, ref. [4] investigate the need to improve efficiency and reduce costs while ensuring that the pairing of students to medical schools is fair and ensures that students are allocated to suitable schools.

Integer programming (IP) is an effective approach to solving optimisation problems [5]. The IBL problem is one that can be modeled effectively as an IP, particularly because the underlying structure is that of the assignment or bi-partite matching problem [6], which can be solved efficiently by commercial solvers. It is known that a linear assignment problem (LAP) can be solved in a polynomial time, and an IP problem is categorised as an NP-complete problem from a complexity point of view [7]. However, when problem instances are large and consist of a very large number of variables and/or complicating constraints, the IP can struggle to find feasible solutions, let alone prove optimality. In such cases, meta-heuristics [8] provide an alternative way of generating good solutions in reasonable time-frames (note that metaheuristics on their own have limitations; in particular, they cannot provide guarantees of solutions they find and they often struggle with problems which consist of nontrivial hard constraints) and in this study ant colony optimisation (ACO) [9] is the meta-heuristic of choice. ACO encompasses a collection of heuristics originally based on the behaviour of ants, which can be used to solve optimisation problems efficiently [9].

Compared to previous studies that investigate similar problems, the IBL problem is unique in that it considers aspects such as partner happiness and gender equity. Moreover, to solve this problem, we develop an exact method (IP) and heuristic method (ACO) to tackle the problem. The motivation for using ACO is that we can model potential solutions to the problem using sequences of students. ACO has proven to be effective in previous studies which modelled the problem in this way [9]. The results of these algorithms are compared with the actual schedules that were manually created using data from previous semesters at the Faculty of IT, Monash University. Additionally, we generate problem instances with varying numbers of students, companies and placements, which are generated based on data for previous semesters. These problem instances allow testing of the algorithms for scalability and provide insight into how the algorithms would perform on allocation problems such as those arising in other domains that have much larger allocation requirements (for example, nursing).

This study makes four contributions:

- Introduces a real-world allocation problem of allocating university students to industry placements;
- Identifies key components of real data and uses this information to generate meaningful artificial data, which can serve as a benchmark dataset for the community;
- Develops exact and heuristic methods based on IP and ACO to solve the allocation problem;
- Investigates the performance of these methods on real data and analyses different aspects of the problem (e.g., proportion of students to placements) using artificial data.

The paper is organised as follows. In Section 3, the IBL problem is described; then, in Section 4, a model for this problem is given. The ACO approach is presented in Section 5. We then describe the random generation of instances of the IBL problem based on distributions from real data in Section 6. The results and performance of the IP and ACO for

these instances is detailed in Section 7. The paper concludes in Section 8 and possibilities for future work are discussed.

2. Previous Work

The problem of finding *matchings* commonly occurs in a number of real-world situations [6]. The basic idea underlying the problem is to identify pairings between objects in any two given sets. Hall's marriage problem [10] is a classic matching problem, where given a set of men and a set of women, the task is to maximise the number of pairs of men and women that can be 'married'. Similarly, a number of allocation problems include those where students need to be allocated to colleges [11] or projects [12], residents allocated to hospitals [11,13], and roommates allocated to rooms [14]. In this paper, we consider allocations of students to placements, and so without loss of generality will refer to the two sets as 'students' and 'placements'.

In many cases, the set of students and (or) the set of placements have a list of preferred partners in the corresponding set. This is referred to as *matching under preferences* (e.g., see [6]). Preferences can be two-sided or, alternatively, only one set may have preferences. Preferences are not given to every member of the set, and a member's preference can be totally or partially ordered. For example, a placement may prefer student s_1 over s_2, but be indifferent between possible matchings with students s_1 and s_3. This gives rise to problems such as the stable matching problem (a matching is *stable* if there are no two pairs in the matching, say (s_1, p_1) and (s_2, p_2), where s_1 prefers p_2 to p_1 and s_2 prefers p_1 to p_2) with partially ordered and incomplete lists (SMPI) and the hospital/residents problem with partially ordered lists (HRP) [15]. In many of these cases, the problem is NP-hard (e.g., [15]).

Irving defines three types of stable matchings [16]. A *weak stable matching* is one where there are no pairs, (s_1, p_1) and (s_2, p_2), in the matching such that s_1 prefers p_2 to p_1 and s_2 prefers p_1 to p_2. Strong stability occurs where, for any two pairs (s_i, p_i), $i = 1, 2$, student s_1 strictly prefers their own partner to p_2 and s_2 either strictly prefers p_2 to p_1 or equally prefers both p_1 and p_2. (Note: in this case at most one of the two pairs can hold an equal preference.) Super-stability occurs where, for any two pairs (s_i, p_i), $i = 1, 2$, each s_i strictly prefers their own partner to the other's partner. Irving and Manlove [17] discuss the case where there are ties in the preference lists which may also be incomplete. In this case, finding a maximum weakly stable matching is NP-hard. According to the findings of [16,17], the WIL problem is similar to a strong stable matching problem with ties in the preference lists and incomplete preference lists, and hence it is an NP-hard problem.

Several types of matching, assignment and allocation problems have been solved with IPs [18–21]. Klaus et al. [18] provide an overview of matching under preferences and thoroughly review the literature in the area. Cholette [19] develops an embellished transportation IP model to find a matching between wineries to distributors. In at least one example, the model proposed a matching that resulted in a long-term partnership. Al-Yakoob and Sherali [20] examine the problem of assigning employees to 86 different gas stations distributed around Kuwait. They consider a multi-stage mixed integer programming approach, where in the first stage employees are assigned to stations and in the second stage individual shifts for the employees are determined. Naik et al. [21] consider the problem of online resource allocation in a grid environment. They use a linear programming approach, which is able to efficiently find new solutions when new information is available. Ref. [22] use an IP which optimises utility (student preference ratings) to allocate students to capstone projects with the constraint that each project requires a combination of students from different disciplines.

Meta-heuristics have also been applied to matching problems [23,24]. Harper et al. [23] investigate a genetic algorithm for assigning students to projects with multiple objectives. Their proposed approach compares well with integer programming, but they observe that nonlinear objectives can be handled by their method, which is more suitable for project assignment. Gupta et al. [24] apply a genetic algorithm for matching security vulnerabilities

to security profiles within an organisation. They show that the genetic algorithm is able to efficiently achieve maximum vulnerability coverage with a minimum cost profile.

ACO is proven on a number of problems, such as job scheduling [25], traffic assignment [26], image edge detection [27], and assembly line balancing [28,29]. There have also been studies with ACO and optimisation of task allocation [30,31] and multi-objective resource allocation [32,33]. Shyu et al. [31] show that ACO can be effective on the cell assignment problem in PCS networks where cells need to be assigned to switches while minimising cable and hand-off costs.

3. Problem Specification

The IBL Problem (IBLP) can be formally defined as follows. A set of n students $S = \{s_1, \ldots, s_n\}$ are to be allocated to industry placements offered by m companies $C = \{c_1, \ldots, c_m\}$. Each company $c \in C$ has a limited number, r_c, of placements available. The total number of placements available is defined as $R = \sum_{c \in C} r_c$. Each company $c \in C$ is associated with one or more of b business sectors. Let $B_i \subseteq C$ denote the companies associated with business sector i and $Q = \{B_1, B_2, \ldots, B_b\}$.

