University Staff Teaching Allocation: Formulating and Optimising a Many-Objective Problem

Jonathan E. Fieldsend
University of Exeter
Exeter, UK EX4 4QF
J.E.Fieldsend@exeter.ac.uk

ABSTRACT

The allocation of university staff to teaching exhibits a range of often competing objectives. We illustrate the use of an augmented version of NSGA-III to undertake the seven-objective optimisation of this problem, to find a trade-off front for a university department using real world data. We highlight its use in decision-making, and compare solutions identified to an actual allocation made prior to the availability of the optimisation tool. The criteria we consider include minimising the imbalance in workload distribution among staff; minimising the average load; minimising the maximum peak load; minimising the staff per module; minimising staff dissatisfaction with teaching allocations; and minimising the variation from the previous year's allocation (allocation churn). We derive mathematical forms for these various criteria, and show we can determine the maximum possible values for all criteria and the minimum values for most exactly (with lower bounds on the remaining criteria). For many of the objectives, when considered in isolation, an optimal solution may be obtained rapidly. We demonstrate the advantage of utilising such extreme solutions to drastically improve the optimisation efficiency in this many-objective optimisation problem. We also identify issues that NSGA-III can experience due to selection between generations.

CCS CONCEPTS

•Computing methodologies → Genetic algorithms; •Applied computing → Multi-criterion optimization and decision-making; •Social and professional topics → *Computational science and engineering education;*

KEYWORDS

Many-objective optimisation; evolutionary optimisation; robust optimisation; resource allocation.

ACM Reference format:
Jonathan E. Fieldsend. 2017. University Staff Teaching Allocation: Formulating and Optimising a Many-Objective Problem. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071230

978-1-4503-4920-8/17/07...$15.00
DOI: http://dx.doi.org/10.1145/3071178.3071230

1 INTRODUCTION

Optimisation approaches have been exploited for a number of years in the university sector for timetabling exams and other module components[1], e.g. lectures, workshops, etc. [3]. With the growth in the higher education sector in many countries, efficient use of human resources (as opposed to just capital resources) has become imperative — due to, for instance, the lag time on new buildings increasing capacity. The optimal allocation of staff to teaching activities has however been far less explored than that of timetabling in the university sector. Staff allocation nevertheless is an important topic. Staff who are regularly overloaded, degrading their work/life balance, will be more likely to suffer from stress. This will affect the quality of the teaching they deliver, their general health, and the likelihood of them seeking employment in another institution or outside the sector.

Academic staff have a number of activities they are required or expected to undertake, depending on their particular role. Tasks required of staff can include running grants and supervising postdoctoral researchers; writing research papers; supervising/mentoring research students; writing grants; professional activity (reviewing, conference organisation, conference presentation, etc.); contract research for companies, charities or the public sector; scholarship/personal development training; administrative roles (Head of Department, Director of Education, Admissions Tutor, Programme Lead, etc.); general administrative duties (departmental meetings, open days, outreach activities); student project supervision; tutorials; teaching — including lectures, seminars, workshops, project supervision on taught programmes, marking, preparation, etc.

Many workload items can be 'locked in' for significant periods of time (e.g. grants often run over multiple years), and the non-teaching load for an academic can be substantial, especially at research intensive universities. Some items can be redistributed on occasion with sufficient planning (e.g. administrative roles).

In many universities there exist the departmental role of Director of Education (or similarly titled), part of whose job is to help the Head of Department by proposing and advising on the allocation of teaching responsibilities to staff, in order as much as possible to ensure fairness regarding the total load each staff member experiences (given their contractual hours). At the author's institution for instance, the Head of Department needs to agree with staff when their total workload exceeds ±10% of their nominal contract.

A parallel concern is the maximisation of student satisfaction with their experience on the modules they are taught. Allocation of teaching needs to be done sufficiently early to allow staff time to

[1] In many universities a degree programme comprises multiple different modules — these may alternatively be referred to as courses.

prepare, especially if their allocated teaching has changed from the previous year.

Often multiple different criteria have been considered for other resource allocation problems, e.g. nurse scheduling. An overview of some of the widely used criteria in human resource allocation is provided in [2].

2 COMPETING CRITERIA WHEN ALLOCATING TEACHING LOAD TO STAFF

A natural objective when deciding upon a teaching allocation is to ensure no member of staff is significantly more overburdened with workload compared to others (fair distribution). I.e. where w_j is the assigned workload hours (including teaching and non-teaching components) of the jth member of staff and h_j is their contractual hours, the problem is

$$\min \max_j \left(\frac{w_j}{h_j} \right). \tag{1}$$

At the same time the average total workload should be minimised, i.e.

$$\min \frac{\sum_j w_j}{\sum_j h_j}. \tag{2}$$

Depending on the processes in the institution, the two objectives can actually end up being in competition, as (2) may vary with the teaching allocation. This is because staff who have not taught a module previously are typically provided additional 'first-time' load due to the extra preparation needed to develop module materials from scratch, as opposed to the yearly reinvigoration of materials required for a module previously taught. Relieving one or more overburdened staff to optimise (1) may increase the total teaching hours (and therefore the average workload) distributed across all staff.[2] There are however further criteria to consider.

Staff may have preferences to teach (or not teach) certain modules, this also needs to be considered. Staff made to teach on modules they do not want to (e.g., because they do not have the necessary background to teach it confidently) may result in module delivery that is less well-received by students, thus affecting their satisfaction and general educational experience, as well as potentially inducing stress in the staff concerned.

