



Mathematical modelling for academic performance status reports in learning analytics

A van der Merwe*

HA Kruger[†]

JV du Toit*

Received: 6 December 2017; Revised: 19 March 2018; Accepted: 20 March 2018

Abstract

The fast changing nature of the educational environment and the subsequent increase in the volumes of generated learner data, have found existing data analysis techniques lacking in certain fields. These techniques form part of the analysis and reporting phases of learning analytics and need to adapt to accommodate the changing face of education. In this paper, a set of interrelated algorithmic solutions that utilise mathematical programming models to generate and provide learning feedback in the form of academic performance status reports, is presented. Three existing mathematical models, more specifically the *benchMark* program, an outputs-only data envelopment analysis and a traditional analytic hierarchy process were evaluated for providing the information required to assist students in improving their academic achievement. The requirements include providing students with their current academic performance status, setting interim improvement goals and calculating improvement targets towards reaching those goals. The evaluated models did not address the requirements satisfactorily. The solution proposed in this paper consists of an algorithm that implements a linear programming model to generate performance status reports based on the current assessment scores of a group of students in a module. The output is used in a second algorithm that utilises the remaining improvement opportunities available to generate a participation future time perspective. The resulting schedule together with each individual student's current assessment scores, is used to calculate discrete improvement goals for each student as well as targets towards reaching those goals. A third algorithm provides a lecturer with some insight into the mastering of module content.

Key words: Analytic hierarchy process, data envelopment analysis, educational feedback, learning analytics, linear programming, non-linear programming.

*School of Computer Science and Information Systems, North-West University, Potchefstroom Campus, South Africa, email: Annette.VanDerMerwe@nwu.ac.za, Tiny.DuToit@nwu.ac.za

[†](Fellow of the Operations Research Society of South Africa) School of Computer Science and Information Systems, North-West University, Potchefstroom Campus, South Africa, email: Hennie.Kruger@nwu.ac.za

1 Introduction

The pace at which technology is advancing necessitates role players in education to adapt their teaching and assessment techniques [24]. Most of the activities relevant to education can now be performed by means of a computer. This makes online learning, where students do not need to be at a campus to interact with a lecturer, possible. The subsequent increase in student numbers results in an increase in the learner data being generated which includes the use of resources, academic performance, and even effectiveness of teaching tools. Analytical methods used to process student data consequently need to adapt as well. Many tools implemented as educational aids already perform some analytical functions on institutional, regional, national, and international level [39].

Predictive analytics incorporates statistics, machine learning and data mining to discover meaningful patterns in data [25]. The focus of learning analytics is, however, on the learning process rather than on an institutional level [39]. It presents the potential to predict and improve student success and resulting retention by incorporating the measurement, collection, analysis and reporting of learner data [8]. Although many methods exist with which these actions can be performed, measurement and collection of learner data are mostly context dependent. Various methods for analysing and reporting such data are commonly implemented in academic environments by means of *learning management systems* (LMSs), which consider learning activities, course administration, and tracking [40]. Examples like PACE (Progress and Course Engagement), GPS (Grade Performance System) [31], and E2Coach (Expert Electronic Coach) [27] are only some LMSs that determine and provide alerts regarding academic issues. Other systems that can furthermore present interventions include the Student Advice Recommender Agent (SARA) [17] and the MiGen teacher assistance tool, which is a system that was developed with some visualization and notification capabilities [26]. However, Stantchev *et al.* [40] argue that LMSs are not yet optimally contributing towards the improvement of learning because they mostly address institutional needs rather than those of the students. Students need specific information relating to their progress in a course to enable change [13].

Learning analytics can change the value of higher education by assisting in the identification of students who are at risk of failing and to provide them with interactive solutions on how to improve during the semester (term). Barker and Garvin-Doxas [4] proposed the use of a hierarchical structure for the purpose of progress feedback to create a communicative environment between a lecturer and the students. Studies show that ranking is an important determinant of student behaviour [14, 42]. Therefore, a solution in the form of a student ranking incorporated into the analysis and reporting phases of learning analytics, is proposed.

Ranking models have been applied in the ranking of NBA basketball players [28], college sport coaches [19], alternatives in sports tournaments, products, political candidates [6], and others such as the PageRank algorithm introduced by Google [21]. The main aim of this study is to evaluate existing ranking models that may be used for student feedback and to develop an improved ranking feedback model by implementing mathematical programming techniques in three interrelated algorithms. Three models were chosen for evaluation in this specific research project. They include *benchMark* [11], an outputs-only based *data envelopment analysis* (DEA) model [22], and the traditional *analytic hierarchy*

process (AHP) [2]. The *benchMark* program was selected because it is an empirical student ranking model that ranks students according to their academic progress in a module. It was successfully implemented in a software application and was evaluated in an academic environment with positive results. Furthermore, the *benchMark* program is directly related to this study. The outputs-only based DEA was chosen because in general, no measurable inputs (*e.g.* numbers of hours studied) are available on students. It allows the calculation of outcome improvement targets and it is an established ranking method. The AHP is another technique that ranks items influenced by several factors, or in this study, to rank students according to their formative assessment outcomes like class test scores, practical assignments, *etc.* Appropriate adjustments and pairwise comparisons were made in this study to eliminate traditional inconsistency- and normalisation problems. This process also allows for the allocation of weights (as predetermined by a lecturer) to the factors. The AHP was applied in this context to operate as an additional means of appraising the rankings obtained in other methods.

The motivation for the study is founded in the repeated observation that student attendance in some courses at tertiary institutions, seems to wane after the initial contact session. Students think that because the semester had just started their absence cannot yet affect their academic progress and when they do return to class, they do not know what to do to improve their performance [11]. There is a demand for an academic performance feedback system that will continuously provide students with information on their current academic status in a module as well as how to improve from an early stage in the semester [18].

The remainder of this paper is organised as follows: Section 2 provides some background on a typical teaching-learning environment and a short discussion on the requirements for effective feedback. Subsequently, the three chosen ranking models are discussed in Section 3. Finally, an improved mathematical ranking model that provides effective feedback to students is proposed in Section 4. This is followed by a discussion in Section 5 and conclusions in Section 6.

2 The teaching-learning environment

In a typical classroom environment, the role of a lecturer ranges from teaching and facilitation to assessment and grading. Students are expected to prepare for and attend contact sessions where lecturers will discuss previously completed work, explain how it relates to new concepts, and introduce and discuss new material. Often during contact sessions students are expected to complete class assignments and may also be given homework related to topics discussed during class. Regular formative assessments in the form of class tests, practical assignments or informal tuition tests among others, can be taken without prior notice. This form of assessment provides lecturers with valuable information on the success of the teaching techniques employed when introducing new material to students.

Lecturers grade formative assessments and use them to compile an overall semester mark for each student. This mark, called a participation mark (*p*-mark), is the measurement by which students are granted admission to the final examination. Some lecturers also reward students for class attendance by letting it contribute a small quantity towards the

p -mark. In the calculation of the p -mark a lecturer has the responsibility to decide which factors he/she wants to consider, and how the averages in each of these factors are to be calculated. The p -marks can then be calculated as the weighted sum of the average marks obtained for each factor according to

$$p = y_1w_1 + y_2w_2 + \cdots + y_nw_n, \quad (1)$$

where p is the participation mark, $y_1, \dots, y_n$ are the respective average marks calculated, $w_1, \dots, w_n$ are the respective weights, and n is the number of factors. A lecturer normally decides on applicable weights for each of the respective factors. In some environments, if the p -mark is larger than a certain minimum threshold the student is allowed to write the final examination. The final module mark is then calculated as the average of the p -mark and the examination mark. This means that students have the opportunity to start building half of their module mark from as early as the first scheduled contact session. Students are, however, not always aware of how they are progressing towards their p -marks. Regular and interactive feedback is needed to keep students informed on how their p -marks are growing.

Ranking is considered a feedback format that will effectively inform students on their positions in a hierarchy based on their outputs. The proposed solution will provide feedback in the form of *performance status reports* (PSRs) throughout the semester. For a PSR to be effective it must address three key issues [18]. Firstly, students must know what the goal is that they are working towards. This requirement is referred to as *feed-up*. At the start of a semester in an academic environment such a goal can be to pass a module or get a distinction. However, as the semester advances and students start to progress or regress, interim goals based on their performance can help them reach their ultimate goals. Secondly, they must know what progress they are making towards reaching the goals, or *feed-back*. Lastly, students must be informed as to which activities they have to perform in order to enhance their progress, or *feed-forward*. Furthermore, Wiggins [45] provided some attributes to which the information in a PSR must conform to. Apart from having to provide feed-back, feed-forward and feed-up, the reports generated by the solution in this paper will also be evaluated for referring to a goal, being actionable, user-friendly, timely, ongoing and for being provided consistently. In the next section, different methods for presenting student progress in the form of academic rankings, are evaluated.

3 Ranking as feedback in learning

Ranking as a hierarchical structure is used in the academic environment for many purposes, among others for evaluating quality in higher education [9, 16], for determining the success of different departments within a university [5], for student selection [15] and for ranking specific student attributes to determine the probability of success [35, 37]. In this section, the three selected models will be discussed and applied to the same data set. This set contains marks for 26 students in four factors that were achieved in a fourth-year module presented at a tertiary institution. The factors (denoted F_1 to F_4 respectively) are a semester test, class tests, theoretical assignments and attendance. The data set is provided in Appendix A.

3.1 The *benchMark* model

A software application called *benchMark* was developed by Du Toit [11] for the purpose of timeously calculating p -marks and rankings for a group of students. The program creates a participation profile for each student which is continuously provided to them as PSRs throughout the semester.

The model utilised in this program is a standardised form of equation (1). After completion of each assessment the new set of marks is used to calculate a new average for each factor. Each student's current factor averages are then used to create a p -mark according to

$$p_\alpha = \sum_{j=1}^n y_{\alpha j} w_j, \quad (2)$$

where w_j is the weight assigned to factor j , $y_{\alpha j}$ is the calculated average for factor j of student α , and n is the number of factors. A unique attribute of *benchMark* is that instead of requiring a specific known weight for each factor beforehand, it empirically calculates a set of possible p -marks for each student using a range of permissible weights per factor. Permissible weights have minimum and maximum values, with their sum constrained to be 1. A drawback to this approach is the exponential increase in the number of possible p -marks that need to be calculated as the number of factors increase. Initially it was found that the calculation of all possible combinations of weights would result in an intractable problem [11]. To cater for this, a priority sequence constraint was implemented on the weights and they were constrained to be in multiples of a predetermined number k . The priority sequence constraint requires the lecturer to predetermine the order of importance for each of the factors. It was empirically determined that by enforcing these constraints the search space was reduced considerably, resulting in a tractable problem.