Each company interviews a subset of the students and gives each interviewed student a *preference rating*. The preference rating assigned to student s by company c is denoted by $p_{sc} \in \{1, 2, 3\}$. The following is the meaning of the preference values:

Rating 1: The company has a strong preference for the student;

Rating 2: The company has a preference for the student but would prefer a student with rank 1;

Rating 3: The company does not have a placement that is suitable for the student.

Students who are not interviewed by a company are automatically given a rating of 3 by that company. The following data are used in the allocation:

1. **Gender**: the students are partitioned into two subsets of S which are denoted female $F \subseteq S$ and male $M \subseteq S$. Some companies explicitly require an equitable gender mix;
2. **Degree type**: there are d different degree types. For example, students may be enrolled in a computer science, software engineering, or an information technology degree. Companies may have a student preference partly based on degree type;
3. **Previous placement sector**: For each student s, the list of the sectors of any previous placements undertaken is denoted by Q_s, which is defined as follows:

$$Q_s = \bigcup_{i \in E_s \subset \{1,\ldots,b\}} B_i \tag{1}$$

where E_s is the set of business sectors for the student.

There are three sets of constraints to be satisfied when creating a feasible allocation: (a) if there are more students than placements available, then all placements should be filled (preferences permitting), (b) if not all placements can be filled, companies requiring fewer placements should be given preference, and (c) students taking more than one placement cannot undertake more than one placement in a given sector.

4. Integer Programming Model

A preliminary IP model is created for the IBLP, which is defined as follows. We define binary variables x for allocating students to companies as follows: $x_{sc} = 1$ if student s is allocated to company c. Additionally, binary variables y are defined for counting purposes. Let A_c be the number of students allocated to company c. The vector y_c gives a unary representation of A_c. Here, $y_{ck} = 1$ if $k \leq A_c$ and 0 otherwise. The length of this unary vector is modulo $|S| = n$ for every company. For example, $y_c = [1, 1, 1, 0, 0]$ would represent three students allocated to company c from a total of $|S| = 5$ students available. This representation enables an efficient way to evenly distribute students among companies when there are insufficient students to fill the placements. The details of the notation used for the model are summarised in Table 1.

Table 1. The table of notations used in the IP model.

Notation	Definition
Sets	
S	The set of n students $S = \{s_1, \ldots, s_n\}$
C	The set of m companies $C = \{c_1, \ldots, c_m\}$
B_i	The set $B_i \subseteq C$ of companies in sector $i \in [1, b]$
Q	The set of sets of companies belonging to each of the sectors: $Q = \{B_1, B_2, \ldots, B_b\}$
Q_s	The set of companies belonging to sectors where student s has been previously placed
M	The set of male students $M \subseteq S$
F	The set of female students $F \subseteq S$
Parameters	
r_c	The number of available placements in company c
R	The total number of placements available: $R = \sum_{c \in C} r_c$
p_{sc}	$p_{sc} \in \{1, 2, 3\}$ is the preference rating of student s by company c
v_c	Constants which are used to prioritise companies with relatively low numbers of placements
Variables	
x_{sc}	Binary variables where $x_{sc} = 1$ if student s is allocated to company c
y_c	A vector of binary variables providing a uniary representation of the number of students allocated to company c

There are multiple components in the objective function of the problem associated with company preferences. The aim is to maximise the number of allocations at the same time whilst respecting gender equity. The IP model is as follows:

$$\text{Maximise} \quad \sum_{s \in S} \sum_{c \in C} (\mathcal{L} - p_{sc}) x_{sc} - \sum_{c \in C} \sum_{k=1}^{n} v_c y_{ck} - \left(\sum_{c \in C} \left| \sum_{s \in M} x_{sc} - \sum_{s \in F} x_{sc} \right| \right) \tag{2}$$

$$\text{Subject to} \quad \sum_{c \in C} x_{sc} \leq 1 \qquad \forall s \in S \tag{3}$$

$$\sum_{s \in S} x_{sc} \leq r_c \qquad \forall c \in C \tag{4}$$

$$x_{sq} = 0 \qquad \forall s \in S,\ \forall q \in Q_s \tag{5}$$

$$\sum_{s \in S} x_{sc} - \sum_{k=1}^{n} y_{ck} = 0 \qquad \forall c \in C \tag{6}$$

$$y_{ck} - y_{c(k-1)} \leq 0 \qquad \forall c \in C,\ \forall k \in \{2, \ldots, n\} \tag{7}$$

$$x_{sc}, y_{ck} \in \{0, 1\}, \qquad \forall s \in S, \forall c \in C,\ \forall k \in \{1, \ldots, n\} \tag{8}$$

The problem is defined as a maximisation problem, which means the penalty terms are negated in the objective function. The first term in the objective sums the allocated preference values for each student s allocated to company c using the preference weights p_{sc}. Here, a constant $\mathcal{L}$ is used to ensure the objective values are positive valued. (In principle, $\mathcal{L}$ is not required, though, positive values makes comparisons between the approaches more straightforward.) The second term includes a weight $v_c = \sum_{k=1}^{r_c} k$, where r_c is the number of students requested by company c, to give priority to companies requesting smaller number

of placements in case the total number of students is less than the total number of available placements. The third term in the objective ensures an equitable gender mix is provided for those companies that request it.

Constraint (3) ensures that a student is assigned to at most one company. Constraint (4) limits the number of students allocated to a company to be at most the number of placements available. Constraint (5) is the sector constraint ensuring a student cannot be allocated to a company with the same sector as any of the student's previous placements. The notation Q_s is used to denote the set of companies belonging to the same sector(s) as the previous placements of student s, which is defined in (1). Constraints (6) and (7) ensure that the appropriate weight is applied to the number of students assigned to a company.

An Illustration of the Problem

In this section, an example of the IBL problem is presented to further illustrate the IP model. There are three students $S = \{s_1, s_2, s_3\}$, where student s_3 is female and the others are male. Moreover, there are two companies $C = \{c_1, c_2\}$ with two and one placement(s) available, respectively. Furthermore, assume that student s_1 has previous experience in the business sector of Company c_1. Table 2 provides the list of preferences.

Table 2. The table of preferences and the number of available placements of companies.

Student \ Company	c_1	c_2	Gender
s_1	$p_{11} = 1$	$p_{12} = 3$	male
s_2	$p_{21} = 2$	$p_{22} = 1$	male
s_3	$p_{31} = 3$	$p_{32} = 2$	female
r_c	$r_1 = 2$	$r_2 = 1$	
v_c	$v_1 = 3$	$v_2 = 1$	

With these parameters fixed, the IP model optimises the allocation of students to companies respecting the preferences and constraints. The model resulting from this problem description is provided in Appendix A. The optimal solution of this IP model is $x_{11} = 0$, $x_{12} = 1$ (Student 1 is allocated to Company 2), $x_{21} = 1$ (Student 2 is allocated to Company 1), $x_{22} = 0$, $x_{31} = 1$ (Student 3 is allocated to Company 1), $x_{32} = 0$. Here, y_1 is unary $[1, 1, 0]$, that is, Company 1 has two students, and y_2 is unary $[1, 0, 0]$, that is, Company 2 has one students.