Historical data at the author's department suggests that modules with greater numbers of staff tend to perform worse on average (in terms of student satisfaction) than those with fewer members of staff. This may be due, for instance, to increased difficulties in projecting a coherent module across multiple staff. It is generally preferable to minimise 'peak load' for staff to prevent burnout and potentially delays in marking turnaround, etc. This means trying to distribute teaching relatively evenly across terms. Finally, change itself is disruptive — so the last criterion is to minimise variation from the previous year's teaching allocation.

Given the above, the following (potentially competing) criteria would appear appropriate to consider when allocating staff to deliver modules:

i Minimise the average workload across staff

ii Minimise the imbalance of workload among staff, as a proportion of their contracted hours

iii Minimise the maximum staff dissatisfaction with teaching allocation

iv Minimise the average number of staff teaching on a module

v Minimising the maximum peak load for staff

vi Minimise variation from the previous year's teaching allocation

We now formally describe the problem. Let us have n staff and m modules, with each staff member having the (pre-allocated) non-teaching workload of w_j^*. Each module has three distinct chunks of time associated with it: c_i, the number of module leader hours required to coordinate the ith module – this is indivisible and assigned to a single member of staff; d_i, the time spent delivering lectures, seminars, surgeries, workshops and marking by staff involved with the module; and p_i, the time spent preparing the various materials required for the module by staff. If staff are allocated to a module they have not previously taught on at any point, there is often additional load for preparation — at the author's institution this is subject to a multiplication factor of 2.

To undertake an evaluation of a teaching allocation, we require the following. The n-dimensional vector $\mathbf{w}^*$ containing the non-teaching workloads of all staff who may be assigned teaching; the n-dimensional vector $\mathbf{h}$ containing the contractual hours of each staff member; the m-dimensional vectors $\mathbf{c}$, $\mathbf{d}$ and $\mathbf{p}$ containing the time costs associated with each module; a $m \times n$ matrix T, whose element T_{ij} indicates if the jth staff member has taught the ith module previously (represented respectively by a 1 if they haven't and a 0 if they have); a $m \times n$ matrix R whose element R_{ij} indicates the proportion of the ith module the jth staff member taught in the year immediately before the current year being optimised; a $m \times n$ matrix of teaching preferences P. Here an element $P_{ij} = 0$ indicates the jth staff member prefers to teach module i, $P_{ij} = 1$ indicates the jth staff member could teach module i, but would prefer not to and $P_{ij} = 2$ indicates that the jth staff feels they do not have the background necessary to deliver module i. (Note, further granularity on preferences could be provided with additional categories, though eliciting a consistent continuous value from staff to represent their preference is likely to be problematic.) Where particular teaching *must* be allocated to particular staff, this can be represented in $\mathbf{w}^*$, with the corresponding reduction in $\mathbf{c}$, $\mathbf{d}$ and $\mathbf{p}$ prior to optimisation, or alternatively a mask used to fix these items in teaching allocations (which would be preferable in terms of some of the criteria calculations, e.g. peak load).

Beyond the integer value constraints mentioned above, additional constraints on valid problem definition vectors and matrices are as follows:

$$w_j^* \geq 0, \quad \forall j, \tag{3}$$

as staff should not have negative non-teaching workload,

$$h_j > 0, \quad \forall j, \tag{4}$$

all staff being considered have contractual hours,

$$c_i > 0, \quad \forall i, \tag{5}$$

[2]Note, we do not use $\min \sum_j \frac{w_j}{h_j}$ for (2), as the optimum for this is the perverse result that all teaching is allocated to the staff with the largest contractual hours and none to staff with fewer hours.

all modules should have a coordinator load,

$$p_i > 0, \quad \forall i, \tag{6}$$

all modules should have preparation load, and

$$d_i > 0, \quad \forall i, \tag{7}$$

all modules should have delivery/marking load. Additionally

$$0 \leq \sum_{j=1}^{n} R_{ij} \leq 1, \quad \forall i, \tag{8}$$

as no module should have had a larger than 100% provision in the previous year. However, it may be less than 100% as represented in R, due to e.g. staff leaving (meaning they do not appear in R) or there being brand new modules.

A teaching allocation solution is represented as two $n \times m$ matrices C and X. C_{ij} denotes if jth member of staff coordinates the ith module, therefore the matrix has the following constraints:

$$\sum_{j=1}^{n} C_{ij} = 1, \quad \forall i, \tag{9}$$

as no module can have more than one leader, and

$$C_{ij} \in \{0, 1\}, \quad \forall i, j. \tag{10}$$

X is matrix of teaching proportions, and has the following constraints:

$$0 \leq X_{ij} \leq 1, \tag{11}$$

as staff members can deliver no more that 100% of a module (even if it might *feel* otherwise when the module is running!), and no staff member can deliver a negative proportion of a module. Additionally

$$\sum_{j=1}^{n} X_{ij} = 1, \quad \forall i, \tag{12}$$

as the total allocation across staff to a module must be 100% (all modules should be appropriately resourced in a legal allocation).

Usually one would not have a module leader who does not also teach on it, so the search space may be further constrained with

$$\left(\delta(C_{ij} = 1) + \delta(X_{ij} = 0)\right) \leq 1 \quad \forall i, j, \tag{13}$$

where δ is the Kronecker delta (identity function).