The program also calculates the average of all the possible p -marks for every student, including the minimum and maximum. After each execution of the program the maximum, minimum and average p -mark, and a student ranking according to the average p -marks, are available. Statistics relating to the current pass/fail rate, the number of distinctions and the number of students without admission to the examination, are also available. During the semester students are frequently given feedback on their latest factor averages, their minimum p -marks, and ranking. Instead of informing the students what the respective weights for the factors are going to be at the start of a semester, the option exists to provide them with the order of the priority of the factors. This affords some freedom in selecting a specific factor weight distribution closer to the examination and students have a discrete measuring tool with which they can gauge their progress and/or improvement in the module. The program therefore guarantees that the final weight distribution selected at the close of a semester, will not cause any student's final p -mark to drop lower than the minimum mark provided to them after having processed all relevant assessments.

The model was implemented and tested with $k = 0.05$. The priority sequence applied in the evaluations was $w_1 \geq w_2 \geq w_3 \geq w_4$ with the weights applying to F_1 to F_4 respectively. Based on these constraints the program calculated a set of 23 possible p -marks for each student from which each respective student's average p -mark was calculated. For illustration purposes the participation profiles of the top seven students are presented in Table 1 (the profiles of all the participants are provided in Table 9 in Appendix B).

Student	F_1	F_2	F_3	F_4	Max	Min	Avg	Rank
2	100.00	95.24	97.92	100.00	98.29	97.58	97.85	1
9	93.33	91.51	87.75	100.00	93.15	91.35	92.33	2
3	96.67	82.62	75.05	100.00	89.22	85.43	87.44	3
13	76.67	87.13	65.83	89.00	83.35	78.25	81.05	4
10	93.33	68.15	74.07	100.00	84.89	78.78	81.04	5
24	90.00	64.76	76.67	100.00	82.86	77.07	79.37	6
23	85.00	72.14	61.85	100.00	79.75	74.31	77.35	7

Table 1: Participation profiles created by *benchMark* for the top seven students. $F_1 - F_4$ are the factor averages arranged according to weight priority. The columns Max, Min and Avg, are the maximum, the minimum and the average participation marks, respectively. The column Rank is the student ranking according to average participation marks. All marks are indicated in percentages.

Apart from providing student p -marks and ranking status, other advantages of the *benchMark* program include the following:

- Performance status can be provided to students as frequently as assessments are graded; and
- Additional throughput- and performance information is available to the lecturer.

Although *benchMark* has been successfully implemented, it uses all weight combinations allowed within the model constraints to empirically calculate the p -marks and ranking.

The disadvantages that were identified are listed below.

- Although some students (especially those at the top of the ranking) prefer to know their ranking, others exhibit a rather indifferent attitude towards their progress and feel that they need discrete information on how to improve to higher ranking positions. The *benchMark* program does not provide improvement targets that will assist in improving students' ranking.
- The program uses empirical calculations in a sequential algorithm to create a set of p -marks for each student and to set up statistical information. It was found that for less than 80 students the execution time is reasonable and the program can be executed a few minutes prior to a contact session [11]. However, for large student numbers program execution time does not scale linearly and it may be more practical to perform the calculations only once a week.
- The *benchMark* program initially has a steep learning curve, both for students and lecturers.
- Execution of the *benchMark* program needs continuous human involvement making importing of marks and calculating new p -marks rather labour intensive.
- Without the priority sequence- and the multiples-of- k constraint sets the search space is so large that the problem is intractable.

To address some of the disadvantages identified in evaluating *benchMark* (the human factor, the iterative and sequential nature of the program, and its intractability), the model was formalised in the non-linear mathematical model [43] that follows.

$$\text{Maximise/minimise } p_\alpha = \sum_{j=1}^n y_{\alpha j} w_j, \quad (3)$$

subject to the constraints

$$\sum_{j=1}^n w_j = 1, \quad (4)$$

$$w_j \bmod k = 0, \quad j = 1, \dots, n, \quad (5)$$

$$l \leq w_j \leq u, \quad j = 1, \dots, n, \quad (6)$$

$$w_j \geq w_{j-1}, \quad j = 1, \dots, n, \quad (7)$$

$$w_j \geq \epsilon, \quad j = 1, \dots, n, \quad (8)$$

with $k \in \mathbb{R}^+$, $l \in \mathbb{R}^+$, $u \in \mathbb{R}^+$ and ϵ a small positive number.

The objective function is similar to the formulation in (2). The constraint set in (5) ensures that each w_j is a multiple of k while (7) represents the sequence ordering requirement of *benchMark*. Weights are constrained to be between an upper and lower limit in (6).

The model is non-linear (due to constraint set (5)) and may be solved using any non-linear mathematical programming software. For the purpose of this study it was decided to obtain a heuristic solution using the Evolutionary Solver in Excel. Evolutionary solvers are designed to mimic the process of biological evolution in certain ways. A full discussion of heuristic solutions and the optimisation of non-smooth models is beyond the scope of this paper and detailed explanations can be found in Baker [3]. The choice of a solver was based on the fact that the solution generated in this manner does not rely on the mathematical structure of the model but rather on solutions generated by combining previous solutions along with some randomness [33]. It should be noted that there is no guarantee of a global optimum and human judgement is required when solving a model using an evolutionary approach. The implementation of a heuristic solution using an evolutionary solver is simple and allows for easy use of mathematical functions such as the modulus function in constraint set (5).

The same data set used to illustrate *benchMark*, was used to evaluate the non-linear model in (3)–(8). The model was solved again with $k = 0.05$, and the same priority sequence applied to the factors as in *benchMark*. The maximum and minimum p -marks were calculated accordingly, and the average p -mark was determined from these two values only. The students were again ranked according to the average p -marks as produced by the model in (3)–(8).

Table 2 shows the calculated p -marks, the top seven of the subsequent rankings, and the associated weights generated by this heuristic solution.

The ranking for all 26 students (all student rankings can be found in Appendix B, Table 10) differs marginally from that produced by *benchMark*. Student 10, for example, is ranked 4th whereas this student was ranked 5th by *benchMark*. The reason for this difference is the fact that *benchMark* calculated the average p -mark for each respective student from all 23

Student	<i>p</i> -marks (%)			Rank	Weights for minimum <i>p</i> -mark				Weights for maximum <i>p</i> -mark			
	Max	Min	Average		w_1	w_2	w_3	w_4	w_1	w_2	w_3	w_4
2	98.63	97.78	98.21	1	0.40	0.40	0.15	0.05	0.40	0.20	0.20	0.20
9	93.15	91.44	92.30	2	0.35	0.30	0.30	0.05	0.25	0.25	0.25	0.25
3	90.20	86.13	88.17	3	0.35	0.30	0.30	0.05	0.40	0.20	0.20	0.20
10	84.52	80.04	82.28	4	0.35	0.35	0.25	0.05	0.35	0.25	0.20	0.20
24	84.27	78.33	81.31	5	0.35	0.35	0.25	0.05	0.40	0.20	0.20	0.20
13	80.54	77.17	78.29	6	0.35	0.30	0.30	0.05	0.35	0.35	0.15	0.15
23	80.80	74.95	76.90	7	0.35	0.30	0.30	0.05	0.40	0.20	0.20	0.20

Table 2: Results of the top seven students, obtained by the non-linear model.

of the empirically calculated possible *p*-marks as opposed to the non-linear model which used only the calculated maximum and minimum values. To produce a more structured comparison between the two rankings, a Spearman’s rank correlation coefficient [7] was calculated. Spearman’s rank correlation coefficient (ρ) is a nonparametric assessment of the statistical relationship between two different rankings which expresses the relationship between them in terms of a monotonic function. It is expressed as

$$\rho = 1 - \frac{6 \sum_{i=1}^m d_i^2}{m^3 - m}, \quad 0 \leq \rho \leq 1, \quad (9)$$

where d_i is the difference between the two ranks at the i^{th} position, and m is the number of ranking positions. The rank correlation coefficient between the two rankings was calculated as 0.99, proving a very strong positive association between them. This is to be expected considering the similar constraints applied in both methods.

Formalisation of the *benchMark* model in a non-linear mathematical model yielded improvements on some of the disadvantages identified earlier:

- The need for continuous human intervention is reduced as the model parameters are specified at the onset of execution and the calculations are repeated for each student by means of an algorithm.
- The evolutionary solver is easy to implement and converges to acceptable solutions for the entire data set, faster than *benchMark* did. This model can therefore be used more frequently and for large student numbers.
- Although a heuristic solution does not always guarantee to be optimal, the model does find acceptable solutions. This was proven by the high Spearman’s rank correlation coefficient calculated for the respective rankings resulting from the *benchMark* program and the non-linear mathematical model.

The *multiples-of-k* constraint set (5) was enforced in the non-linear model so that the resulting ranking could be compared with that obtained in *benchMark*. However, this constraint creates a non-linear relationship between the decision variables and some pre-determined constant.

Enforcing this constraint set was justified in *benchMark*, but its exclusion will result in a linear mathematical model which, under certain conditions, can be solved in polynomial

time [30]. The model was therefore modified and solved as a linear model, which is discussed in more detail in Section 4.1.

Another drawback to the *benchMark* and non-linear modelling approaches is that no discrete improvement targets are calculated with which students can attempt to improve their rankings. This means that although these methods address the feed-back requirement for effective PSRs, no feed-forward or feed-up is provided. In a pilot study done by Du Toit [11] it was noted that although students at the top of the ranking felt motivated to improve, those in the middle and lower parts of the ranking indicated that they wanted more information on exactly how to improve. One of the attributes PSRs must have, is that they must be actionable. In other words, students have to be provided with some form of incentive or target that shows them what they can do to improve their rankings or more specifically, feed-forward. This problem was partly addressed by implementing an outputs-only based data envelopment analysis (DEA) on the student mark data set [44], and is discussed next.

3.2 Class-ranking by means of an outputs-only based DEA

DEA is a mathematical programming method commonly employed to compare service units called *decision making units* (DMUs) in a specified environment, based on a calculated ratio between their inputs and outputs [29]. However, student input data such as study environment, hours of study or lifestyle aspects are considered either unavailable or unreliable. Therefore, a model that utilises inputs was not included in this study. Outputs-only based DEA effectively categorises DMUs considered equally efficient into groups or classes. Examples of such analyses include the class-ranking of management colleges [22], and of secondary schools [23]. To be able to compare the student rankings, an outputs-only DEA was performed on the marks (factor averages) obtained in certain outputs (factors) [44]. The students were treated as DMUs and the factor averages were considered the outputs. This method effectively searches for the most favourable weights and calculates an efficiency score for each student [22] by solving the model in (10)–(12).

$$\text{Minimise } E_{\alpha} = \sum_{j=1}^n y_{\alpha j} w_j, \quad (10)$$

subject to the constraints

$$\sum_{j=1}^n y_{ij} w_j \leq 1, \quad i = 1, \dots, m, \quad (11)$$

$$w_j \geq \epsilon > 0, \quad j = 1, \dots, n, \quad (12)$$

where E_{α} is the composite index for student α , y_{ij} is the output measure of student i in factor j , w_j is the weight for factor j , n is the number of factors, m is the number of students to be evaluated, and ϵ is a small positive number ensuring all factors are considered.