5. Ant Colony Optimisation

ACO is proven on a number of practical problems [9], and among the ACO implementations, $\mathcal{M}$ax-$\mathcal{M}$in Ant System ($\mathcal{MM}$AS) has been one of the most effective practical ACO variants [34]. Hence, this approach is used as the basis for our ACO implementation for the IBL problem. We also found that the heuristic selection of the ant colony system [35] provides improved intensification; hence, we use this step for the selection of the solution components. Algorithm 1 shows the $\mathcal{M}$ax-$\mathcal{M}$in Ant System applied to the IBL problem.

A solution is represented by π, which is a vector of placements. The length of π corresponds to the number of students and $\pi_i = j$ means that student i is allocated to placement j. (Note that this is not an allocation to a company but to a specific placement of a company. This difference is important in the case where a company requires more than one placement.) The input to the algorithm is an instance of the problem. The pheromone trails are initialised (Initialise($\mathcal{T}$)), where $\tau_{ij} = \frac{1}{p_{ij}}$ is the desirability of allocating student i to placement j.

Algorithm 1 $\mathcal{M}$ax-$\mathcal{M}$in Ant System for the IBL allocation problem.

1: **Input:** An IBL instance
2: Initialise($\mathcal{T}$)
3: **while** termination conditions not satisfied **do**
4: **for** $j = 1$ to n_{ants} **do**
5: $\pi^j :=$ AllocateStudents($\mathcal{T}$)
6: **end for**
7: $\pi^{ib} := \arg\min_{j=1,\ldots,n_{ants}} f(\pi^j)$
8: $\pi^{bs} :=$ Update(π^{ib})
9: $\mathcal{T} :=$ PheromoneUpdate(π^{bs})
10: **end while**
11: return π^{bs}

A number of solutions (n_{ants}) to the problem are constructed in Lines 4–5. A solution is built incrementally by allocating a student to a placement using the pheromone information (AllocateStudents($\mathcal{T}$)). If a company preference for a student is 3 (not desired), the student is not allowed to be allocated to the corresponding placement(s). A placement for a student is selected using one of two rules. First, a random number is generated $q \in (0, 1]$ and compared with the parameter q_0. If $q < q_0$, placement k is chosen for student i according to:

$$k = \underset{j \in \mathcal{C}}{\operatorname{argmax}}\, \tau_{ij} \cdot \eta_{ij} \tag{9}$$

Otherwise, if $q \geq q_0$, placement k is chosen probabilistically according to:

$$P(\pi_i = k) = \frac{\tau_{ik} \cdot \eta_{ik}}{\sum_{j \in \mathcal{C}} \tau_{ij} \cdot \eta_j} \tag{10}$$

Here, η is a heuristic factor that biases the selection towards higher preference placements, i.e., $\eta_{ij} = \frac{1}{p_{ij}}$ for student i and placement j.

In Line 7, the iteration best solution is determined as the one with the minimum objective value (Equation (2)). The global best solution (Line 8) is then updated if a new better iteration of the best solution was found. Finally, in Line 9, the pheromone trails are updated in two steps. First, all solution components that are not found in the global best solution have evaporation applied:

$$\tau_{ij} = \tau_{ij} \cdot (1.0 - \rho) \tag{11}$$

Otherwise, the solutions components are updated with the allocation found in the global best solution. In addition to evaporation, this step additionally includes a reward factor to favour components in the global best solution. The pheromones are updated according to:

$$\tau_{ij} = \tau_{ij} \cdot (1.0 - \rho) + \delta \tag{12}$$

where $\delta = Q/f(\pi^{bs})$, $f(\pi^{bs})$ is the objective value of the global best solution determined according to Equation (2) and Q is selected such that $0.01 \leq \delta \leq 0.1$. The learning rate, ρ, was chosen to be relatively high at 0.1, and determined on a subset of the instances from the values [0.1, 0.05, 0.01].

6. Generating Problem Instances

The problem instances were generated using a number of distributions based on actual data from instances of the Faculty of Information Technology's Industry Based Learning (IBL) program. The distributions used were the sector distribution, the degree distribution, the student distribution, the preference distributions, and the number of placement distributions:

1. Sector distribution: the IBL industry partner companies come from a variety of sectors (e.g., banking, consulting, energy, not for profit). The *sector distribution* gives the probability distribution of a partner company belonging to a given sector. This distribution is based on the frequencies of partner companies' sectors based on historical data.
2. Degree distribution: students from a variety of degrees take part in the IBL program. The *degree distribution* gives the probability of a student belonging to a given degree. This distribution is based on the frequencies of student's degree type based on historical data.
3. Student rating distribution: the overall performance of students during the selection interviewing process depends partially on the students' abilities (soft and hard skills). Three categories were used to denote a student's rating level: strong, medium and weak based on the number of 1s the student received. A student receiving at least ten 1s was rated as 'strong'; a student receiving between three and nine 1s was rated as 'medium', and a student with at most two 1s was rated as 'weak'.
4. Preferences distributions: for each student, the likelihood of being allocated a 1, 2 or 3 depends on a number of factors. If a student is not interviewed by a company, by default they will be allocated a rating of 3 for that company's placements. Not all degree types are suitable for all the company sectors. For example, a company in sector X may prefer students in degree Y.
   Based on historical data, a matrix of preference distributions for each pair (company sector and the degree of the student) was obtained.
   For each given sector, degree pair, we weighted the preference distribution $[a, b, c]$, where a, b, and c are the frequency of 1s, 2s and 3s, respectively, as follows. For a student rated as medium, the distribution is unchanged. For a student rated as weak, the distribution becomes$[(1-3\alpha)a, (1+\alpha)b, (1+2\alpha)c]$, and for a student rated as strong, the distribution becomes $[(1+3\alpha)a, (1-\alpha)b, (1-2\alpha)c]$ where $\alpha \in [0, 1/3]$. A value $\alpha = 0.01$ means that on average a strong student will be ranked a '1' more often than a weak student.
5. Number of placements: different partners offer different numbers of placements in a given placement period. Additionally, the same partner might offer a different numbers of placements in different placement periods. For each sector, a distribution based on the frequencies of the number of placements offered by companies in this sector was used. This distribution is based on the historical data.

To provide further information about the distributions, the bar charts of the sector distribution and the degree distribution are provided in Figure 1. The exact names are removed and the terms deg$i$ and sec$j$ are used for privacy reasons.

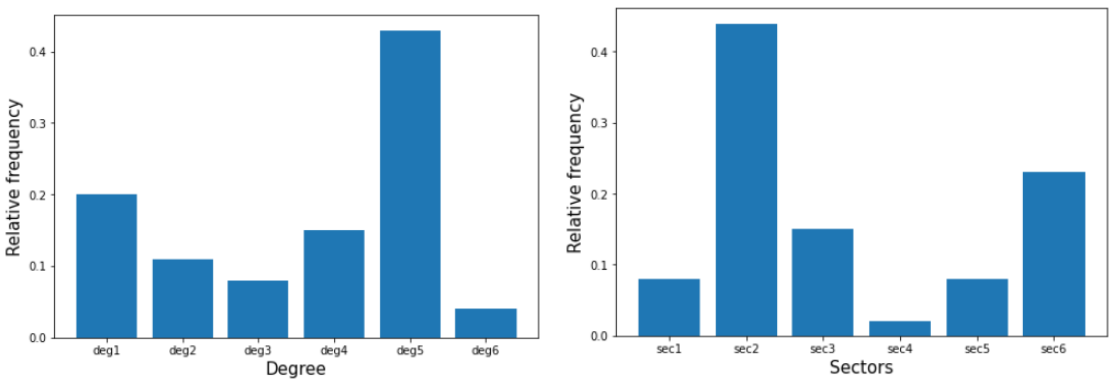

Figure 1. Bar charts representing the distributions of sectors and degrees used in generating the artificial data.