Now we have described the necessary variables and constraints, we can formally express the competing criteria we are concerned with optimising, and highlight where we can quickly compute allocations which return the minimum/maximum value for each criterion given the system constraints. This is useful, as it allows us to judge trade-off solutions in light of their relative performance against these bounds. It also can provide good initial solutions to consider in an optimisation as it allows us to effectively 'pin down' many of the corners of the Pareto front.

2.1 Discrete nature of problem

Although we have discussed the elements of X thus far in terms of real values (representing the proportion of a module), from a practical standpoint a module's teaching load cannot be split into arbitrary proportions. Realistically, teaching is partitioned into substantial chunks, usually no smaller than 5% or 10%. We can therefore view X as containing values which increment by 0.05. For optimisation purposes we will use integers (and divide through when quality calculations are required), making the optimisation problem discrete.

2.2 Total workload

Minimising the average staff workload is equivalent to minimising the total workload across staff. This can be calculated for any given solution $\mathbf{s} = \{C, X\}$ through the follow cost function:

$$f_1(\mathbf{s}) = \frac{\sum_{j=1}^{n} \left(w_j^* + \sum_{i=1}^{m} \left(c_i C_{ij} + (d_i + (1 + \alpha T_{ij})p_i)X_{ij}\right)\right)}{\sum_{j=1}^{n} h_j}. \tag{14}$$

Due to the constraints, the minimum and maximum values possible for (14) are entirely determined by the component multiplied by α. If all modules have been taught at least once before by available staff, then it is possible to allocate teaching such that no module is delivered for the first time, ensuring all αT_{ij} elements in the calculation are zero. The minimum total workload is thus $(\sum_j w_j^* + \sum_i (c_i + d_i + p_i))/\sum_j h_j$. Correspondingly, if for each module there is at least one staff member that has not taught it the maximum is $(\sum_j w_j^* + \sum_i (c_i + d_i + \alpha p_i))/\sum_j h_j$ (corresponding to the allocation of all staff to modules they have not previously taught). In the case where some modules have never been taught before by any staff available the lower bound is increase by $\alpha p_i/\sum_j h_j$ for each of these particular modules, and likewise for modules which have been taught by all staff previously the upper bound decreases by $\alpha p_i/\sum_j h_j$ for each.

2.3 Balanced workload

Maximum individual workload, as a proportion of contracted hours can be calculated as:

$$g(\mathbf{s}) = \max_j \left(\frac{1}{h_j}\left(w_j^* + \sum_{i=1}^{m} \left(c_i C_{ij} + (d_i + (1 + \alpha T_{ij})p_i)X_{ij}\right)\right)\right). \tag{15}$$

However, this criteria doesn't adequately differentiate solutions in situations where one (or more) staff have a high w_j^*, as this can determine the maximum (without any teaching being allocated to them) thus unbalanced and balanced workload distributions are assigned the same quality value. Instead we use the balanced assignment problem (see e.g. [2]), which aims to minimise the spread of workloads:

$$f_2(\mathbf{s}) = \max_j \left(\frac{1}{h_j}\left(w_j^* + \sum_{i=1}^{m} \left(c_i C_{ij} + (d_i + (1 + \alpha T_{ij})p_i)X_{ij}\right)\right)\right) - \min_j \left(\frac{1}{h_j}\left(w_j^* + \sum_{i=1}^{m} \left(c_i C_{ij} + (d_i + (1 + \alpha T_{ij})p_i)X_{ij}\right)\right)\right). \tag{16}$$

Determining the lower and upper bounds for (16) is more problematic than for (14). Although determining the upper bound may seem trivial — by e.g. allocating all teaching to the staff member with the highest w_j^*, this does not necessarily result in the maximum, as a staff member with a lower w_j^* may have taught fewer of the modules previously, and therefore experience a higher preparation cost for the same teaching allocation (and a higher total imbalance). As such the upper limit requires in general n evaluations of (16). Allocating all teaching to each staff member in turn, to determine which results in the highest overall individual workload. Unfortunately the minimum value for (16), and the corresponding $\mathbf{s}$, cannot be determined *a priori*, and requires an active search. However a lower bound can be determined. This is

$$\max\left(0, \max_j (w_j^*/h_j) - \frac{\sum w_j^* + \sum (c_i + d_i + (1+\alpha)p_i)}{\sum h_j}\right), \quad (17)$$

which captures the degree to which it is possible to allocate teaching to result in a perfectly balanced workload, if teaching hours could be distributed with arbitrary precision.

2.4 Staff dissatisfaction

We could calculate maximum individual staff dissatisfaction as $\max_i \sum_j X_{ij} P_{ij}$. This effectively makes a linear sum of allocations weighted by the preferences. However, deciding how much worse it is to have teaching assigned from one category than another is difficult (and is likely to vary *between* individuals). Instead we pull these apart and minimise the maximum number in each category and above that an individual must deliver (excluding from consideration the first 'preferred' category) – scaled by the amount of the module being delivered (as it is worse to have to teach all of a module that you would prefer not to, rather than just a part of it). These cost functions are therefore

$$f_3(\mathbf{s}) = \max_i \sum_j X_{ij} \delta(P_{ij} \geq 1), \quad (18)$$

$$f_4(\mathbf{s}) = \max_i \sum_j X_{ij} \delta(P_{ij} = 2). \quad (19)$$

Where v is the number of rows in P which *only* have entries equal to 1 or 2, the minimum value for (18) is $\lceil \frac{vk}{n} \rceil / k$, where k is maximum number of indivisible parts a module can be broken down into (e.g., if the smallest chunk is 5%, $k = 20$). This corresponds to distributing all modules which are not preferred by any staff equally as possible across all staff. The maximum value for (18) is the number of rows where *any* element is 1 or 2. The bounds for (19) may be determined in a similar fashion, with reference solely to elements with value 2.