This model was used to create a class-ranking that effectively sorted the students into groups of similar efficiency. The principle of Pareto optimality applies wherein the efficiency score (E_α) represents the ratio of the current student's performance to that of all the other students in that specific factor. If $E_\alpha = 1$, then student α is considered Pareto optimal which means that no output can increase without decreasing another [41]. Such a student would then be categorised into the highest available class, lie on the Pareto frontier, and is said to dominate those in lower classes. If $E_\alpha < 1$, then student α is Pareto non-optimal and is said to be dominated by those in higher classes. Detailed discussions of Pareto optimality and the dominance principle can be found in Kao and Lin [22], and Seiford and Zhu [38].

The outputs-only DEA was also performed on the same data set containing marks for 26 students, in which the four factors as described in the previous sections, are considered as outputs. This resulted in a class-ranking that categorised the students in the data set into 8 different classes of efficiency (Figure 1).

Number of students in each class

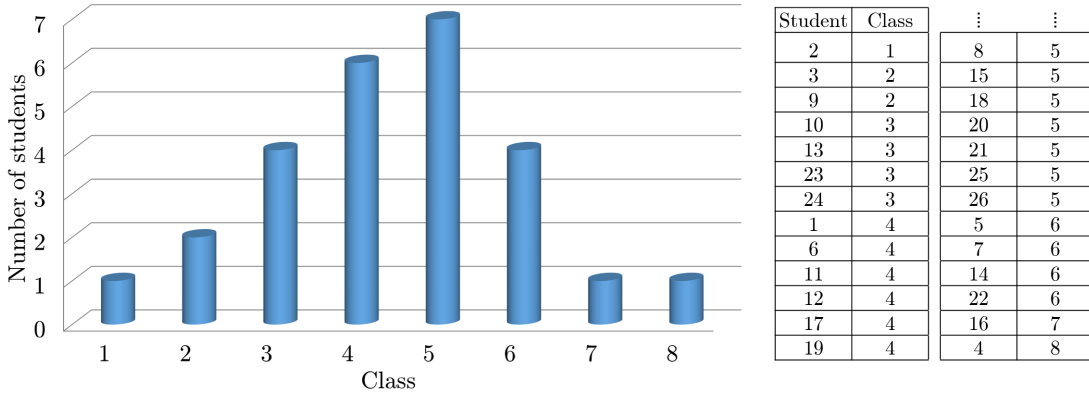


Figure 1: Class-ranking of the 26 students.

Figure 1 shows the class-ranking: one student was classified into class 1, followed by two students in class 2 and so forth. The student class-ranking cannot directly be compared to the rankings obtained in Section 3.1, for a few reasons. The outputs-only based DEA method calculates the efficiency scores by determining the weights that will benefit each student most, with no predetermined factor priority sequence included. Also, the basis for calculating E_α is determined by the outputs of the students on the Pareto frontier and is therefore not necessarily the same for each dominated student [16]. Although the class-ranking produced by this model cannot be compared to the rankings resulting from the methods in Section 3.1, the students classified into the first three classes, correspond with the top seven from Section 3.1.

One of the disadvantages identified in Section 3.1, was that neither the *benchMark* program nor the non-linear model offered students any actionable targets with which they could attempt to improve their rankings. A major advantage presented by the outputs-only DEA model formulated in (10)–(12), is that by implementing its dual formulation, discrete

improvement targets per student per factor could be calculated. The dual formulation [22] is expressed in (13)–(15).

$$\text{Minimise } E_\alpha = \sum_{i=1}^m \lambda_i - \epsilon \sum_{j=1}^n s_j, \quad (13)$$

subject to the constraints

$$\sum_{i=1}^m y_{ij} \lambda_i - s_j = y_{\alpha j}, \quad j = 1, \dots, n, \quad (14)$$

$$\lambda_i, s_j \geq 0, \quad i = 1, \dots, m; \quad j = 1, \dots, n. \quad (15)$$

The variable definitions are the same as for the model in (10)–(12). Furthermore, let $\theta = \sum_{i=0}^m \lambda_i$, then the target for factor j can be calculated as $(y_{\alpha j} + s_j)/\theta$ and $j = 1, \dots, n$.

For each student not in class 1 ($c \neq 1$), this model would calculate $n(c-1)$ improvement targets, one per factor for every class higher than the current. Consider, for instance, Table 3 that shows Student 1 who was categorised into class 4.

c	Factor(%)			
	F_1	F_2	F_3	F_4
4	45.00	55.00	50.00	75.00
Targets for improvement (%)				
3	49.00	58.00	53.00	80.00
2	64.00	71.00	77.00	98.00
1	65.00	94.00	84.00	100.00

Table 3: Improvement targets for Student 1.

If Student 1 wanted to progress to class 3, the semester test (F_1) must improve from 45% to 49%, class tests (F_2) from 55% to 58% and so forth. Likewise, for each student not categorised into the highest class, targets were calculated in each factor for improvement into the next higher class. Unfortunately the model does not cater for factors that have been completed. A semester test for example, is written once in the middle of the term and the mark cannot change after that. Also, any changes in student marks will have a ripple effect on the entire class–ranking as follows:

- the model would need to be solved again and new optimal weight sets will be determined for each student;
- new weights per student could mean that the Pareto frontier for each class could change; and
- the outputs of all students are used in calculating the improvement targets. This means that even if a student reaches a target, the class–ranking might not change if other students improved in the same factor.

Another drawback to the outputs-only based DEA is that because the output of a student is compared to that of every other student, the size of the problem increases considerably when performing this analysis for modules with large student numbers. A solution to this problem is proposed through the construction of an algorithmic approach which is discussed in detail in Section 4.2.

The outputs-only DEA model can be used to calculate individual improvement targets for each student, but it has certain disadvantages. Another student ranking was generated in this context by means of the AHP for comparative purposes, and is discussed in the next section.

3.3 An AHP model for ranking students

The *analytic hierarchy process* (AHP) is a mathematical method used to make decisions when multiple objectives or criteria need to be considered [34]. The method is used to find the best contender among a certain number of alternatives (students), based on a number of criteria (factors).

Each one of the students is compared to the others in terms of each of the factors. This is followed by a comparison of the importance of the factors resulting in an average rating for every student.

Typically, the following steps are performed [29]:

1. Develop a ranking for each student for each factor:
 - develop a pairwise comparison matrix for the students in each factor;
 - normalise the resulting matrices;
 - find the normalised principal eigenvector, or priority vector; and
 - check the consistency of the pairwise comparisons.
2. Develop the weights for the factors:
 - develop a pairwise comparison matrix for the factors;
 - normalise the resulting matrix;
 - find the priority vector; and
 - check the consistency of the pairwise comparisons.
3. Calculate the weighted average rating for each student. Rank them according to their average ratings.

Pairwise comparisons are based on a preference scale which usually ranges from 1 to 9 [1]. The ratings show how much one alternative is preferred above another, and vice versa. In this application pairwise comparisons were made with student factor averages and the rating scale was modified as follows: At tertiary level an *admission point score* (APS) is used to determine whether a learner who completed the South African National Senior Certificate, may study at a South African university [12]. An adaptation of the APS system was therefore used for generating the pairwise comparison ratings of student averages where the difference between the mark for factor j of student α and student β is calculated as

$$y_{\alpha j} - y_{\beta j}, \quad \alpha, \beta \in \{1, \dots, m\} \text{ and } \alpha \neq \beta, \quad (16)$$

with m representing the number of students. These calculations are then associated with the normal AHP preference scale of 1–9. Table 4 shows this association for $y_{\alpha j} \geq y_{\beta j}$. For cases where $y_{\alpha j} < y_{\beta j}$, the inverse of the rating is used. If for example $y_{\alpha j} < y_{\beta j}$ for the interval 20%–29% then the corresponding rating would be $1/4$.

if $y_{\alpha j} \geq y_{\beta j}$	Rating
0%	1
1-9%	2
10-19%	3
20-29%	4
30-39%	5
40-49%	6
50-59%	7
60-79%	8
$\geq 80\%$	9

Table 4: Preference rating for pairwise comparison of academic marks.

The non-linear model discussed in Section 3.1 yielded a set of weights used for calculating the minimum and maximum p -marks of each student (Table 2). To enable comparison between the AHP ranking and those obtained in previous sections, the weights were pre-set to averages calculated as

$$\bar{w}_j = (\bar{w}_{j(max)} + \bar{w}_{j(min)})/2, \quad (17)$$

where $\bar{w}_j$ is the average weight for factor j , $\bar{w}_{j(max)}$ is the average of all the weights calculated for factor j for the maximum p -marks and $\bar{w}_{j(min)}$ is the average of all the weights calculated for factor j for the minimum p -marks. The weights used in the AHP are 0.35, 0.30, 0.20 and 0.15 for $\bar{w}_1$ to $\bar{w}_4$, respectively. The weights can be varied at the lecturer's discretion but lecturers do not always know what the expected class throughput resulting from specific weight combinations might be. This problem was addressed by formulation of a non-linear weight selection model and is discussed in more detail in Section 4.3.

Consistency checks on pairwise comparisons are usually required because with a large number of decision alternatives inconsistent comparisons occur frequently. A consistency ratio is calculated [10] using the formula

$$CR = CI/RI, \quad (18)$$

where

$$CI = (\lambda_{max} - m)/(m - 1) \quad (19)$$

is the consistency index with λ_{max} the principal eigenvalue, m the number of students, and RI is a random consistency index [36]. Generally, the degree of consistency is acceptable if $CR < 0.10$. The scale used for pairwise comparison of student marks in this study is considered consistent because it compares each pair of marks by means of a mathematical equation (Table 4). Therefore, it was not necessary to perform consistency checks for these pairwise comparison matrices.

The number of comparisons (s_m) to be made for m students and n factors, is expressed as

$$s_m = mn\left(\frac{m-1}{2}\right). \quad (20)$$

Accordingly for the mark set of 26 students and four factors, a total of 1300 pairwise comparisons were made.

The process was implemented on the same mark set used previously. The marks of the students were rated according to Table 4 by calculating the difference between each pair of marks and setting up the pairwise comparison matrix based on the given preference scale. The matrix was completed and normalised for all of the 26 students in the set after which the averages were calculated to determine each factor priority vector. This step was performed for all four of the factors. Using the preference weights as discussed previously, a composite rating $r_{\alpha j}$, was calculated for each student by finding

$$r_{\alpha} = \sum_{j=1}^n \gamma_{\alpha j} w_j, \quad \alpha = 1, \dots, m, \quad (21)$$

for student α , with $\gamma_{\alpha j}$ the average priority rating of student α in factor j , w_j the pre-set weight for factor j , m the number of students, and n the number of factors.

The students were then ranked according to the calculated composite ratings. Table 5 shows the top seven rankings determined by the AHP (the complete set is available in Table 11, Appendix B).

Student	Priority Vector				Composite Rating $r_{\alpha j}$	Pairwise Ranking
	F_1	F_2	F_3	F_4		
2	0.1019027	0.1226755	0.145867	0.0612267	11.0826	1
9	0.0793605	0.1101616	0.1166742	0.0612267	9.334351	2
3	0.0872195	0.0810881	0.0704773	0.0612267	7.813273	3
10	0.0793605	0.0506373	0.0643211	0.0612267	6.501559	4
24	0.0614319	0.0370232	0.0824396	0.0612267	5.828006	5
23	0.0491764	0.0670063	0.0387908	0.0612267	5.425578	6
13	0.0326883	0.0977652	0.0460599	0.0283514	5.423514	7

Table 5: Top seven of the student ranking resulting from the AHP.