7. Experiments and Results

Two sets of experiments were undertaken. The first set of experiments compares the IP model with actual allocations from previous semesters. The second set of experiments is to determine how the model scales, as allocations must be generated in short run-times. There are two aspects to investigate in the second set of experiments. These are to test how the IP model scales with problem size (numbers of students and companies) and also to see how the performance varies with respect to the ratio of the number of students to the number of companies.

The parameter settings for $\mathcal{MM}$AS were determined by using previous studies as a guide and fine-tuning by hand. The initial pheromone values were chosen as $\tau_{ik} = \frac{1}{p_{ij}}$ and $n_{ants} = 10$, which is the number of solutions constructed per iteration. To choose between deterministic or probabilistic selection, $q_0 = 0.9$ was determined from the range [0.1, 0.2, 0.5, 0.8, 0.9]. The learning rate $\rho = 0.1$ was determined from the range [0.1, 0.05, 0.01].

7.1. Integer Programming and Real Data

The IP model was run on real datasets. The original allocations were built by hand taking several hours over a two week period of the scheduler's time. In a few cases in the manual allocation, private arrangement students were allocated to companies that had not interviewed them. This process was incorporated in the simulations by making a comparison with these students included and excluded from the student data set. In the former case, their ranking was given a value of 2.

The results are presented in Table 3. The first column specifies the year and semester (*Year-Sem.*). The second and third columns specify the number of students and the number of companies for the corresponding year/semester. For the original allocation, we report the average allocation (*Avg.*), the number of first preferences assigned (*#1 s*) and the number of last preferences assigned (*#3 s*). (Note: the IP model does not allow these by implementing it as a hard constraint.) For the IP model, the average allocation (*Avg.*), number of first preferences (*#1 s*) and time taken in seconds (*Time* (s)) to allocate are reported.

Table 3. Comparison between the IP model and actual (manually created) schedules. Adjusted runs (*) remove students who were not interviewed by any of the partners, as there are no preferences for these students. For 2012-1 and 2012-2, the IP model is unable to place 1 student, and for 2013-2, two students.

			Original Allocation			**IP Model**		
Year-Sem.	**Stu.**	**Comp./Depts.**	**Avg.**	**#1 s**	**#3 s**	**Avg.**	**#1 s**	**Time (s)**
2012-1	43	21	1.35	28	0	1.21	32	0.073
2012-1 *	42	21	1.33	28	0	1.21	32	0.084
2012-2	62	33	1.29	45	1	1.10	55	0.141
2012-2 *	61	33	1.28	45	1	1.10	55	0.135
2013-1	46	31	1.39	29	2	1.24	35	0.140
2013-2	53	31	1.38	39	6	1.04	46	0.123
2014-1	38	31	1.37	25	1	1.44	34	0.090
2014-2	60	41	1.24	46	2	1.05	57	0.162
2015-1	51	47	1.39	31	0	1.21	40	0.148
2015-1 *	50	47	1.38	31	0	1.20	40	0.147
2015-2	75	65	1.25	58	2	1.09	68	0.265
2016-1	44	52	1.41	26	0	1.25	33	0.139
2016-1 *	42	52	1.38	26	0	1.21	33	0.138
2016-2	82	55	1.13	71	0	1.02	80	0.257

In all but one instance, the IP model achieves better average allocation outcomes than the manual allocation. The only exception is 2014-1, where the IP model has a higher average allocation. The reason this happens is because the sector constraints were overridden by the manual allocation and as this is implemented as a hard constraint in the IP model, the IP is unable to find improvements. However, if this constraint is relaxed in the IP model, the average allocation improves to 1.15, which is significantly better than the manual allocation.

7.2. Investigating Artificial Data

Problem instances were generated using the respective distributions detailed in Section 6. The numbers of students and the numbers of companies were limited to values that would be practically feasible. The number of students ranged between 250 and 900 and the number of companies ranged between 200 and 600. The number of students was chosen to be at least as many as the number of companies.

The problem generator and the IP model were implemented in Python 3.6. In order to solve the IPs, Gurobi 8.0.0 [36] was used. Each run was given 30 min of wall-clock time. During an actual allocation, 30 min was seen as a time limit that was acceptable. Since the real instances always had less than 100 students, the run-times were within a second. For those instances which could not be solved in the 30 min limit, an additional run with a limit of 2 h was conducted to determine if any solution could be found. For $\mathcal{MM}$AS, 25 runs per instance were conducted and the average values and associated standard deviations across all the runs are reported.

Results

Tables 4 and 5 present the results of the IP model and $\mathcal{MM}$AS on problem instances generated by the various distributions discussed in Section 6. The first two columns specify the number of students (*Stu.*) and companies (*Comp.*), respectively. The next four columns specify the criteria used to measure the quality of solutions (three columns) and the time taken (in seconds) by the IP. The first is *#1 s*, or the number of students assigned to companies with a preference rating of 1. The larger the number of students with *#1 s*, the better the solution. Second, *Avg.* or the average allocation of student preferences assigned to the companies in a solution. Third, *Obj* is objective value in Equation (2). The next six columns are associated with $\mathcal{MM}$AS. Since 25 runs per instance with $\mathcal{MM}$AS were conducted, standard deviations (SDs) associated with each criteria are also shown.

For most instances (up to 3000 students and 3000 placements), the IP is able to find optimal solutions, often quite quickly. The IP solutions are better than the $\mathcal{MM}$AS solutions considering number #1 preferences and average allocations for all these instances. When the proportion of students to companies is high (25% and 50%), solutions with students having a preference of 1 are found for all placements. Beyond 3000 students, the IP struggles to find solutions within the 30 min time limit. In these cases, $\mathcal{MM}$AS finds solutions, but it is not possible to quantify their quality. Despite no guarantee on the quality of the solution, at the very least a feasible solution to any large instance can be found within five minutes. For real problems, these solutions can be provided to the industry partners in lieu of a manual allocation that can be very problematic in construction.

Not surprisingly, the IP run-times progressively increase as the problem size increases. Runs take more than a minute when there are more than 1000 students and 750 companies. However, this is still a feasible time-frame to obtain good solutions. As previously discussed, $\mathcal{MM}$AS provides an alternative when the IP run-times become too large. This leads to a distinct advantage in real student allocation systems, where solutions can be built, modified (e.g., incorporating a constraint such as a student must not be allocated to a company), and re-built quickly if the user is unhappy with the allocation for some reason.

Table 4. The results of the IP model and $\mathcal{MM}$AS in instances with varying number of students (250–900) and companies (200–700). Stu.: number of students; Comp.: number of companies; **#1 s**: number of times a rating of 1 was achieved; Avg.: average allocation.