2.5 Average number of staff

Minimising the average number of staff teaching on each module may be calculated by counting the non-zero elements of X:[3]

$$f_5(\mathbf{s}) = \frac{\sum_i \sum_j \delta(X_{ij} \neq 0)}{m}. \quad (20)$$

Then minimum value this can take is 1 — which corresponds to all staff being allocated solely to deliver $\lceil n/m \rceil$ or $\lfloor n/m \rfloor$ modules. The maximum value it can take is $\min(m, k)$.

2.6 Peak load

The peak load is indirectly optimised by minimising the maximum imbalance in teaching allocation across terms. For this we need an additional m-dimensional vector, $\mathbf{t}$, which holds the delivery term for each module. At the University of Exeter there are two teaching terms and a shorter examination term. Here we use $t_i = 1$ to indicate the ith module is delivered only in term 1, $t_i = 2$ to indicate it is delivered only in term 2, and $t_i = 0$ to indicate it is delivered across terms. The final criteria to minimise is thus

$$f_6(\mathbf{s}) = \max_j \left(\left| \frac{1}{h_i} (\sum_i (C_{ij} c_i + X_{ij}(d_i + p_i + \alpha p_i T_{ij})) \delta(t_i = 1) - \sum_i (C_{ij} c_i + X_{ij}(d_i + p_i + \alpha p_i T_{ij})) \delta(t_i = 2)) \right| \right). \quad (21)$$

The maximum value this criterion can take can be computed in $2n$ calculations, by calculating the effect of allocating each staff member in turn all the term 1 teaching or all the term 2 teaching, and computing the resultant unbalanced teaching load. If the elements of X were not limited in resolution, then the minimum would be 0. However, in the case where the elements of X can only vary in fixed increments, we cannot determine the optimal value *a priori* (though the lower bound is 0).

2.7 Variation from previous year

An allocation which minimises the average load does not guarantee by itself that the allocation is the minimum change from the previous year. This is because a number of staff may have experience teaching on a module previously, but one or more of them might not have taught it in the *immediately* previous year, furthermore the distribution of teaching *within* a module may also have changed.

This criterion aim to minimises the "churn" of staff on a module between years. Given R, the allocations of *current* staff to modules in the previous year, we may calculate

$$f_7(\mathbf{s}) = \sum_i \sum_j |X_{ij} - R_{ij}|, \quad (22)$$

which measures the difference in module teaching allocation. Note, the construction of R needs to deal with situations where new modules have been introduced (in which case a corresponding row of zeros will be in R), where a module is discontinued (in which case it should be omitted from R). Also, when staff leave/retire columns will be omitted from R, and new staff will have columns of 0, to ensure both matrices are the same size.

The minimum of f_7 can be obtained by ensuring the non-zero elements of R are duplicated in X, the maximum when the non-zero elements of R are zeros in X for all locations.[4]

2.8 Criteria summary

Table 1 summarises the properties of the various criteria we are interested in here. As discussed above, for many of the individual criteria we can rapidly determine an optimal allocation, however for (16) and (21) this is not possible. They may be cast as integer linear programming problems, however this problem type is NP-complete ([10], page 3). One approach in such a situation is to relax the integer constraint on the elements of X, solve the corresponding linear programming problem, and then convert the real-valued elements of solutions to their nearest corresponding integer. However

[3] If you would want to assign module lead to a staff member who does not otherwise deliver on the module, (20) should be replaced with $\sum_i \sum_j \delta((C_{ij} + X_{ij}) \neq 0)$.

[4] If this is not possible, for instance due to a module being delivered previously by all current staff, that the maximum is ensured by allocating all the load for that module to the least loaded staff member on it the previous year.

Table 1: Table of criteria properties

	Can rapidly calculate [evaluations of $f_i(\mathbf{s})$ required]:			
#	min	Lower Bound	max	Minimising **s**
f_1	✓ [1]	✓ [1]	✓ [1]	✓ [1]
f_2	✗ [−]	✓ [n]	✓ [n]	✗ [−]
f_3	✓ [1]	✓ [1]	✓ [1]	✓ [1]
f_4	✓ [1]	✓ [1]	✓ [1]	✓ [1]
f_5	✓ [1]	✓ [1]	✓ [1]	✓ [1]
f_6	✗ [−]	✓ [1]	✓ [$2n$]	✗ [−]
f_7	✓ [1]	✓ [1]	✓ [1]	✓ [1]

solutions using such an approach may not be optimal, or indeed even feasible — in this problem domain the latter is especially likely given constraints (9) and (12). Instead, for both (16) and (21) we exploit a rapid downhill search to a local optimum. The point based local optimisers proceed as follows, given an initial solution:

Balanced load.