The use of the AHP on student marks in different factors, is an unconventional approach to student ranking. It creates a ranking without the calculation of p -marks and can be used to provide an alternative form of feedback. The Spearman's rank correlation coefficient, ρ (Section 3.1) was calculated to compare the AHP ranking to those obtained previously. The AHP ranking correlated with the *benchMark* ranking by $\rho = 0.97$, and with that of the non-linear model, by $\rho = 0.98$.

One of the problems with using the AHP is that although the resulting ranking highly correlated with those from other methods, a different normalisation order can lead to different rankings [32]. Another disadvantage is that the AHP is difficult to use with large

volumes of data because the number of pairwise comparisons that need to be made is directly related to the number of alternatives to consider [20]. This method also did not address the feed-up or feed-forward requirements. Despite these difficulties, the results showed the AHP to be another technique which can be used for ranking students. For this study, the AHP also acted as an efficient control measure, by which the rankings obtained in the other methods could be evaluated according to the Spearman's rank correlation coefficient.

The existing mathematical models discussed in Section 3 were implemented on the same data set to generate different student rankings. The rankings were evaluated according to specific criteria essential to a PSR, as identified in literature and practice. Although some of the criteria were addressed, other issues remain unattended. To this end, a solution consisting of three different but related algorithms, is proposed and is discussed in Section 4.

4 Proposed mathematical model for effective PSRs

The requirements for effective PSRs in an academic environment were identified in Section 2 and used as evaluation guidelines for the three methods evaluated in Section 3. An effective PSR needs to address these important aspects: it must feed-up or provide a goal to work towards, feed-back or present current performance, and feed-forward or offer actions to perform towards reaching the goal. The evaluated methods were implemented as mathematical models which were chosen for specific reasons and applied on the same data set for comparative purposes. Although each one of those methods offered a ranking solution and addressed some of the requirements, none of them satisfied all of the criteria for PSRs. The *benchMark* program, non-linear model and the AHP model provided feed-back but neither feed-forward nor feed-up. The outputs-only DEA model addressed the feed-forward criteria by providing factor improvement targets and partially addressed the feed-back criteria in the form of a class-ranking. However, numerous disadvantages were identified that makes this method unsuitable for the use in PSRs. Evaluation of these methods is illustrated in Figure 2.

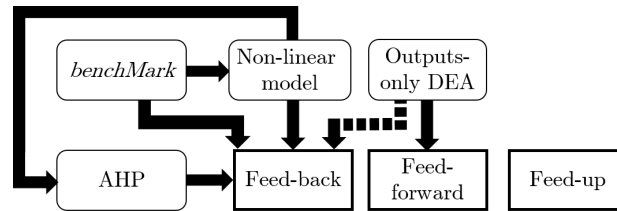


Figure 2: Evaluation of selected methods for PSRs.

As Figure 2 illustrates, none of the methods comprehensively addressed the feed-up requirement which demands that students be continuously presented with new goals aimed at progression. The requirements for an effective PSR has therefore not been addressed completely. Evaluation of the methods in Section 3 presented some valuable insights that led to the proposal and development of an algorithmic solution that is illustrated in Figure 3 and discussed in subsequent sections.

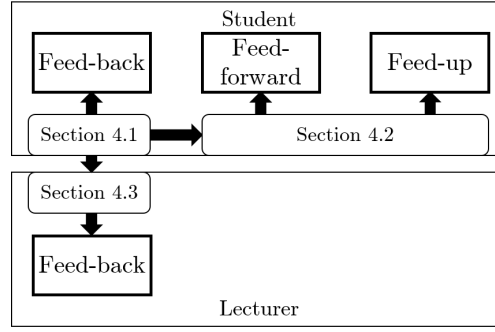


Figure 3: Proposed solution to address PSR criteria.

An important issue to consider when generating the information required for PSRs, is the fact that as time passes by, so does a student's opportunity for improvement. At the start of a semester when only a few assessments have been completed, the factor averages are calculated from the only scores available. Each assessment therefore has a greater influence on the calculated p -mark at an early stage. As a semester elapses and the number of assessments remaining in each factor decreases, the contribution of each individual assessment towards the p -mark diminishes and the opportunities to improve decreases. Figure 3 illustrates how the requirements for PSRs relate to one another in the proposed solution. A linear model for p -mark calculation and its associated algorithm are discussed in Section 4.1. This is followed by the algorithmic approach to calculate improvement scenarios in Section 4.2. The algorithms in Sections 4.1 and 4.2 provide the information required for a student PSR. Section 4 will conclude with a weight selection model that provides feed-back to a lecturer, which is discussed in Section 4.3.

4.1 Linear model for p -mark calculation

To address the feed-back requirement, p -mark calculation is performed by means of a linear model according to the framework proposed in Figure 3. The linear model is based directly on the non-linear model (3)–(8) discussed in Section 3.1. It was converted to a linear model by omitting constraint set (5) which stipulated that the weights be in multiples of a number k . This constraint set can be excluded because no empirical calculations are required to find a solution. The model was implemented by means of Algorithm 1.

The linear model was solved for the data set in Appendix A. For the purpose of validating the model the p -marks were used for compiling a student ranking. This ranking correlated favourably with that of the *benchMark* program ($\rho = 0.96$) as well as with that of the non-linear model ($\rho = 0.97$). It may therefore be accepted that the proposed linear model produces acceptable results and may be used in the proposed framework of Figure 3 to address the feed-back requirement. The results of this model are attached in Appendix C and are used as input to the algorithm implemented to address the feed-forward and feed-up requirements, which is discussed in the next section.

4.2 Improvement target calculation and potential participation plan

At the start of a semester a lecturer typically plans a certain number of assessments to be taken by the students in each of the factors. As the semester progresses and assessments

Algorithm 1: Linear modelling of p -mark calculation

Input : $Y = \{\bar{y}_1, \dots, \bar{y}_m\}$ is an array representing the current factor averages for a group of m students in n factors.

Output: A set of arrays $P = \{\bar{p}_1, \dots, \bar{p}_m\}$ with the minimum and maximum p -marks, their associated weights, and the average p -marks for m students.

```

1 for  $i = 1$  to  $m$  do
2   read  $\bar{y}_i$  from  $Y$ ;
3   calculate minimum  $p$ -mark and associated weights for student  $i$ ;
4   calculate maximum  $p$ -mark and associated weights for student  $i$ ;
5   calculate average  $p$ -mark for student  $i$ ;
6   populate  $\bar{p}_i$  with the values calculated in lines 3–5;
7 end
8 output  $P$ ;
```

are given by the lecturer, the remaining assessments in each of the factors decrease. Feed-forward is therefore dependent on the point in time at which it is generated and must consider completed as well as remaining assessments in each factor. This time dependency means that different scenarios exist according to which students can complete a semester but with limited combinations of assessments available in each factor. An algorithmic approach was followed to determine the available scenarios at any specific point in time. The purpose of the algorithm is to provide a student with a future time perspective based on assessments planned for the remainder of the semester, the student's current factor averages and minimum p -mark. This perspective will provide students with insight into the effect that their continued participation or lack thereof, can have on their minimum p -marks.

The approach takes as input the current factor averages for a student at a specific time during the semester, the number and layout of the assessments remaining in each one of the appropriate factors and the required percentage improvement (δ). A table, based on truth table layout principles, is developed containing all the possible participation scenarios for the remainder of the semester. If there are assessment opportunities remaining, there would be a total of 2^s possible participation scenarios. The assessment plan is then used to eliminate participation scenarios that are impossible within the time constraints stipulated. This approach effectively caters for students engaging in, or missing out on any number of the remaining assessments. The algorithm provides output in the form of a set of participation scenarios based on the remaining possibilities and shows corresponding scores that a student must achieve in each scenario to affect the desired improvement in each factor. The projected change in the minimum p -mark is also shown for each scenario. The algorithmic approach is based on mathematical modelling principles and is shown below.

The algorithm starts by developing the participation scenarios in lines 1–4 and continues to calculate the scores required in upcoming assessments to produce the desired improvement

Algorithm 2: Potential participation plan for a student

Input : $\bar{y} = \{\bar{y}_1, \dots, \bar{y}_n\}$ is an array representing the current factor averages for a student in n factors, $A = \{\bar{a}_1, \dots, \bar{a}_s\}$ and array denoting the s remaining assessment sessions and δ the required percentage factor improvement.

Output: An array $\bar{x}_r$, with r possible improvement scenarios.

```

1 declare  $q(2^s, n)$ ; // to store all scenarios for  $s$  sessions
2 for  $c = 1$  to  $2^s$  do
3   | populate  $q$  with participation scenario  $c$ ;
4 end
5 remove duplicate rows in  $q$ ;
6  $r \leftarrow \text{number\_of\_rows}(q)$ ;
7 declare  $\text{marks}(r, n)$ ; // scores required for a required improvement of  $\delta$ 
8 declare  $\text{min\_f\_assessments}(n)$ ; // minimum assessments per factor to
   | cause a required improvement of  $\delta$ 
9 for  $j = 1$  to  $n$  do
10  | determine  $e_j$  from  $\bar{a}_j$ ; // remaining number of assessments in factor  $j$ 
11  | determine  $t_j$  from  $\bar{a}_j$ ; // 'total number of assessments in factor  $j$ 
12  | for  $u = 1$  to  $e_j$  do
13    | read  $y_u$  from  $\bar{y}$ ;
14    | calculate  $\text{newValue}$  according to equation (22) with  $y_u$ ,  $\delta$ , and  $\eta_u = u$ ;
15    | if  $q(j, u)$  is possible then
16      |  $\text{marks}(j, u) = \text{newValue}$ ;
17      | set  $\text{min\_f\_assessments}(j)$ ;
18    | else
19      |  $\text{marks}(j, u) = \text{"not possible"}$ ;
20    | end
21  | end
22 end
23 declare  $x(r, n+1)$ ; // for  $r$  scenarios,  $x$  has a required score per
   | factor and minimum change in  $p$ --mark
24  $\text{row} \leftarrow 1$ ;
25 while  $\text{row} \leq r$  do
26   | for  $\text{element} = 1$  to  $n$  do
27     | if  $q(\text{row}, \text{element}) \leq \text{min\_f\_assessments}(\text{element})$  then
28       | populate  $x(\text{row}, \text{element})$  using  $\text{marks}(\text{row}, \text{element})$ ;
29     | end
30   | end
31   |  $\text{row} \leftarrow \text{row} + 1$ ;
32 end
33 output  $\bar{x}_r$ ;

```

in line 14 according to

$$\text{marks}_j = \frac{\delta t_j + e_j y_{\alpha j}}{\eta_j} \quad (22)$$

where $marks_j$ is the score the student must achieve in the next η_j assessments of factor j , t_j is the total number of assessments in the semester for factor j , e_j is the remaining number of assessments for factor j and $y_{\alpha j}$ is the current factor average in factor j for student α . In lines 15–22, these scores are used to determine the minimum number of assessments to be completed to achieve an improvement of δ in the factors for which it will be possible. In lines 23–32 the possible scenarios are reduced further by comparing the number of assessments in the remaining scenarios to the minimum required in each factor.