		IP				$\mathcal{MM}$AS					
Stu.	**Comp.**	**#1 s**	**Avg.**	**Obj.**	**Time (s)**	**#1 s**	**SD**	**Avg.**	**SD**	**Obj.**	**SD**
200	50	63	1.00	441	0.69	60.40	1.06	1.03	0.02	431.40	1.06
200	100	120	1.00	852	1.36	109.00	1.98	1.08	0.02	834.00	1.98
200	150	173	1.14	1419	3.15	142.40	3.22	1.29	0.02	1385.08	3.20
200	200	106	1.47	1442	3.22	76.36	3.58	1.62	0.02	1405.32	4.83
300	75	109	1.00	777	1.90	63.84	2.91	1.41	0.03	724.84	2.91
300	150	223	1.00	1590	3.70	195.08	2.78	1.12	0.01	1558.08	2.78
300	225	271	1.00	1944	5.63	267.92	1.65	1.01	0.01	1933.92	1.65
300	300	258	1.14	2272	7.73	202.44	4.61	1.33	0.02	2164.00	8.51
400	100	120	1.00	867	2.80	67.88	1.37	1.43	0.01	809.88	1.37
400	200	258	1.14	2070	5.82	163.04	4.70	1.45	0.02	1968.04	4.70
400	300	320	1.20	2760	14.30	285.72	2.47	1.26	0.01	2668.64	11.56
400	400	322	1.15	2857	14.34	293.56	3.73	1.21	0.01	2715.44	12.67
500	125	188	1.00	1324	5.35	175.04	1.25	1.06	0.01	1304.04	1.25
500	250	342	1.00	2450	11.38	326.88	1.86	1.04	0.01	2427.88	1.86
500	375	388	1.12	3203	17.60	369.64	1.67	1.16	0.00	3131.76	7.23
500	500	420	1.15	3688	23.51	391.92	2.26	1.19	0.01	3512.96	10.04
600	150	196	1.00	1400	6.80	187.68	1.54	1.04	0.01	1386.68	1.54
600	300	352	1.00	2552	13.82	311.28	2.11	1.11	0.01	2504.28	2.11
600	450	482	1.12	3966	35.46	461.24	2.50	1.16	0.01	3875.40	7.26
600	600	446	1.21	4227	33.57	442.24	1.58	1.21	0.00	4055.44	11.57
700	175	242	1.00	1723	11.02	210.48	2.33	1.13	0.01	1684.48	2.33
700	350	496	1.00	3491	23.51	402.08	2.48	1.19	0.01	3394.08	2.48
700	525	571	1.18	4983	36.54	490.48	5.39	1.29	0.01	4762.24	16.04
700	700	399	1.43	5008	47.05	282.76	8.08	1.57	0.01	4486.56	21.27
800	200	249	1.16	2061	12.55	111.92	3.59	1.62	0.01	1918.92	3.59
800	400	356	1.34	3690	26.59	212.56	3.43	1.61	0.01	3539.56	3.43
800	600	266	1.63	4875	67.95	240.84	2.85	1.66	0.00	4704.36	12.93
800	800	302	1.62	5641	65.46	261.48	2.82	1.67	0.00	5399.92	7.42
900	225	325	1.00	2301	19.34	118.00	0.85	1.64	0.00	2089.00	0.85
900	450	377	1.39	4158	40.76	221.84	2.87	1.64	0.00	3995.84	2.87
900	675	614	1.32	6186	62.17	518.16	6.32	1.42	0.01	6019.64	8.04
900	900	753	1.16	6753	85.79	501.08	11.47	1.44	0.01	6280.96	16.70

The objective criteria are investigated further in Figures 2–4, using data with at most 2500 students. Larger instances are not considered since the IP model often fails to find solutions within the time limit. Figure 2 shows the difference in #1 preferences from $\mathcal{MM}$AS to the IP Model ($\frac{\mathcal{MM}AS-IP}{IP}$). The chart on the left splits the results by student numbers, whereas the figure on the right splits the results by the proportion of students to placements ([0.25, 0.5, 0.75, 1.0]). Firstly, regarding the number of students, we see that the differences increase with increasing problem size, i.e., $\mathcal{MM}$AS does not scale well. However, once the problem instances are very large (>2000 students) the advantage of the IP model is diminished. In fact, as it is seen in Table 5, if the problem instances are very large, $\mathcal{MM}$AS is by far the preferred method since on occasion the IP model struggles to even find solutions. On examining the chart to the right of Figure 2, we see that in the datasets with low proportions (0.25), the differences between $\mathcal{MM}$AS and the IP model are large. The difference reduces with increasing proportions (0.5 and 0.75) followed by a slight increase when the proportion is 1.0. This is not surprising since when the proportions of placements are low, the IP model is significantly more efficient and is able to find the optimal solutions.

Table 5. The results of the IP model and $\mathcal{MM}$AS on instances with varying number of students (250–900) and companies (200–700). Stu.: number of students; Comp.: number of companies; **#1 s**: number of times a rating of 1 was achieved; Avg.: average allocation.

		IP				$\mathcal{MM}$AS					
Stu.	**Comp.**	**#1sr**	**Avg.**	**Obj.**	**Time (s)**	**#1 s**	**SD**	**Avg.**	**SD**	**Obj.**	**SD**
1000	250	350	1.00	2485	20.69	167.88	2.42	1.52	0.01	2295.88	2.42
1000	500	659	1.05	4926	45.13	433.64	7.50	1.38	0.01	4693.64	7.50
1000	750	620	1.38	6823	76.93	430.20	6.14	1.57	0.01	6577.32	6.67
1000	1000	464	1.54	7165	114.53	314.24	8.65	1.69	0.01	6738.84	11.25
1200	300	479	1.00	3387	32.43	187.48	4.47	1.61	0.01	3090.48	4.47
1200	600	902	1.00	6372	63.72	575.88	7.24	1.36	0.01	6041.88	7.24
1200	900	852	1.29	8295	197.89	684.20	10.30	1.43	0.01	8070.72	13.05
1200	1200	1150	1.04	9134	169.86	840.80	10.83	1.30	0.01	8484.72	17.66
1400	350	481	1.00	3437	53.28	234.24	3.74	1.51	0.01	3185.24	3.74
1400	700	723	1.30	7025	107.78	417.48	4.40	1.60	0.00	6714.48	4.40
1400	1050	776	1.45	9475	139.75	583.60	10.23	1.58	0.01	9194.16	10.18
1400	1400	788	1.44	10,159	251.00	654.60	8.82	1.53	0.01	9611.12	12.17
1600	400	564	1.00	4014	72.83	252.48	2.42	1.55	0.00	3695.48	2.42
1600	800	768	1.33	7702	179.50	521.64	4.77	1.54	0.00	7450.64	4.77
1600	1200	906	1.43	10,932	191.22	707.00	6.37	1.55	0.00	10,408.52	11.53
1600	1600	902	1.44	11,617	322.23	698.36	7.17	1.56	0.00	10,917.16	11.05
1800	450	600	1.00	4297	91.67	284.40	2.37	1.53	0.00	3976.40	2.37
1800	900	949	1.27	8904	262.27	580.72	5.36	1.55	0.00	8530.72	5.36
1800	1350	1020	1.43	12,298	246.36	798.00	5.77	1.56	0.00	11,867.20	7.66
1800	1800	1276	1.29	13,295	772.45	987.04	11.07	1.45	0.01	12,411.96	15.25
2000	500	652	1.00	4639	112.73	529.84	4.22	1.19	0.01	4509.84	4.22
2000	1000	1395	1.00	9925	283.88	1135.68	8.93	1.19	0.01	9660.68	8.93
2000	1500	1916	1.04	14,266	335.39	1511.16	12.21	1.24	0.01	13,652.52	14.24
2000	2000	1948	1.03	15,351	549.03	1539.72	12.02	1.23	0.01	14,288.28	11.87
2500	625	857	1.00	6073	216.84	660.68	4.56	1.23	0.01	5869.68	4.56
2500	1250	1712	1.00	12,209	884.32	1286.00	6.95	1.25	0.00	11,776.00	6.95
2500	1875	2478	1.01	17,895	1402.82	1966.48	15.04	1.21	0.01	17,238.04	28.29
2500	2500	2500	1.00	19,292	1586.59	2036.44	15.70	1.19	0.01	17,993.36	17.09
3000	750	1005	1.00	7168	354.08	879.36	10.22	1.12	0.01	7035.36	10.22
3000	1500	2033	1.00	14,519	777.97	1698.75	8.13	1.16	0.00	14,177.75	8.13
3000	2250	-	-	-	1800.00	2370.88	15.28	1.21	0.01	20,819.96	16.30
3000	3000	3000	1.00	23,126	1361.27	2321.92	23.58	1.23	0.01	21,458.24	14.66
3500	875	1232	1.00	8766	1047.17	993.52	10.33	1.19	0.01	8522.52	10.33
3500	1750	-	-	-	1800.00	1948.61	7.61	1.19	0.00	16,594.61	7.61
3500	2625	-	-	-	1800.00	2662.12	11.65	1.24	0.00	24,192.00	14.27
3500	3500	-	-	-	1800.00	2743.84	13.72	1.22	0.00	25,022.63	16.98
4000	1000	1514	1.00	10,685	1893.59	1218.71	3.88	1.19	0.00	10,385.71	3.88
4000	2000	2784	1.00	19,755	2037.15	2210.61	6.59	1.21	0.00	19,176.61	6.59
4000	3000	-	-	-	1800.00	3179.00	17.73	1.20	0.00	27,805.08	12.89
4000	4000	-	-	-	1800.00	3324.33	22.52	1.17	0.01	28,407.67	10.16