1. Identify the two staff allocated teaching with the highest ($\mathbf{u}$) and lowest ($\mathbf{v}$) proportionate loads overall.
2. Transfer a unit of teaching load from $\mathbf{u}$ to $\mathbf{v}$. If $\mathbf{u}$ is the coordinator for the chosen module, pass this responsibility too with a 50% probability (100% if the reduction in teaching means $\mathbf{u}$ will no longer teach on the module).
3. Return to 1, unless stopping criteria are met.

Peak load.

1. Identify the staff allocated teaching with the highest ($\mathbf{u}$) and lowest ($\mathbf{v}$) load differential between their teaching terms.
2. Transfer a unit of teaching load from $\mathbf{u}$, from their most intense term to $\mathbf{v}$. If $\mathbf{u}$ is the coordinator for the chosen module, pass this responsibility too with a 50% probability (100% if the reduction in teaching means $\mathbf{u}$ will no longer teach on the module).
3. Return to 1, unless stopping criteria are met.

3 MANY-OBJECTIVE OPTIMISATION

Without loss of generality, when optimising a multi-objective problem we seek to simultaneously minimise D objectives: $f_d(\mathbf{s})$, $d = 1, \ldots, D$ where each objective depends upon design solutions $\mathbf{s}$ which are made up of K parameters or decision variables. These parameters may also be subject to equality and inequality constraints, which together define $\mathcal{S}$, the feasible search space. Related to this is $\mathcal{Y}$, the objective space image of $\mathcal{S}$ (also referred to as the feasible objective space). When faced with only a single objective an optimal solution $\mathbf{s}^*$ is one which minimises the objective, subject to $\mathbf{s}^* \in \mathcal{S}$. However, when there is more than one objective to be minimised, solutions may exist for which performance on one objective cannot be improved without reducing performance on at least one other. Such solutions are said to be *Pareto optimal*. The set of all Pareto optimal solutions is said to form the *Pareto set*, whose image in objective space is known as the *Pareto front*.

A solution $\mathbf{s}$ is said to *dominate* another $\mathbf{s}'$ iff

$$f_d(\mathbf{s}) \leq f_d(\mathbf{s}') \quad \forall d = 1, \ldots, D \quad \text{and} \quad \mathbf{f}(\mathbf{s}) \neq \mathbf{f}(\mathbf{s}') \qquad (23)$$

This is often simply denoted as $\mathbf{s} \prec \mathbf{s}'$.

Many-objective evolutionary algorithms (MaOEAs), evolutionary optimisers designed specifically for four or more competing criteria, typically rely on comparison measures in addition to, or as a replacement for, Pareto dominance to order putative solutions. This is due to dominance providing a lesser discriminatory ability as D increases [4]. Decomposition-based MaOEA approaches have proved successful and popular in the last few years since the early work in their development [9] and we employ the recent NSGA-III algorithm [6] for our problem here. NSGA-III attempts to optimise multiple scalarisations of the problem, which are optimised in parallel by a single search population.

4 OPERATOR DEFINITIONS

Due to the nature of our representation and constraints, we do not employ an 'off-the-shelf' implementation of NSGA-III, as we need to incorporate domain-specific evolutionary operators into the optimiser. Here we define these operators, which ensure domain constraints are always adhered to.

Initialisation: In the initialisation of a random solution, each module is processed in turn. The teaching for a module is incrementally allocated, 5% at a time, to a randomly selected member of staff. At the end of teaching assignment for a module, the staff member allocated the most teaching is assigned the corresponding coordinator role.

Recombination: A form of uniform crossover is employed, which is at the module level (thereby ensuring the legality of children). On the selection of two parents, there is a 80% probability the parents will be crossed over (otherwise they are copied directly for subsequent mutation). On crossover each overall module allocation has a 50% probability of being swapped between parents in the resulting children.

Mutation: When a solution is mutated a module is selected at random, and then a staff member currently teaching on that module is selected at random. With equal probability, either the teaching allotted to that member of staff is given to a different random member of staff, or just 5% is moved from the selected member of staff to a random other staff member.

5 EXPERIMENTS

We conduct experiments using real workload data for the Computer Science group at the University of Exeter for the academic year 2016-17. Solutions have to allocate the teaching of 42 modules across 16 staff, making 672 binary decision variables and 672 discrete decision variables that can take values in the range [0,20].

5.1 Initialisation and preservation of extremes

Often a MaOEA is initialised with a population of well-distributed random solutions (in design space). For this problem we can rapidly compute solutions which will effectively 'pin' the Pareto front by minimising each of the individual criteria either directly (see Table 1 for details), or for a couple of criteria using rapid downhill search (see Section 2.8, we use 50 downhill evaluations here) and can augment an initial random population with these solutions.[5] This

[5]It should be noted however that a solution that minimises a single criterion isn't guaranteed to be Pareto optimal, as the landscape on this criterion may have multiple optimal, which may differ in performance on the other ignored criteria.