For a number of s remaining assessment sessions, this approach can be illustrated according to Figure 4.

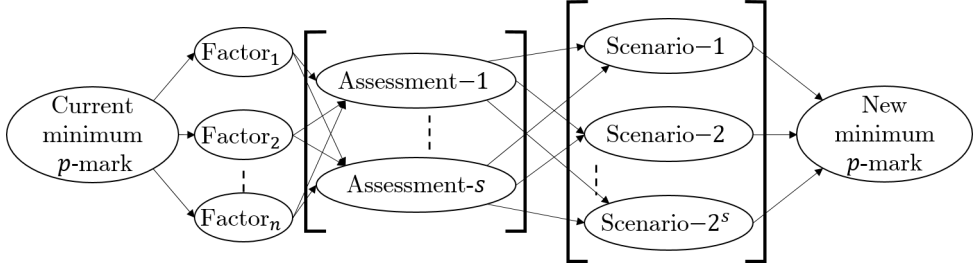


Figure 4: Expansion diagram for potential participation.

As an illustrative example, consider the student data set in Appendix A with four factors and 26 students and an example set of five remaining assessment sessions as shown in Table 6.

Assessment session				
1	2	3	4	5
Attendance (F_4)	Attendance (F_4)	Class test (F_2)	Attendance (F_4)	Attendance (F_4)
	Theoretical assignment (F_3)			Theoretical assignment (F_3)

Table 6: Example of a future assessment plan for the remainder of a semester.

In session 1, a student can participate in one attendance (F_4). In session 2, a student can participate in one attendance (F_4) and one theoretical assignment (F_3), and so forth. In this example, assessments can be taken in more than one factor per session and it is assumed that if a student participates in one session, all assessments given during that session are completed. By attending class in the second session for example, a student can obtain a mark for attendance (F_4) as well as a theoretical assignment (F_3). Assume that each factor is represented by a number that increases when a student participates in an assessment. In the example, a set of four numbers will then represent assessment participation in $F_1, \dots, F_4$ respectively. The possible participation options available for the assessment plan given in Table 6 were generated by means of a truth table in Algorithm 2 and is illustrated in Figure 5.

Each node represents a unique participation scenario for the semester, with self-loops indicating no further participation for the remaining assessment sessions. The set $\{0, 1, 0, 2\}$,

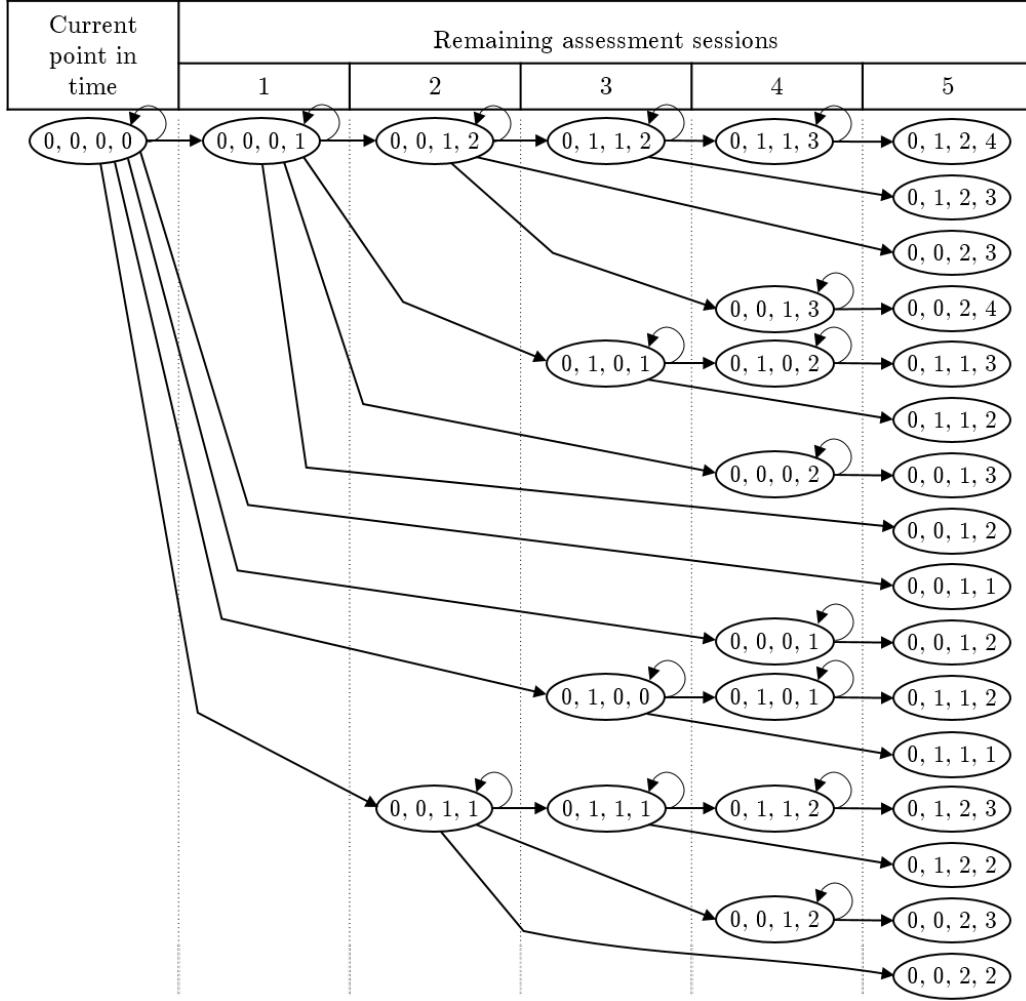


Figure 5: Possible participation scenarios for a student with the assessment plan as laid out in Table 6.

for example, represents a scenario in which the student will have completed only one assessment in F_2 and two assessments in F_4 at the close of the semester.

The total number of possible participation scenarios for this assessment plan is $2^5 = 32$. The algorithm was implemented for the assessment plan in Table 6, for Student 22. Table 7 shows a potential participation plan with an improvement target of $\delta = 5\%$.

The table shows the current factor averages and minimum p -mark for Student 22, followed by the scenarios possible for a 5% improvement in the relevant factors. Note that for attendance (F_4) a student can either get 100% for being in class or 0% for being absent. The scores for F_4 were, therefore, calculated as

$$marks_j = \frac{y_{\alpha j}(t_j - e_j) + 100\eta_j}{t_j}, \quad (23)$$

with similar variables as in (22). The layout of the assessments in Table 6 and their time

Improvement of a maximum of 5% on factor averages for Student 22					
Current performance status report:					
	F_1	F_2	F_3	F_4	Current minimum p -mark(%)
Factor averages(%):	40.00	54.07	42.67	88.89	45.48
Number of remaining assessments at current stage in the semester:	0	1	2	4	
Total number of assessments for the entire semester:	1	4	9	12	
Improvement plan:					
The scenarios for which improvement is possible in F_2 , F_3 and F_4	Scores (%) required for a 5% improvement:				Increase in minimum p -mark(%)
	F_1	F_2	F_3	F_4	
$Scenario_1 = \{0, 1, 2, 2\}$	0.00	74.07	65.17	75.92	2.78
$Scenario_2 = \{0, 1, 2, 3\}$	0.00	74.07	65.17	84.26	2.86
$Scenario_3 = \{0, 1, 2, 4\}$	0.00	74.07	65.17	92.59	2.96

Table 7: Improvement plan for Student 22 with an improvement of $\delta = 5\%$ on the current factor averages.

dependency were used to establish which scenarios can lead to an improvement of 5%. As the assessment plan included no more available assessments in F_1 , the improvement plan shows all the possible scenarios to produce an improvement of 5% on F_2 and F_3 .

Consider $Scenario_3$ in Table 7: to increase the current factor averages of F_2 and F_3 by 5%, the student will need to complete at least one assessment in F_2 with a score of 74.07% and at least two assessments in F_3 with an average score of 65.17%. The average for F_4 will increase to 92.59% if the student participates in all four assessments available. If all of these conditions are met, the increase in minimum participation mark will be 2.96%. Although this increase in minimum p -mark seems small, it is important to note that it is dependent on the required improvement and the point in time at which these scenarios are calculated. Early in the semester, the number of remaining assessments per factor will have a greater influence on the extent to which the minimum p -mark can change than at a later stage. Furthermore, due to the small value of δ the resulting change in minimum p -mark is also small. The algorithm was also implemented for the same student with $\delta = 10\%$, and resulted in the same three scenarios shown in Table 7. The maximum possible increase in minimum p -mark for $Scenario_3$ in this instance was 5.9%. Should a higher improvement percentage be required, it may be possible that the targets cannot be reached because the remaining number of assessments is not enough. In the example a required improvement of 15% under the same conditions, will be impossible.

An advantage of using this algorithm is that by setting $\delta = 0\%$ and consequently excluding

any improvement, it can be used to generate a projected report for a student who does not plan to participate in any number of further assessments. Students can build their factor averages up to a certain stage and then fail to participate in any further assessments for numerous reasons, for example illness. Providing them with a projected status record that shows how much their minimum p -marks can decrease, can provide the motivation for their continued participation. $Scenario_0 = \{0, 0, 0, 0\}$ was generated for the current factor averages of Student 22 and the assessment plan in Table 6 and showed that the student's minimum p -mark can decrease by 7% if there is no further participation in any further assessments. This scenario will have more serious consequences earlier in the semester when there are many assessments to be completed for each factor.

Algorithm 2 effectively addresses the feed-forward as well as feed-up requirements. Feed-up is provided in the form of the predicted increase or decrease in minimum p -mark. Feed-forward is presented in the form of target scores that a student must obtain to reach said change in the minimum p -mark.

Although the combination of Algorithms 1 and 2 effectively and efficiently addresses all three requirements for student PSRs, lecturers sometimes require additional information before deciding on the weights to use for final p -mark calculation. To assist a lecturer in deciding which weight combination to select for calculation of the final p -marks, a non-linear model was formulated and implemented in an iterative algorithm. This algorithm was developed to provide feed-back solely to the lecturer and is discussed next.

4.3 Non-linear programming model in a weight selection algorithm

The linear model in Section 4.1 produced not only the p -marks and subsequent student ranking, but also the weights used to calculate the maximum and minimum p -marks per student. A lecturer decides only at the close of a semester which factor weights to use for the calculation of the final p -marks. Generally, lecturers have certain guideline throughput objectives (*e.g.* a 70% pass rate or p -mark class average) according to which they can measure the success of a group of students. It will, however, be useful to know throughout the course of a module whether a certain objective is reachable with the marks available at a certain point in time. To this end, a non-linear model was formulated to determine the required weights that will produce a p -mark class average as close as possible to some predetermined target. The model is expressed in (24).

$$\text{Minimise } z = |p_t - p_a|, \quad (24)$$

subject to constraints (4) and (6)–(8) in Section 3.1. In this model, p_t represents the target p -mark class average, p_a is the current p -mark class average calculated with fixed weights as

$$p_a = \frac{1}{m} [w_1 y_{11} + \cdots + w_m y_{m1} + \cdots + w_1 y_{1n} + \cdots + w_m y_{mn}], \quad (25)$$

with m the number of students and n the number of factors. Due to the non-linearity of the model, it can be solved using a heuristic approach. As with the non-linear model in Section 3.1, the Evolutionary Solver in Excel was used. The approach for determining the weights by means of the model in (24) was implemented according to Algorithm 3.