Figure 3 shows a breakdown of the average allocations for the IP model by proportion of student to placements ([0.25, 0.5, 0.75, 1.00]). In low proportions of placements to students (0.25), we see that the average allocation is always close to 1.0, demonstrating that most of the placements were satisfied with #1 preferences, whereas, when the proportions become higher and there are more placements to fill, it is more difficult for a company to obtain its ideal preference/s. This is expected since if there are very few placements and many students (0.25), several first choice students can be allocated to each placement, whereas, at the other end of the scale with one placement per student (1.0), unless there are many (possibly unique) first choice students for every placement, an average allocation 1.0 can never be achieved.

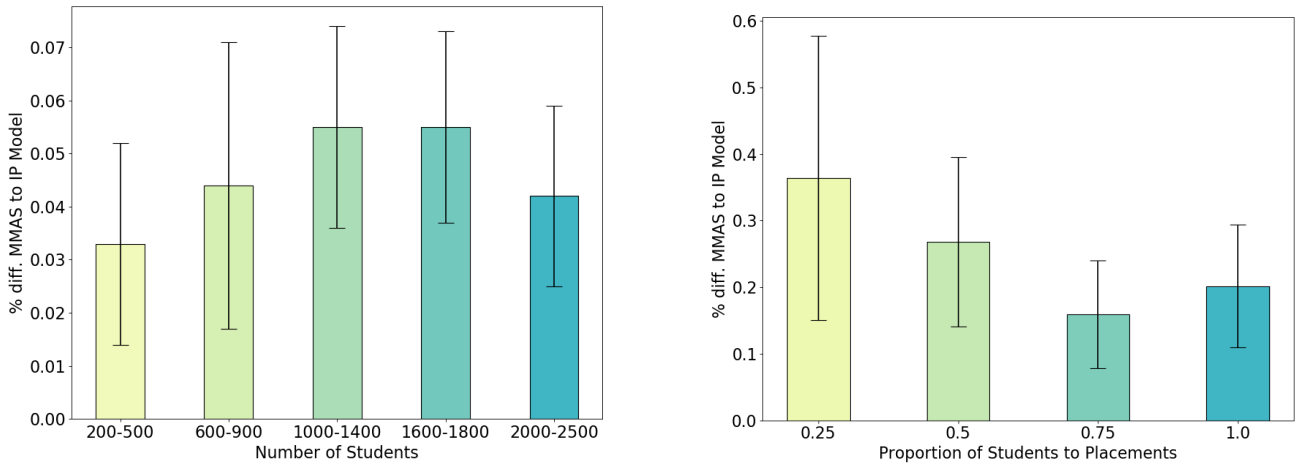

Figure 2. The % difference from $\mathcal{MM}$AS to the IP model with respect to #1 Prefs. On the left, results are presented as averages over number of students. For example, 600–900 is the average for the problem instances in the range of 600–900 students. On the right, the results are split by proportion of placements to students [0.25, 0.5, 0.75, 1.00].

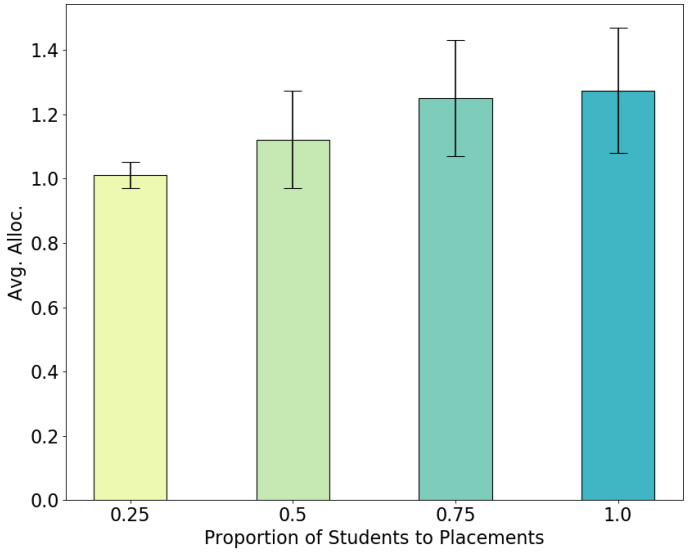

Figure 3. The average allocation for the IP split by proportion of placements to students [0.25, 0.5, 0.75, 1.00].

Figure 3 shows a breakdown of the average allocation for $\mathcal{MM}$AS by proportion of student to placements ([0.25, 0.5, 0.75, 1.00]). First, we note that the allocations are generally in a tight band, slowly increasing with increasing proportions. A comparison to the IP model shows several obvious differences. Overall, the average allocation is lower with low proportions of placements to students and increases slightly when the proportions are higher (0.75 and 1.0 compared to 0.25 and 0.5).