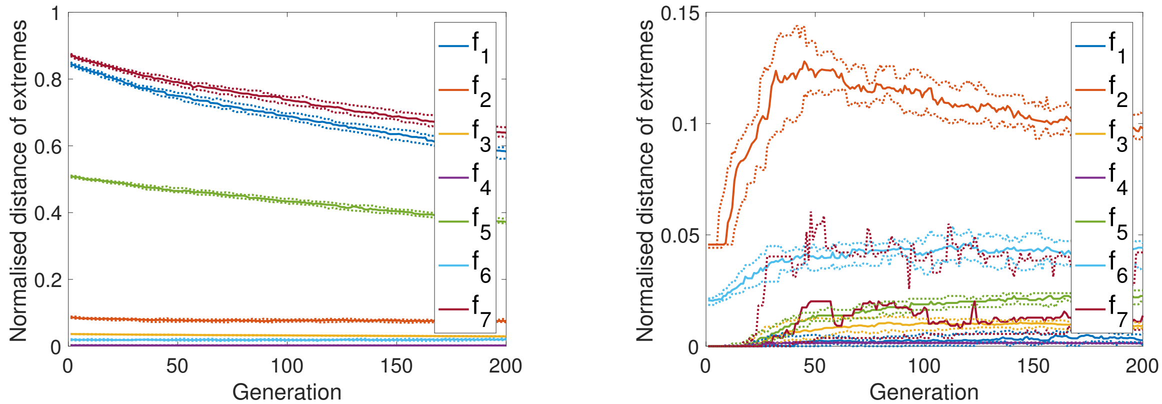

Figure 1: Median of minimum values of each criteria in non-dominated subset of search population. a) Random initial population, decomposition based on population properties. b) Seeded initial population, decomposition based on population properties. Interquartile ranges denoted with dotted lines.

is likely to be advantageous, but the degree of advantage it delivers is not known *a priori*. Furthermore, whereas NSGA-II [5] ensured that solutions which minimised individual criteria are not discarded from one generation to the next, this is not the case for NSGA-III (where if the parent set is smaller than the current estimated Pareto front, the latter is truncated in terms of distance from, and overcrowding of, the scalarisation rays used in the decomposition). We thus also examine the effect on the population members over generations in NSGA-III on our problem.

As a quality measure we estimate the hypervolume over time over multiple runs. Exact calculation of the hypervolume scales poorly with the number of dimensions, so it is infeasible to calculate exactly here. Instead we approximate the value via Monte Carlo (MC) sampling (as used in e.g. [7]). However, rather than taking a new set of MC draws each generation in the search, we can leverage past calculations to improved the fidelity of the hypervolume estimate as search progresses (as we maintain a passive unconstrained archive, as recommended in [11]). To be precise, we initially sample N random uniform points in the D-dimensional objective space, and map the objective vectors of the estimated Pareto front into the unit hypercube using the upper and lower criterion bounds we can determine directly (as derived in Section 2). The number of MC samples dominated can be counted, and used to estimate the volume dominated. The dominated samples can then be *discarded* (as they will always be dominated by subsequent updated passive archive members), and additional random samples taken in the unit hypercube to refill the samples to N points. This complete set is then used in the next generation and the process repeats (note that the cumulative number of discarded dominated samples each generation needs to be tracked). We use $N = 10000$ in our experiments. We have the advantage in this application domain of being able to derive the maximum values each criteria can take, with the minima for most criteria and lower bound values for the remainder (see Section 2). I.e. a good approximation to the coordinates of the minimum containing axis-parallel hyper-rectangle containing $\mathcal{Y}$ – the feasible objective space, which we sample for the hypervolume calculation. We do not however have access to the true Pareto front to determine what region in the minimum bounding box is unobtainable. As such, unless there exists a solution which minimises all criteria simultaneously, a hypervolume value of 1.0 is unobtainable in our experiments.

In all our experiments the probability of crossover is 0.8, $p_{outer} = 3$ and $p_{inner} = 2$, leading to a population size of 112 (please refer to [6] for further details on NSGA-III parameters and how they determine population size). 30 runs are taken for each experimental set-up with a different random seed, each run lasting 200 generations. We also generate an equivalent number of random solutions as a baseline.

Figure 1 shows the minimum values for each criteria stored in the active search population of NSGA-III (and). The left-hand panel shows the performance with a random initial population. The right panel using the same number of initial solutions, with seven of them being solutions which optimise, exactly or approximately, each of the individual criteria.

A number of interesting behaviours are apparent — the random initialisation provides good values for some criteria, but not for others (the y-axis of Figure 1 normalises each criteria between the minimum and maximum values possible). Additionally the right panel plainly show estimated front *shrinkage* as a number of the extremes in the front degrade as search progressed (see earlier work on this problem in multi-objective optimisers in e.g. [7, 8]). Those estimated initially with restart local search are also extremely close to their respective hypothetical minimum values (though global optimality is not assured).

The left panel of Figure 2 shows the median hypervolume over time calculated on the passive archive maintained by each of the NSGA-III configurations, and the random baseline. This is split into two panels as the random initialised runs (and random baseline) performs substantially worse than the extreme seeded runs (by an order of magnitude), with the distributions never overlapping. This underlines the positive impact of using the the extreme solutions in the initialisation.