Algorithm 3: Non-linear modelling of weight selection

Input : $Y = \{\bar{y}_1, \dots, \bar{y}_m\}$ is an array representing the current factor averages for m students in n factors, p_a the actual p -mark class average and p_t the target p -mark class average.

Output: $\bar{w} = \{w_1, \dots, w_n\}$ is an array with the weights required to reach p_t .

```

1 find  $\bar{w}$  by solving model (24) using  $Y$  and  $p_a$ ;
2 if ( $p_t$  was not reached with  $\bar{w}$ ) then
3   if (upper limits on  $\bar{w}$  is mandatory) then
4     | solve model (24) until no further improvement;
5   else
6     | relax upper limits on weights;
7     | find  $\bar{w}$  by solving model (24) using  $Y$  and  $p_a$ ;
8   end
9 end
10 output  $\bar{w}$ ;
```

To evaluate the model, it was implemented on the data set in Appendix A with a target p -mark class average of 70%. Initially the resulting set of weights that was produced by the algorithm was such that a p -mark class average of 70% could not be reached due to the upper limits imposed on the weights by constraint set (6) in Section 3.1. When the upper limit constraints on the weights are relaxed, the target p -mark class average of 70% can be obtained using the resulting weight set. In the example, the weights required to reach a p -mark class average of 70% were 0.66, 0.12, 0.11 and 0.11 for w_1 to w_4 respectively.

The model produces insight which a lecturer can use to determine how close to a specific target the entire class is. Although useful results can be obtained, they must be approached and interpreted with caution. Issues that need to be considered include:

- The model is solved heuristically which means that optimal solutions cannot be guaranteed. Although minor deviations may appear, empirical experiments have shown that the results were consistently acceptable.
- When constraints like upper limits are enforced on the weights, the target p -mark class average may not necessarily be reachable. In such an instance the model can be solved iteratively so that a solution as close as possible to the target p -mark class average can be reached.
- The weights resulting from an iterative approach may not conform to assurances made to students at the onset of the semester. A lecturer can for example, indicate at the start of a semester that no factor weight will exceed 0.4 but the algorithm may show that to reach a target p -mark class average of 70%, w_1 must be 0.66. In such a case the lecturer will either need to discuss the possibility of allowing the weight ranges to change with the students, or be content with a p -mark class average that is lower than the target.

Implementation of Algorithms 1 and 2 saw the feed-back, feed-up and feed-forward requirements for efficient PSRs being met. The model in Section 4.3 used the resulting

p -marks obtained in Section 4.1 as input to determine an estimate of the weights required to reach a class average as close as possible to a specific target.

5 Discussion

None of the existing models evaluated in Section 3 fully addressed all of the requirements for effective PSRs. In Section 4, three algorithms were discussed that not only address the requirements but also provide additional feed-back to a lecturer. The three algorithms are closely related because the feed-back is used to generate the feed-forward and feed-up which are included in a PSR. This is illustrated in Figure 6.

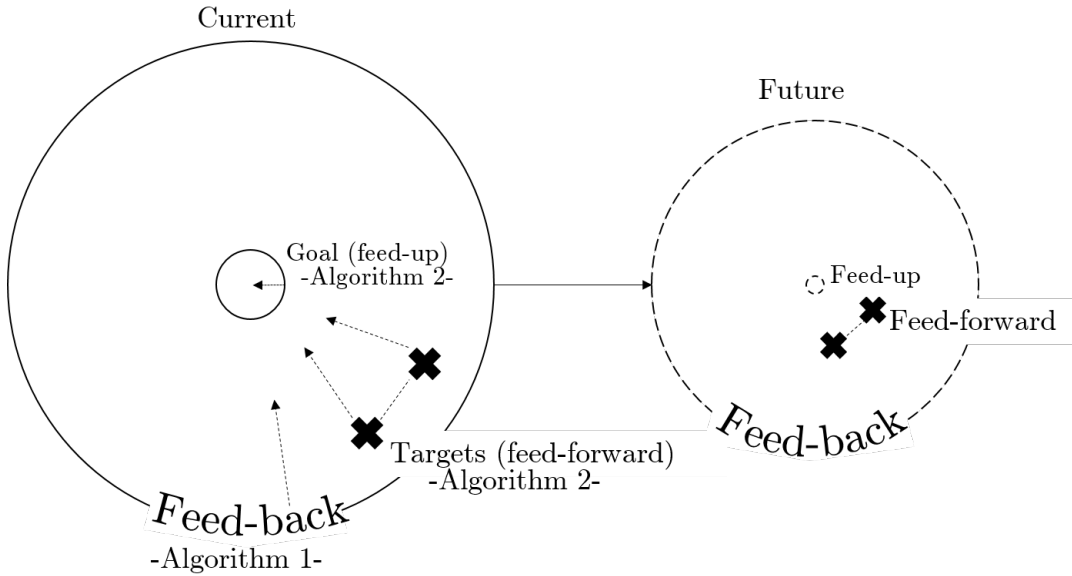


Figure 6: Algorithmic solution to effective PSRs.

In Figure 6, the arrows represent the passing of time and show how feed-back is required to set interim goals, or feed-up, as well as to calculate improvement targets as feed-forward. As time passes, the number of completed assessments grows and the number of remaining assessments decreases. Feed-back therefore progressively gets more accurate and closer to the final marks that will be used for examination admission. The decreasing size of the feed-back circle is therefore an indication of the decreasing number of remaining assessments. The interim goal(s) as well as the related improvement targets, are calculated based on completed and remaining assessments and therefore get stricter as time passes and the remaining assessments decrease. This means that closer to the end of a semester, certain improvements may not be possible to achieve within the remaining available assessments. The algorithmic solutions offer specific as well as general contributions towards PSRs in the academic environment. These are discussed below.

5.1 Specific remarks

The ranking produced by Algorithm 1 proves to be acceptable because of the high Spearman's ranking correlation between it and those obtained from *benchMark* and the non-linear model discussed in Section 3.1. The algorithm eliminates the need for iterative empirical calculations. Furthermore, the weights calculated by Algorithm 1 are considered more accurate than those obtained by *benchMark* because they are not restricted to be integer or in multiples of a predetermined number.

Algorithm 2 uses the minimum p -mark and its associated weights to develop a future time participation perspective for a student. The scenarios according to which the required improvement can be reached are determined and provided. The algorithm can be used to generate a participation plan at any point in time during the semester.

Algorithm 3 provides additional progress information to a lecturer which can be used to determine the weights required to reach a target p -mark class average. The calculated weights may also provide a lecturer with some insight into how successful students were in each factor. A specific factor weight which is much higher than all the others would for example indicate that the students overall performed much better in that factor than in all the others.

Although the proposed algorithmic solutions provide the information essential to PSRs, some disadvantages were identified and are listed accordingly. Algorithm 2 was developed to generate participation scenarios for an individual student. The algorithm can be used iteratively to generate improvement plans for an entire class. However, depending on the marks, the same improvement percentage will not be possible for all students. Therefore every time the lecturer generates an improvement plan for a student, an improvement percentage related to the student's current performance is required. This makes execution of the algorithm for an entire class rather labour intensive.

If the upper limit weight constraints in Algorithm 3 need to be relaxed because a target p -mark class average cannot be reached, the resulting weights may not conform to the guidelines set by the lecturer at the start of the semester.

5.2 General remarks

The purpose of the proposed algorithmic solutions was to address the demand for a system that will continuously provide students with academic performance status reports in a module, at any point in time during a semester. The discussed algorithmic solutions address the feed-back, feed-forward and feed-up requirements and furthermore exhibit the attributes necessary for it to be constructive and to motivate students to improve.

The proposed framework provides ongoing feed-back that is timely because it can be used as frequently as required. The results are readily available and can be made available to students instantly. The approach is user-friendly and consistently uses an updated set of the students' current marks to obtain the information needed for feed-back. Furthermore, the suggested approach provides a student with personalised and goal-referenced feed-up which is based on the student's current performance. The feed-forward improvement targets are presented in the form of actionable scores that a student needs to acquire in a

specified number of upcoming assessments, in each factor.

The collective algorithmic approach sufficiently generates feed-back, feed-up and feed-forward for academic PSRs based on a set of students' current performance. Implementation of the framework could also be useful in environments where PSRs are required for motivation to participate in specific learning opportunities or assessments or also, for example, in sports activities or tournaments.

6 Conclusions

The purpose of this study was to evaluate existing mathematical models against the requirements for effective PSRs and develop an improved algorithmic approach for that purpose. An example data set containing the marks for 26 students in four factors (Appendix A) was used throughout the study to evaluate the methods and solutions.

Three existing models were evaluated. The *benchMark* program was assessed and formalised in a non-linear mathematical model. Although these options did address feed-back in PSRs, other disadvantages made them impractical to use regularly and under certain conditions. An outputs-only DEA was performed to group students into classes of similar efficiency. Improvement targets (feed-forward) were then calculated to help students advance to one class higher. The interim goal (feed-up) for each student was not clear because it was not linked to the p -mark. The method was independent of weight constraints and impractical to use for classes with many students. The analytic hierarchy process provided a means to rank students without calculating their p -marks but did not address the feed-up or feed-forward requirements.

To address the deficiencies, a set of related algorithmic approaches in which alternative mathematical models are implemented, was developed. The reports generated were consequently assessed in terms of the required attributes for effective PSRs. Three algorithms were developed to attend to the requirements for PSRs. Firstly, a linear programming model was implemented in an algorithm to address the requirements of feed-back in the form of a student ranking. The next algorithm generated a future participation and improvement plan for each student and addressed the feed-up and feed-forward requirements for PSRs. A non-linear model was implemented in a third algorithm to provide additional feed-back to a lecturer during the course of a semester.

Some aspects that are currently receiving attention to improve the proposed algorithmic approach include the calculation of the maximum possible improvement per individual student in a class and automatic generation of the future perspective plans; and development of an interactive user-friendly graphical interface that combines the different algorithmic activities.