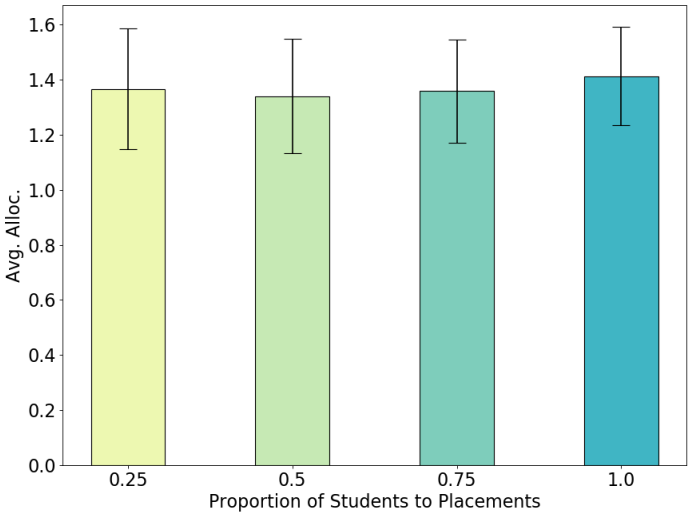

Figure 4. The average allocation for $\mathcal{MM}$AS split by proportion of placements to students [0.25, 0.5, 0.75, 1.00].

8. Conclusions and Future Work

In this study, we investigate the problem of allocating students to industry placements, which we refer to as the IBL allocation problem. This problem differs from those in previous studies by considering multiple requirements of the companies, such as company preferences for students, gender mix and sector constraints. We propose an exact approach to solve the problem via integer programming. We find that although the IP model is very effective on real data, it has limitations in the way it scales with large problem sizes. Hence, we also propose an ACO-based heuristic with the aim of finding good solutions in reasonable time-frames. Compared to manual allocations on real instances since 2012, we see that the IP model improves upon these allocations in several respects, including the average allocation preference and the number of placements allocated their *#1* preferences. This study demonstrates the value of automation in industry placement programs at universities by providing excellent solutions with a high degree of flexibility.

The real data show a trend of increasing numbers of students who would like to participate in IBL placements. Moreover, other university departments (e.g., nursing) typically require much larger numbers of placements (up to 4000 placements at some institutions). This provides the motivation to investigate how the proposed approaches deal with increasing scales. Hence, based on information from the real data (e.g., distribution of preferences), we generated a number of artificial problem instances. The IP solves small to medium problems very effectively, finding optimal solutions within 30 min, but struggles when the instances consist of more than 3000 students and may fail to find solutions for the largest instances. For these problem instances, we see a clear advantage provided by ACO, where it is able to find good solutions in small time-frames, although optimality cannot be proved.

Future Work

For the IBL placement allocation, the IP model is a viable option and scales well for the current program ($<$200 students per placement cycle). As such, it will be interesting to investigate whether or not the model can easily be modified to deal with industry placement programs at other universities. The authors of this study are currently investigating possibilities in this direction.

For very large similar allocation problems (such as allocating students to nursing placements), ACO is the preferable option, but with no guarantee of optimality. This leads to several potential research directions, where the IP and ACO approaches may be further enhanced to provide improved solutions in short time-frames. In particular, IP-based decompositions, such as Lagrangian relaxation [37,38], have proved to be effective in the past

and combinations of IPs and ACO have also been proven on a range of problems [39–42]. Moreover, large neighbourhood search approaches [43,44] may prove to be very effective in this context. Additionally, parallel implementations of ACO and associated hybrids [45,46] have proved to be effective in the past and such approaches could help to reduce run-times, enabling good solutions for large problems to be found in under a minute.

Author Contributions: Conceptualization, D.T.; Data curation, K.M. and S.B.; Methodology, D.T., K.M. and A.N.; Software, D.T.; Validation, A.N.; Writing – original draft, D.T., K.M. and S.B.; Writing—review & editing, S.B. and A.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This research was supported in part by the Monash eResearch Centre and eSolutions-Research Support Services through the use of the MonARCH HPC Cluster. Exemption from Ethical Review Project Number 12409, MUHREC, 15 March 2018. Deakin University Ethics Exemption: 2018-403: Work Integrated learning applications.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. An Illustration of the Problem: The IP Model

$$
\begin{aligned}
\text{Maximise } & \mathcal{L} - (x_{11} + 3x_{12} + 2x_{21} + x_{22} + 3x_{31} + 2x_{32}) - \\
& 3(y_{11} + y_{12} + y_{13}) + (y_{21} + y_{22} + y_{23}) - \\
& (|x_{11} + x_{21} - x_{31}| + |x_{12} + x_{22} - x_{32}|) \\
\text{Subject to} & \\
& x_{11} + x_{12} \leq 1 \\
& x_{21} + x_{22} \leq 1 \\
& x_{31} + x_{32} \leq 1 \\
& x_{11} + x_{21} + x_{31} \leq 2 \\
& x_{12} + x_{22} + x_{32} \leq 1 \\
& x_{11} = 0 \\
& x_{11} + x_{21} + x_{31} - (y_{11} + y_{12} + y_{13}) = 0 \\
& x_{12} + x_{22} + x_{32} - (y_{21} + y_{22} + y_{23}) = 0 \\
& y_{12} - y_{11} \leq 0 \\
& y_{13} - y_{12} \leq 0 \\
& y_{22} - y_{21} \leq 0 \\
& y_{23} - y_{22} \leq 0 \\
& x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, y_{11}, y_{12}, y_{13}, y_{21}, y_{22}, y_{23} \in \{0, 1\}
\end{aligned}
\tag{A1}
$$

References

1. Patrick, C.; Peach, D.; Pocknee, C. The WIL (Work Integrated Learning) Report: A National Scoping Study (Australian Learning and Teaching Council (ALTC) Final Report). 2008. Available online: www.altc.edu.auandwww.acen.edu.au (accessed on 21 July 2021).
2. Sun, Z. Matchings with Constraints. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), Macao, China, 10–16 August 2019; pp. 6464–6465.
3. Doherty, C.L.; Fogg, L.; Bigley, M.B.; Todd, B.; O'Sullivan, A.L. Nurse practitioner student clinical placement processes: A national survey of nurse practitioner programs. *Nurs. Outlook* **2019**, *68*, 55–61. [CrossRef] [PubMed]
4. Eltorai, A.; Daniels, A. National Medical School Matching Program: Optimizing outcomes. *Adv. Med Educ. Pract.* **2016**, *7*, 371–373. [CrossRef]