The right panel of Figure 2 shows the proportion of the estimated Pareto front maintained by the NSGA-III search population that is

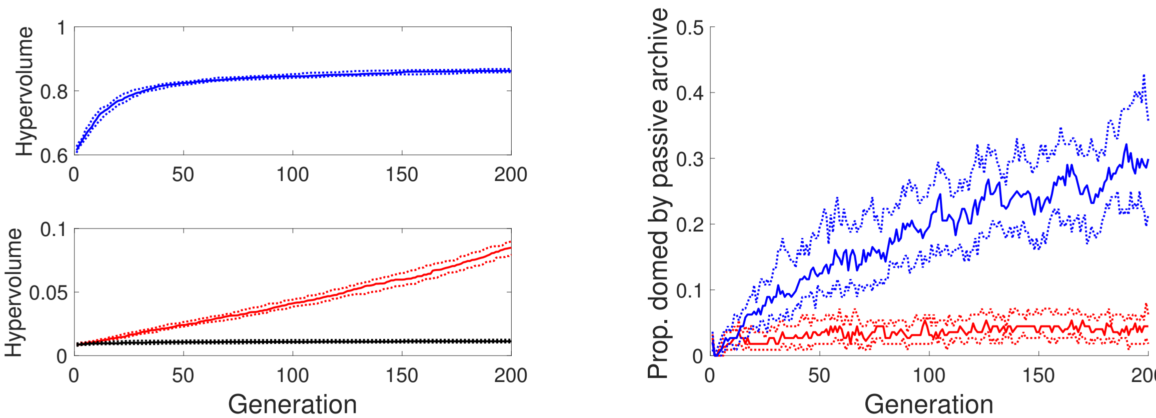

Figure 2: *Left*: Estimated hypervolume, calculations on passive archives. *Right*: Proportion of non-dominated subset of current NSGA-III population which is dominated by its passive archive at the same generation. Red line shows median results for unaltered NSGA-III (but with specialised crossover/mutation operators), blue line shows results when initial population is seeded with extreme solutions. Black line shows baseline generation of eqivalent number of random legal solutions (random search). Dotted lines denote inter-quartile ranges.

dominated by the corresponding passive archive, it therefore illustrates the degree of estimated front *oscillation* experienced during the search. This figure is interesting for two reasons. First, decomposition using scalarisation rays is generally considered to aid rapid convergence to a point on the Pareto front, however oscillation will impact on this. Figure 3 illustrates how this can occur using the selection routines in NSGA-III. The second point of interest is that the configuration which reports better hypervolume actually experiences worse oscillation. This may indicate the property worsens in many-objective problems as you converge using scalarisation. We also note that many MaOEAs have been developed and tested on continuous problems, which have a connected $\mathcal{Y}$. Where it may be impossible to achieve the objective combinations which lie on a scalarisation ray (as in this problem) such oscillation issues may be increased. Recent analysis of the dynamics of decomposition-based optimisation algorithms supports this, but also states that the weighting used in decomposition-based approaches can lead to an inferior solution being chosen over a superior solution (in a Pareto sense) from one generation to the next [1]. However, we note this observation is not a *general* property of decomposition-based approaches. It is not possible when using the decomposition mechanism in NSGA-III to select a dominated solution over a solution which dominates it, due to the initial ranking into non-dominated fronts. Instead it is the movement to a mutually non-dominated position which leads eventually to the quality deterioration in subsequent generations in NSGA-III.

5.2 Extended run and comparison

Following the initial investigation on specialising NSGA-III for the problem, we perform an extended run (10,000 generations). Parallel coordinate plots of the optimised solutions are shown in Figure 4. The two top panels show the final search population (raw values, and scaled by feasible range of criteria) and

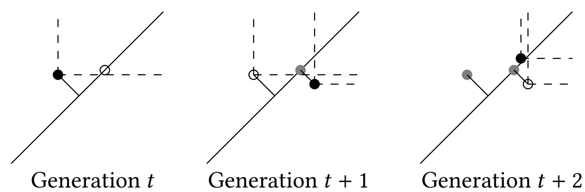

Figure 3: Illustration of scalarisation degradation for NSGA-III selection routine, e.g. identify non-dominated in search population, truncate if necessary based on distance to scalarisation rays. Filled circle denotes selected solution, unfilled circle denoted solution selected in previous generation. Grey circles denote selected solutions from earlier generations. At generation $t+2$ the selected solution is dominated by the solutions discarded at t and $t+1$.

the bottom panels shows the same for the passive archive. General run statistics are shown in Figure 5 (hypervolume and proportion of search population dominated by the parallel passive archive). The current allocation for 2016–17 is close to optimal (at least with respect to the passive archive), being dominated by just 76 elements of the archive. Its objective values, as a proportion of the range possible for each criteria (from the bounds), are $(0.309, 0.077, 0.012, 0.002, 0.043, 0.071, 0.397)$, where lower is better. The current allocation therefore prioritises the minimisation of staff dissatisfaction, at the cost mainly of overall load and churn. The root of this can be seen in the left panel of Figure 6 which shows the problem preference matrix, with staff preferring very few modules.

The archive had 27,847 elements (roughly 2.5% of the total number of many-objective evaluated solutions visited in the search). The right panel of Figure 6 shows how well the different criteria are correlated (based on the passive archive members of the single

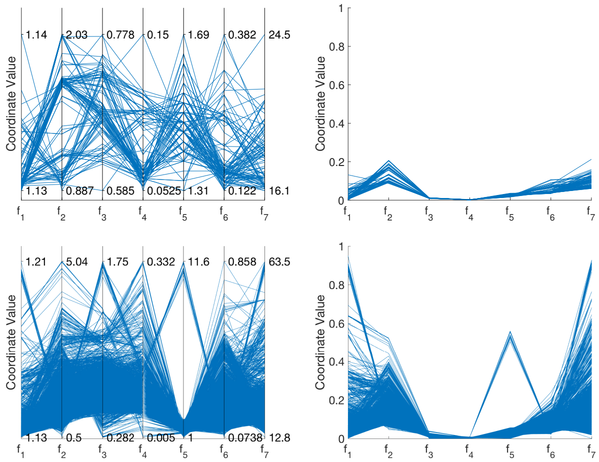

Figure 4: Parallel coordinate plots. Top pair: NSGA-III population, raw and normalised by criteria limits. Bottom pair: Same with passive archive.