References

- [1] ALEXANDER M, 2012, *Decision-making using the analytic hierarchy process (AHP) and SAS/IML (R)*, Paper presented at the 20th Annual SouthEast SAS Users Group (SESUG), Durham (NC).
- [2] ANIS A & ISLAM R, 2015, *The application of analytic hierarchy process in higher-learning insti-*

- tutions: A literature review, *Journal of International Business and Entrepreneurship Development*, **8(2)**, pp. 166–182.
- [3] BAKER KR, 2011, *Optimization modeling with spreadsheets*, 2nd Edition, John Wiley & Sons, Hoboken (NJ).
- [4] BARKER LJ & GARVIN-DOXAS K, 2004, *Making visible the behaviors that influence learning environment: A qualitative exploration of computer science classrooms*, *Computer Science Education*, **14(2)**, pp. 119–145.
- [5] CHU WJ, 2014, *An application to academic rankings of new evaluation algorithm*, *Applied Mechanics and Materials*, **651-653**, pp. 2112–2115.
- [6] CSATÓ L, 2015, *A graph interpretation of the least squares ranking method*, *Social Choice and Welfare*, **44(1)**, pp. 51–69.
- [7] DANIEL WW, 1990, *Spearman rank correlation coefficient*, pp. 358–365 in *Applied nonparametric statistics*, PWS-Kent, Boston.
- [8] DIETZ-UHLER B & HURN JE, 2013, *Using learning analytics to predict (and improve) student success: A faculty perspective*, *Journal of Interactive Online Learning*, **12(1)**, pp. 17–26.
- [9] DOBROTA M, BULAJIC M, BORNMAN L & JEREMIC V, 2016, *A new approach to the QS university ranking using the composite I-distance indicator: Uncertainty and sensitivity analyses*, *Journal of the Association for Information Science and Technology*, **67(1)**, pp. 200–211.
- [10] DONEGAN HA & DODD FJ, 1991, *A note on Saaty's random indexes*, *Mathematical and Computer Modelling*, **15(10)**, pp. 135–137.
- [11] DU TOIT JV, 2015, *Using participation marks to manage, motivate and inform on academic progress*, Paper presented at the Conference for Outstanding Teaching/Learning and Innovative Technology Use, Potchefstroom.
- [12] EDUCONNECT, 2018, *Calculate your APS at North West University (NWU)*, [Online], [Cited March 19th, 2018], Available from <https://educonnect.co.za/how-to-calculate-your-aps-at-nwu/>
- [13] ELIAS H, MUSTAFA SMS, ROSLAN S & NOAH SM, 2011, *Motivational predictors of academic performance in end year examination*, *Procedia-Social and Behavioral Sciences*, **29**, pp. 1179–1188.
- [14] ELSNER B & ISPHORDING IE, 2018, *Rank, sex, drugs, and crime*, *Journal of Human Resources*, **53(2)**, pp. 356–381.
- [15] FARLEY AA & RAMSAY AL, 1988, *Student performance in first year tertiary accounting courses and its relationship to secondary accounting education*, *Accounting & Finance*, **28(1)**, pp. 29–44.
- [16] GIANNOULIS C & ISHIZAKA A, 2010, *A web-based decision support system with ELECTRE III for a personalised ranking of British universities*, *Decision Support Systems*, **48(3)**, pp. 488–497.
- [17] GREER JE, FROST S, BANOW R, THOMPSON C, KULEZA S, WILSON K & KOEHN G, 2015, *The student advice recommender agent: SARA*, *Proceedings of the 5th Workshop on Personalization Approaches for Learning Environments*, Dublin, pp. 1–8.
- [18] HATTIE J & TIMPERLEY H, 2007, *The power of feedback*, *Review of Educational Research*, **77(1)**, pp. 81–112.
- [19] HU Z, ZHOU J, ZHANG M & ZHAO Y, 2015, *Methods for ranking college sports coaches based on data envelopment analysis and pagerank*, *Expert Systems*, **32(6)**, pp. 652–673.
- [20] ISHIZAKA A, 2008, *A multicriteria approach with AHP and clusters for supplier selection*, Paper presented at the 15th International Annual EurOMA Conference, Groningen.

- [21] KAMVAR S, HAVELIWALA T & GOLUB G, 2004, *Adaptive methods for the computation of pagerank*, Linear Algebra and its Applications, **386**, pp. 51–65.
- [22] KAO C & LIN PH, 2008, *Class ranking of the managment colleges in Taiwan*, Lecture Notes in Management Science, **1**, pp. 129–140.
- [23] KENT G, KRUGER HA & DU TOIT JV, 2016, *Class ranking of secondary schools in the North West province of South Africa*, ORiON, **32(2)**, pp. 123–146.
- [24] LANDAU R, MULDER G, HOLMES R, BORINSKAYA S, KANG N & BORDEIANU C, 2014, *Instances: incorporating computational scientific thinking advances into education and science courses*, Concurrency and Computation: Practice and Experience, **26(13)**, pp. 2316–2328.
- [25] LOURENS A & BLEAZARD D, 2016, *Applying predictive analytics in identifying students at risk: A case study*, South African Journal of Higher Education, **30(2)**, pp. 129–142.
- [26] MAVRIKIS M, GUTIERREZ-SANTOS S & POULOVASSILIS A, 2016, *Design and evaluation of teacher assistance tools for exploratory learning environments*, Proceedings of the 6th International Conference on Learning Analytics & Knowledge, Edinburgh, pp. 168–172.
- [27] MCKAY T, MILLER K & TRITZ J, 2012, *What to do with actionable intelligence: E 2 Coach as an intervention engine*, Proceedings of the 2nd International Conference on Learning Analytics & Knowledge, Vancouver, pp. 88–91.
- [28] MERTZ J, 2015, *Ranking the greatest NBA players: An analytics analysis*, Honours Thesis, Ball State University, Muncie.
- [29] MOORE JH & WEATHERFORD LR, 2001, *Decision modeling with Microsoft Excel*, 6th Edition, Prentice Hall, Upper Saddle River (NJ).
- [30] NOCEDAL J & WRIGHT SJ, 2006, *Numerical optimization (Springer series in operational research)*, 2nd Edition, Springer, New York (NY).
- [31] PICCIANO AG, 2014, *Big data and learning analytics in blended learning environments: Benefits and concerns*, International Journal of Artificial Intelligence and Interactive Multimedia: Special issue on multisensor user tracking and analytics to improve education and other application fields, **2(7)**, pp. 35–43.
- [32] POMEROL JC & BARBA-ROMERO S, 2000, *Multicriterion decision in management: Principles and practice*, Kluwer Academic Publishers, New York (NY).
- [33] POWELL SG & BAKER KR, 2009, *Management science: The art of modeling with spreadsheets*, 4th Edition, John Wiley & Sons, New York (NY).
- [34] RAGSDALE CT, 2007, *Managerial decision modelling*, 1st Edition, Thomson Higher Education, Mason (OH).
- [35] ROFF S, CHANDRATILAKE M, MCALEER S & GIBSON J, 2012, *Medical student rankings of proposed sanction for unprofessional behaviours relating to academic integrity: Results from a Scottish medical school*, Scottish Medical Journal, **57(2)**, pp. 76–79.
- [36] SAATY TL, 1980, *The analytic hierarchy process: Planning, priority setting, resource allocation*, McGraw-Hill.
- [37] SADLER PM & GOOD E, 2006, *The impact of self-and peer-grading on student learning*, Educational Assessment, **11(1)**, pp. 1–31.
- [38] SEIFORD LM & ZHU J, 2003, *Context-dependent data envelopment analysis - measuring attractiveness and progress*, The International Journal of Management Science, **31(5)**, pp. 397–408.

- [39] SIEMENS G & LONG P, 2011, *Penetrating the fog: Analytics in learning and education*, EDUCAUSE Review, **46(5)**, pp. 30–32.
- [40] STANTCHEV V, COLOMO-PALACIOS R, SOTO-ACOSTA P & MISRA S, 2014, *Learning management systems and cloud file hosting services: A study on students' acceptance*, Computers in Human Behavior, **31**, pp. 612–619.
- [41] THANASSOULIS E, 2001, *Introduction to the theory and application of data envelopment analysis: A foundation text with integrated software*, Kluwer Academic Publishers, Norwell (MA).
- [42] TINCANI MM, 2017, *Heterogeneous peer effects and rank concerns: Theory and evidence*, (Unpublished) CESifo Working Paper 6331.
- [43] VAN DER MERWE A, DU TOIT JV & KRUGER HA, 2016, *Student ranking by means of non-linear mathematical optimization of participation marks*, Lecture Notes in Management Science, **8**, pp. 113–119.
- [44] VAN DER MERWE A, KRUGER HA & DU TOIT JV, 2016, *A mathematical ranking model in learning analytics*, Proceedings of the 16th International Conference on Computational and Mathematical Methods in Science and Engineering, Rota (Spain), pp. 1525–1535.
- [45] WIGGINS G, 2012, *7 keys to effective feedback*, Educational Leadership, **70(1)**, pp. 10–16.

Appendices

A. Data set used in the mathematical models

Table 8 presents the data set containing the factor averages of 26 students in four factors, where the factors (denoted F_1 to F_4 respectively) are a semester test, class tests, theoretical 14 assignments and attendance. This set was used in the paper to evaluate the relevant ranking methods.

Student	F_1 (%)	F_2 (%)	F_3 (%)	F_4 (%)
1	73.33	64.74	13.92	62.50
2	100.00	95.24	97.92	100.00
3	96.67	82.62	75.05	100.00
4	35.00	35.27	21.01	77.78
5	61.67	56.90	40.59	77.78
6	75.00	63.10	57.59	88.89
7	58.33	22.39	34.68	100.00
8	58.33	43.76	53.53	87.50
9	93.33	91.51	87.75	100.00
10	93.33	68.15	74.07	100.00
11	88.33	53.38	64.91	100.00
12	73.33	57.01	63.71	100.00
13	76.67	87.13	65.83	88.89
14	10.00	44.40	52.97	77.78
15	83.33	57.27	23.65	88.89
16	51.67	46.19	26.36	77.78
17	90.00	58.15	68.42	77.78
18	63.33	58.30	42.67	100.00
19	73.33	63.38	41.01	100.00
20	60.00	55.24	51.58	88.89
21	85.00	59.46	35.90	75.00
22	40.00	54.07	42.67	88.89
23	85.00	72.14	61.85	100.00
24	90.00	64.76	76.67	100.00
25	83.33	60.46	26.07	66.67
26	65.00	55.48	24.53	100.00

Table 8: Data set utilised in the mathematical models in this study.

B. Participation profiles of the 26 participants as calculated by *benchMark*

Participation profiles of the 26 participants as calculated by *benchMark*.