5. Nemhauser, G.L.; Wolsey, L.A. *Integer and Combinatorial Optimization*; Wiley-Interscience: New York, NY, USA, 1988.
6. Manlove, D. *Algorithmics of Matching under Preferences*; World Scientific: Singapore, 2013.
7. Papadimitriou, C.H.; Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity*; Courier Corporation: Elizabeth, NJ, USA, 1998.
8. Blum, C.; Roli, A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Comput. Surv.* **2003**, *35*, 268–308. [CrossRef]
9. Dorigo, M.; Stűtzle, T. *Ant Colony Optimization*; MIT Press: Cambridge, MA, USA, 2004.
10. Hall, P. On Representatives of Subsets. *J. Lond. Math. Soc.* **1935**, *10*, 26–30. [CrossRef]
11. Gale, D.; Shapley, L.S. College Admissions and the Stability of Marriage. *Am. Math. Mon.* **1962**, *69*, 9–15. [CrossRef]
12. Dov, A.G.B. Optimal Assignment of Research and Development Projects in a Large Company Using an Integer Programming Model. *IEEE Trans. Eng. Manag.* **1965**, *EM-12*, 138–142. [CrossRef]
13. Gusfield, D.; Irving, R. *The Stable Marriage Problem: Structure and Algorithms*; MIT Press: Cambridge, MA, USA, 1989.
14. Irving, R.W. An efficient algorithm for the stable roommates problem. *J. Algorithms* **1985**, *6*, 577–595. [CrossRef]
15. Manlove, D.; Irving, R.; Iwama, K.; Miyazaki, S.; Morita, Y. Hard variants of stable marriage. *Theor. Comput. Sci.* **2002**, *276*, 261–279. [CrossRef]
16. Irving, R.W. Stable marriage and indifference. *Discret. Appl. Math* **1994**, *48*, 261–272. [CrossRef]
17. Irving, R.W.; Manlove, D.F. The Stable Roommates Problem with Ties. *J. Algorithms* **2002**, *43*, 35–105. [CrossRef]
18. Klaus, B.; Manlove, D.F.; Rossi, F. *Matching Under Preferences*; Cambridge University Press: Cambridge, UK, 2016.
19. Cholette, S. A Novel Problem for a Vintage Technique: Using Mixed-Integer Programming to Match Wineries and Distributors. *Interfaces* **2007**, *37*, 231–239. [CrossRef]
20. Al-Yakoob, S.M.; Sherali, H.D. Mixed-integer Programming Models for an Employee Scheduling Problem with Multiple Shifts and Work Locations. *Ann. Oper. Res.* **2007**, *155*, 119–142. [CrossRef]
21. Naik, V.K.; Liu, C.; Yang, L.; Wagner, J. Online Resource Matching for Heterogeneous Grid Environments. *IEEE Int. Symp. Clust. Comput. Grid* **2005**, *2*, 607–614.
22. Magnanti, T.L.; Natarajan, K. Allocating Students to Multidisciplinary Capstone Projects Using Discrete Optimization. *INFORMS J. Appl. Anal.* **2018**, *48*, 204–216. [CrossRef]
23. Harper, P.R.; de Senna, V.; Vieira, I.T.; Shahani, A.K. A Genetic Algorithm for the Project Assignment Problem. *Comput. Oper. Res.* **2005**, *32*, 1255–1265. [CrossRef]
24. Gupta, M.; Rees, J.; Chaturvedi, A.; Chi, J. Matching Information Security Vulnerabilities to Organizational Security Profiles: A Genetic Algorithm Approach. *Decis. Support Syst.* **2006**, *41*, 592–603. [CrossRef]
25. Omkumar, M.; Shahabudeen, P. Ant Colony Optimization for Multilevel Assembly Job Shop Scheduling. *Int. J. Appl. Manag. Technol.* **2008**, *6*, 127–152.
26. D'Acierno, L.; Gallo, M.; Montella, B. An Ant Colony Optimisation Algorithm for Solving the Asymmetric Traffic Assignment Problem. *Eur. J. Oper. Res.* **2012**, *217*, 459–469. [CrossRef]
27. Huan, Y. Image Edge Detection Based on Ant Colony Optimization Algorithm. *Int. J. Adv. Pervasive Ubiquitous Comput.* **2016**, *8*, 1–12. [CrossRef]
28. Thiruvady, D.; Elmi, A.; Nazari, A.; Schneider, J.G. Minimising Cycle Time in Assembly Lines: A Novel Ant Colony Optimisation Approach. In *Australasian Joint Conference on Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 125–137.
29. Thiruvady, D.; Nazari, A.; Elmi, A. An Ant Colony Optimisation Based Heuristic for Mixed-model Assembly Line Balancing with Setups. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–8.
30. Pendharkar, P.C. An Ant Colony Optimization Heuristic for Constrained Task Allocation Problem. *J. Comput. Sci.* **2015**, *7*, 37–47. [CrossRef]
31. Shyu, S.J.; Lin, B.; Hsiao, T.S. Ant Colony Optimization for the Cell Assignment Problem in PCS Networks. *Comput. Oper. Res.* **2006**, *33*, 1713–1740. [CrossRef]
32. Chaharsooghi, S.K.; Kermani, A.H.M. An effective ant colony optimization algorithm (ACO) for multi-objective resource allocation problem (MORAP). *Appl. Math. Comput.* **2008**, *200*, 167–177. [CrossRef]
33. Thiruvady, D.; Nazari, A.; Aleti, A. Multi-objective Beam-ACO for Maximising Reliability and Minimising Communication Overhead in the Component Deployment Problem. *Algorithms* **2020**, *13*, 252. [CrossRef]
34. Stützle, T.; Hoos, H.H. MAX–MIN ant system. *Future Gener. Comput. Syst.* **2000**, *16*, 889–914. [CrossRef]
35. Dorigo, M.; Gambardella, L.M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Trans. Evol. Comput.* **1997**, *1*, 53–66. [CrossRef]
36. Gurobi Optimization. Gurobi Optimizer Reference Manual. 2021. Available online: https://www.gurobi.com/ (accessed on 21 July 2021).
37. Fisher, M. The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Manag. Sci.* **2004**, *50*, 1861–1871. [CrossRef]
38. Wolsey, L.A. *Integer Programming*; Wiley-Interscience: New York, NY, USA, 1998.
39. Thiruvady, D.; Singh, G.; Ernst, A.T. Hybrids of Integer Programming and ACO for Resource Constrained Job Scheduling. In *Hybrid Metaheuristics*; Blesa, M.J., Blum, C., Voß, S., Eds.; Springer International Publishing: Cham, Switzerland, 2014; Volume 8457, pp. 130–144.

40. Thiruvady, D.; Wallace, M.; Gu, H.; Schutt, A. A Lagrangian Relaxation and ACO Hybrid for Resource Constrained Project Scheduling with Discounted Cash Flows. *J. Heuristics* **2014**, *20*, 643–676. [CrossRef]
41. Thiruvady, D.; Ernst, A.T.; Wallace, M. A Lagrangian-ACO Matheuristic for Car Sequencing. *EURO J. Comput. Optim.* **2014**, *2*, 279–296. [CrossRef]
42. Thiruvady, D.; Blum, C.; Ernst, A.T. Solution Merging in Matheuristics for Resource Constrained Job Scheduling. *Algorithms* **2020**, *13*, 256. [CrossRef]
43. Perron, L.; Shaw, P.; Furnon, V. Propagation Guided Large Neighborhood Search. In *Principles and Practice of Constraint Programming—CP 2004*; Wallace, M., Ed.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 468–481.
44. Thiruvady, D.; Morgan, K.; Amir, A.; Ernst, A.T. Large Neighbourhood Search based on Mixed Integer Programming and Ant Colony Optimisation for Car Sequencing. *Int. J. Prod. Res.* **2020**, *58*, 2696–2711. [CrossRef]
45. Brent, O.; Thiruvady, D.; Gómez-Iglesias, A.; Garcia-Flores, R. A Parallel Lagrangian-ACO Heuristic for Project Scheduling. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 2985–2991.
46. Cohen, D.; Gómez-Iglesias, A.; Thiruvady, D.; Ernst, A.T., Resource Constrained Job Scheduling with Parallel Constraint-Based ACO. In Proceedings of the Artificial Life and Computational Intelligence: Third Australasian Conference, ACALCI 2017, Geelong, VIC, Australia, 31 January–2 February 2017; Wagner, M., Li, X., Hendtlass, T., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 266–278.