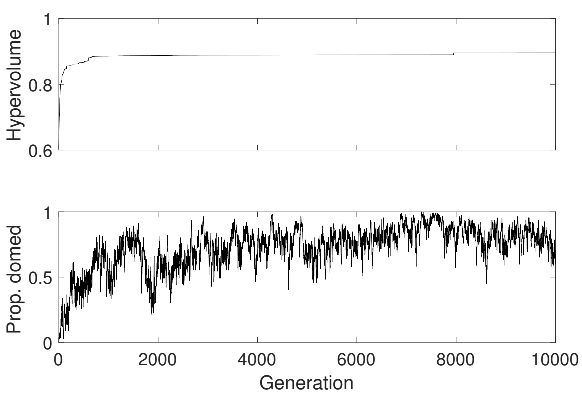

Figure 5: Statistics of the single NSGA-III run (including extremal seeds).

run) — the only large correlation is between f_1 and f_7 (as reducing churn will usually also reduce overall load).

6 CONCLUSIONS

We present the formulation and optimisation of the teaching allocation problem — a many-objective problem confronted by universities worldwide. We derive seven different criteria plus a number of constraints for this problem. As part of the process of identifying a set of trade-off solutions that may be of interest, we also highlight problem-specific augmentations of the NSGA-III algorithm that have been useful. We also raise some general points that would be interesting to explore further beyond this specific problem: the use of extreme solutions in seeding MaOEA runs, and the issue of front oscillation in decomposition approaches.

The optimisation of teaching allocation, as proposed here, requires the elicitation of teaching preferences from staff (albeit coarse

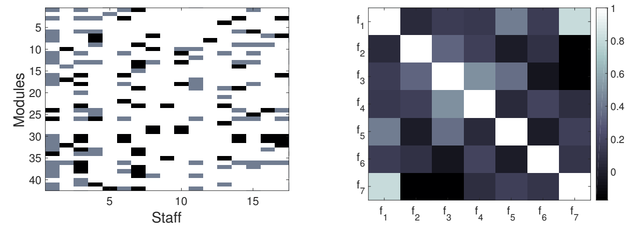

Figure 6: Left: Preference matrix. Black cells denote modules a staff member is happy to teach, grey those they could teach if required, and white those they feel they cannot deliver. Right: Correlation matrix (Pearson's linear correlation coefficient between criteria). Calculated on final passive archive from the long run of 10,000 generations.

ones). From a decision making process, separate from the generation of trade-off solutions, this can prove useful in identifying modules and teaching areas in risk (where few staff feel competent to deliver the material), which may influence both hiring and future course design. It also can highlight a wide disparity between staff in the range of modules they feel they can deliver.

There could well be other criteria and constraints which are faced by other departments, e.g. minimum amounts of teaching required of all staff, which some institutions enforce, and strict adherence to national maximum working time regulations. Embedding such additional aspects into the developed framework (available at https://github.com/fieldsend) is current work.

7 ACKNOWLEDGMENTS

The author would like to thank David Wakeling and Pete Vukusic, for their support in pursuing this work, and to David Walker, for suggesting that the work should be published.

REFERENCES

[1] H. Aguirre, S. Zapotecas, A Liefooghe, S. Verel, and K. Tanaka. 2016. Approaches for Many-objective Optimization: Analysis and Comparison on MNK-landscapes. In *12th International Conference on Artificial Evolution (LNCS)*, Vol. 9554. 14–28.
[2] S. Bouajaja and N. Dridi. 2016. A survey on human resource allocation problem and its applications. *Operational Research* (2016), 1–31.
[3] E. K. Burke and S. Petrovic. 2002. Recent research directions in automated timetabling. *European Journal of Operational Research* 140, 2 (2002), 266 – 280.
[4] D. W. Corne and J. D. Knowles. 2007. Techniques for Highly Multiobjective Optimisation: Some Nondominated Points Are Better Than Others. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07)*. ACM, New York, NY, USA, 773–780.
[5] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. 2000. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In *Parallel Problem Solving from Nature*. Springer, 849–858.
[6] K. Deb and H. Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Compuation* 18, 4 (2014), 577–601.
[7] J. E. Fieldsend, R. M. Everson, and S. Singh. 2003. Using Unconstrained Elite Archives for Multi-Objective Optimisation. *IEEE Transactions on Evolutionary Computation* 7, 3 (2003), 305–323.
[8] T. Hanne. 1999. On the convergence of multi objective evolutionary algorithms. *European Journal of Operational Research* 117 (1999), 553–564.
[9] E. J. Hughes. 2003. Multiple Single Objective Pareto Sampling. In *IEEE Congress on Evolutionary Computation*.
[10] C. H. Papadimitriou and K. Steiglitz. 1998. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications.
[11] D. van Veldhuizen and G. Lamont. 2000. Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. *Evolutionary Computation* 8, 2 (2000), 125–147.