Student	F_1	F_2	F_3	F_4	Max	Min	Avg	Rank
2	100.00	95.24	97.92	100.00	98.29	97.58	97.85	1
9	93.33	91.51	87.75	100.00	93.15	91.35	92.33	2
3	96.67	82.62	75.05	100.00	89.22	85.43	87.44	3
13	76.67	87.13	65.83	89.00	83.35	78.25	81.05	4
10	93.33	68.15	74.07	100.00	84.89	78.78	81.04	5
24	90.00	64.76	76.67	100.00	82.86	77.07	79.37	6
23	85.00	72.14	61.85	100.00	79.75	74.31	77.35	7
17	90.00	58.15	68.42	77.78	76.02	71.62	73.66	8
11	88.33	53.38	64.91	100.00	76.66	69.08	72.28	9
12	73.33	57.01	63.71	100.00	73.51	65.73	68.79	10
6	75.00	63.10	57.59	88.89	70.80	66.24	68.60	11
19	73.33	63.38	41.01	100.00	69.43	61.49	66.24	12
21	85.00	59.46	35.90	75.00	68.52	60.65	63.67	13
25	83.33	60.46	26.07	66.67	67.63	57.73	61.88	14
15	83.33	57.27	23.65	88.89	67.36	56.52	61.80	15
18	63.33	58.30	42.67	100.00	66.08	57.21	61.75	16
20	60.00	55.24	51.58	88.89	63.58	57.18	59.97	17
5	61.67	56.90	40.59	77.78	61.67	54.97	58.73	18
26	65.00	55.48	24.53	100.00	61.25	51.28	57.37	19
1	73.33	64.74	13.92	62.50	63.76	52.40	57.35	20
8	58.33	43.76	53.53	87.50	60.33	52.67	55.53	21
22	40.00	54.07	42.67	88.89	56.06	47.97	51.61	22
16	51.67	46.19	26.36	77.78	49.81	43.33	47.25	23
7	58.33	22.39	34.68	100.00	47.60	38.88	42.50	24
14	10.00	44.40	52.97	77.78	45.59	33.46	39.69	25
4	35.00	35.27	21.01	77.78	41.57	32.90	37.16	26

Table 9: Participation profiles created by *benchMark* in Section 3.1, for the 26 students. $F_1 - F_4$ are the factor averages arranged according to weight priority. The columns Max, Min and Avg, are the maximum, the minimum and the average participation marks, respectively. The column Rank is the student ranking according to average participation marks. All marks are indicated in percentages.

The maximum, minimum and average p -marks, student ranking, and corresponding weights for calculating the minimum and maximum p -marks for 26 students using the non-linear model in Section 3.1.

Student	p -marks (%)			Rank	Weights for minimum p -mark				Weights for maximum p -mark			
	Max	Min	Average		w_1	w_2	w_3	w_4	w_1	w_2	w_3	w_4
2	98.63	97.78	98.21	1	0.40	0.40	0.15	0.05	0.40	0.20	0.20	0.20
9	93.15	91.44	92.30	2	0.35	0.30	0.30	0.05	0.25	0.25	0.25	0.25
3	90.20	86.13	88.17	3	0.35	0.30	0.30	0.05	0.40	0.20	0.20	0.20
10	84.52	80.04	82.28	4	0.35	0.35	0.25	0.05	0.35	0.25	0.20	0.20
24	84.27	78.33	81.31	5	0.35	0.35	0.25	0.05	0.40	0.20	0.20	0.20
13	80.54	77.17	78.29	6	0.35	0.30	0.30	0.05	0.35	0.35	0.15	0.15
23	80.80	74.95	76.90	7	0.35	0.30	0.30	0.05	0.40	0.20	0.20	0.20
11	78.99	70.83	73.55	8	0.35	0.35	0.25	0.05	0.40	0.20	0.20	0.20
17	76.87	72.75	74.12	9	0.30	0.30	0.30	0.10	0.40	0.20	0.20	0.20
12	73.51	66.55	68.87	10	0.35	0.35	0.25	0.05	0.25	0.25	0.25	0.25
6	71.92	66.90	68.57	11	0.35	0.30	0.30	0.05	0.40	0.20	0.20	0.20
19	70.21	61.98	64.73	12	0.35	0.30	0.30	0.05	0.40	0.20	0.20	0.20
21	68.47	61.61	63.90	13	0.30	0.30	0.30	0.10	0.40	0.30	0.15	0.15
15	67.40	59.07	61.85	14	0.35	0.30	0.30	0.05	0.40	0.30	0.15	0.15
18	66.08	57.46	60.33	15	0.35	0.30	0.30	0.05	0.25	0.25	0.25	0.25
25	63.97	59.13	60.75	16	0.30	0.30	0.30	0.10	0.40	0.30	0.15	0.15
20	63.93	57.49	59.64	17	0.35	0.30	0.30	0.05	0.25	0.25	0.25	0.25
1	62.87	51.85	55.52	18	0.30	0.30	0.30	0.10	0.40	0.35	0.15	0.10
5	59.72	54.72	56.39	19	0.35	0.30	0.30	0.05	0.40	0.20	0.20	0.20
8	60.78	53.24	55.75	20	0.35	0.35	0.25	0.05	0.25	0.25	0.25	0.25
26	62.00	51.75	55.17	21	0.35	0.30	0.35	0.05	0.4	0.20	0.20	0.20
22	56.41	47.33	50.36	22	0.40	0.30	0.25	0.05	0.25	0.25	0.25	0.25
7	54.75	41.93	46.20	23	0.35	0.35	0.25	0.05	0.25	0.25	0.25	0.25
16	50.73	43.74	46.07	24	0.35	0.30	0.30	0.05	0.40	0.2	0.20	0.20
14	46.29	33.60	37.83	25	0.40	0.35	0.20	0.05	0.25	0.25	0.25	0.25
4	42.26	33.02	36.10	26	0.40	0.30	0.25	0.05	0.25	0.25	0.25	0.25

Table 10: Resulting maximum, minimum, and average p -marks, ranking and weights calculated by means of the non-linear model in Section 3.1.

The student ranking of the 26 students, as determined by the AHP discussed in Section 3.3.

Student	Priority Vector				Composite Rating $r_{\alpha j}$	Pairwise Ranking
	F_1	F_2	F_3	F_4		
2	0.1019027	0.1226755	0.1458670	0.0612267	11.082600	1
9	0.0793605	0.1101616	0.1166742	0.0612267	9.334351	2
3	0.0872195	0.0810881	0.0704773	0.0612267	7.813273	3
10	0.0793605	0.0506373	0.0643211	0.0612267	6.501559	4
24	0.0614319	0.0370232	0.0824396	0.0612267	5.828006	5
23	0.0491764	0.0670063	0.0387908	0.0612267	5.425578	6
13	0.0326883	0.0977652	0.0460599	0.0283514	5.423514	7
17	0.0614319	0.0280258	0.0869943	0.0148856	4.954059	8
11	0.0541918	0.0172421	0.0495889	0.0612267	4.324156	9
15	0.0568494	0.0347459	0.0065793	0.0283514	3.588964	10
12	0.0260657	0.0276756	0.0418947	0.0612267	3.498863	11
6	0.0306510	0.0374787	0.0359753	0.0283514	3.341923	12
19	0.0260657	0.0368469	0.0181735	0.0612267	3.299577	13
25	0.0568494	0.0312238	0.0097345	0.0093700	3.261681	14
21	0.0491764	0.0297609	0.0133239	0.0127000	3.070979	15
18	0.0171022	0.0287135	0.0199034	0.0612267	2.776450	16
1	0.0260657	0.0377650	0.0051790	0.0087337	2.279836	17
26	0.0190688	0.0175284	0.0075953	0.0612267	2.263566	18
20	0.0147006	0.0158600	0.0259862	0.0283514	1.935315	19
5	0.0162776	0.0262947	0.0150281	0.0148856	1.882403	20
7	0.0128273	0.0058818	0.0127885	0.0612267	1.799583	21
8	0.0128273	0.0103791	0.0252893	0.0182904	1.540473	22
22	0.0074831	0.0158520	0.0177094	0.0283514	1.516930	23
14	0.0043365	0.0113759	0.0277737	0.0148856	1.271811	24
16	0.0104266	0.0141118	0.0089887	0.0148856	1.191343	25
4	0.0064634	0.0068808	0.0068640	0.0148856	0.793206	26

Table 11: Ranking of the 26 students produced by the AHP in Section 3.3.

C. Results of the linear model in Section 4.1

The maximum, minimum and average p -marks, student ranking, and corresponding weights for calculating the minimum and maximum p -marks for the 26 students using the linear model in Section 4.1.

Student	p -marks (%)			Rank	Weights for minimum p -mark				Weights for maximum p -mark			
	Max	Min	Average		w_1	w_2	w_3	w_4	w_1	w_2	w_3	w_4
2	98.63	97.70	98.17	1	0.40	0.40	0.19	0.01	0.40	0.20	0.20	0.20
9	93.19	90.96	92.07	2	0.33	0.33	0.33	0.01	0.40	0.20	0.20	0.20
3	90.20	84.93	87.57	3	0.33	0.33	0.33	0.01	0.40	0.20	0.20	0.20
10	85.78	78.73	82.26	4	0.33	0.33	0.33	0.01	0.40	0.20	0.20	0.20
24	84.29	77.37	80.83	5	0.33	0.33	0.33	0.01	0.40	0.20	0.20	0.20
13	80.99	76.67	78.83	6	0.33	0.33	0.33	0.01	0.40	0.40	0.10	0.10
23	80.80	73.27	77.03	7	0.33	0.33	0.33	0.01	0.40	0.20	0.20	0.20
17	76.87	72.25	74.56	8	0.33	0.33	0.33	0.01	0.40	0.20	0.20	0.20
11	78.99	69.19	74.09	9	0.33	0.33	0.33	0.01	0.40	0.20	0.20	0.20
12	73.51	65.04	69.28	10	0.33	0.33	0.33	0.01	0.25	0.25	0.25	0.25
6	71.92	65.47	68.69	11	0.33	0.33	0.33	0.01	0.40	0.20	0.20	0.20
19	70.21	59.65	64.93	12	0.33	0.33	0.33	0.01	0.40	0.20	0.20	0.20
21	68.88	60.27	64.57	13	0.33	0.33	0.33	0.01	0.40	0.40	0.10	0.10
25	66.79	56.72	61.76	14	0.33	0.33	0.33	0.01	0.40	0.40	0.10	0.10
15	67.50	55.09	61.29	15	0.33	0.33	0.33	0.01	0.40	0.40	0.10	0.10
18	66.08	55.22	60.65	16	0.33	0.33	0.33	0.01	0.25	0.25	0.25	0.25
1	66.36	54.15	60.26	17	0.33	0.33	0.33	0.01	0.40	0.40	0.10	0.10
20	63.93	55.94	59.93	18	0.33	0.33	0.33	0.01	0.25	0.25	0.25	0.25
5	59.72	53.30	56.51	19	0.33	0.33	0.33	0.01	0.40	0.20	0.20	0.20
8	60.78	51.88	56.33	20	0.40	0.40	0.19	0.01	0.25	0.25	0.25	0.25
26	62.00	48.85	55.43	21	0.33	0.33	0.33	0.01	0.40	0.20	0.20	0.20
22	56.41	45.48	50.95	22	0.40	0.30	0.29	0.01	0.25	0.25	0.25	0.25
7	54.75	39.09	46.92	23	0.33	0.33	0.33	0.01	0.40	0.20	0.20	0.20
16	50.73	41.77	46.25	24	0.33	0.33	0.33	0.01	0.40	0.20	0.20	0.20
14	46.29	32.60	39.45	25	0.40	0.40	0.19	0.01	0.25	0.25	0.25	0.25
4	42.26	30.90	36.58	26	0.33	0.33	0.33	0.01	0.25	0.25	0.25	0.25

Table 12: Resulting maximum, minimum, and average p -marks, ranking and weights calculated by means of the linear model in Section 4.1.