

Introducing Optimization Techniques to Students: An Exam Case Distribution Model

Mario Vanhoucke

Faculty of Economics and Business Administration, University Ghent, 9000 Ghent, Belgium, mario.vanhoucke@ugent.be

In this paper, an integer programming (IP) model is presented to assign MBA and undergraduate students to groups to solve an exam case in an operations research (O.R.) course. It is assumed that the students have a basic understanding of mathematical programming and are now ready to build their first real-life model in class. Thanks to the direct link with the student's situation and the immediate repercussion on the exam assignment, students can quickly understand the problem and are willing to help to define the problem in class. The example illustrates many O.R.-related issues, such as the balance between problem complexity and solution quality, and the need for dynamic rather than static models. Thanks to its simplicity and practicality, this exercise is an ideal tool to make the often complex domain of O.R. more accessible.

Key words: exam assignment problem; operations research teaching

History: Received: September 2008; accepted: November 2009.

1. Introduction

Various sources in the literature mention that teaching O.R. to undergraduates and MBA students is not an easy task. Teachers can easily overwhelm students by minimum cost flow or traveling salesman problem procedures, by dynamic programming principles, or any other useful tool, often without succeeding in convincing people of the usefulness of these models. The column "Issues in Education," published in *OR/MS Today*, illustrates the complexity of the O.R. teaching process and provides useful hints and tricks about how to actually reach your public when teaching O.R.

The literature on teaching O.R. to undergraduate and MBA students is rich and widespread. Although this paper has no intention of summarizing the bulk of material available in the literature, it mentions two valuable publication streams that might be interesting to an O.R. teacher. A first set of publications spends effort on the choice of example models taken from business applications when teaching O.R. and using their model formulations and solution techniques. Selecting the business examples that are attractive and easily understandable to an MBA audience with an often diverse background is a key parameter for the success of an O.R. course. Guidelines are already given in early publications such as Grenander (1965) and others. More recent publications are also available in the public domain (see, e.g., Carraway and Clyman 1997, Miser 2000, Borsting et al. 1988,

Strasser and Ozgur 1995, Chen 1981, Kao et al. 1997). Second, another stream of publications focuses on the teaching method rather than on the applications and spends a lot of attention on the use of, e.g., spreadsheets and more sophisticated computer software, as well as the applicability of, e.g., independent learning or case teaching methods in O.R. courses. These topics have already been discussed decades ago by various authors (see e.g., Grenander 1965 and James 1988, among others). The most recent publications that present fresh and novel ideas in O.R. teaching methods are by Belton and Scott (1998), Böcker (1987), Bodily (1996), Franz (1989), Lasdon and Liebman (1998), Powell (1995, 1998, 1997), Tingley (1987), Liberatore and Nydick (1998), Liebman (1994, 1998), Winston (1996), Zahedi (1985), Tavares (1994), Evans (1992), Aggarwal (1978), Corner and Corner (2003), Robinson et al. (2003), Corner (1997), and Eaves (1997).

This paper presents a simple problem formulation for undergraduate and MBA students. The model is fairly easy to understand, but has a direct impact on the students and the whole class. Moreover, the problem formulation is general, leaving it open for small modifications to practical problems from the service sector or from industry. The experience is that when students need to build a real model that builds a solution that immediately affects their own current situation, they actively participate in the model formulation development and easily see opportunities

for extensions to other environments or even their own business. The model discussed in this paper is used in a teaching session to promote student activity and collaboration. The concept of authentic and active learning tasks has been widely discussed in literature, outside the field of O.R. A complete and general overview of research results, advantages, and disadvantages is outside the scope of this paper. However, a short reference and comparison with the proposed model and teaching method is given in §2 of this manuscript.

The presentation of this paper is as follows. Section 2 gives a short description of the problem and its relevance for pedagogical purposes and gives some background information about the types of students. Section 3 presents a mathematical model formulation for the assignment of cases to students, which is illustrated with a small example. Section 4 gives some potential topics that can be used to open a discussion in class and highlights some repeating questions/concerns/remarks that frequently come up during the teaching session. It also gives some notions of my own experience when using the model in classes. Section 5 provides conclusions.

2. Background Information

This section gives a brief overview of the problem description for which a mathematical model will be built, the teaching method used in an single three-hour class session, and the type of students participating in the course.

2.1. Problem Description

Students in O.R. courses often cannot see the value of optimization, because problem descriptions are too trivial or too complex; both are far from their own brief business experience. In my course, students must analyze a case study for their final exam. This case is analyzed in groups, and each group has a different case. For a problem that is challenging and applicable to students, I use the final case assignment as an example of an integer-programming modeling problem. In one of the first sessions, I give the students short abstracts of the exam cases. Based on these abstracts, they rank order the cases by preference, as well as their level of knowledge of O.R. This indication of knowledge level is optional and is particularly useful with diverse groups such as MBA students. The first two- or three-class sessions are on modeling and optimization. The problem formulation of this manuscript requires a basic understanding of linear programming (LP) and integer programming (IP).

The model is formulated and built in a software tool AIMMS (<http://www.aimms.com>) in a three-hour

session. The session closes with the final distribution of the cases to the students according to the solution found. The literature has already discussed allocation of students to groups, taking various constraints and objectives into account. Results are published in Mingers and O'Brien (1995), Beheshtian-Ardekani and Mahmood (1986), Donohue and Fox (1993), Miyaji et al. (1987), and Muller (1989). The papers by Baker and Powell (2002), Sampson et al. (1995), and Stallaert (1997) even refer to alternative methods for assigning students to groups in a class environment.

2.2. Active Learning

The introduction of this paper showed numerous papers about the O.R. teaching process, which seeks to convince a diverse audience of MBA students about the ease and relevance of O.R. tools and techniques. The current paper presents a simple and easily understandable O.R. model used in an active and collaborative way with immediate results and effect to the audience.

Active learning has received considerable attention in literature outside O.R. The pedagogical approach for this example can be classified as an active learning process, for the following reasons:

1. *Active learning* requires student activity and engagement during the learning process. Because the problem formulation of the current paper is built up through class discussions, it requires student activity from start until end. The engagement is promoted by the model's relevance for every student and its immediate impact on their course evaluation.

2. *Collaborative learning* refers to teaching methods with student interactions working towards a common goal. During model formulation, interactions between students are crucial (§4.1). Through discussions and careful choice of parameters, constraints, and objectives with often conflicting goals, this example seeks to convince students that a model is an abstraction, a simplification, and a reflection of the real-life problem. The common goal aspect is embedded in the nature of the proposed problem, which is to assign every student to the best group to solve the final exam case. The concept of "best" is reflected in the objectives and constraints of the formulation, which contains, among others, elements of fairness between the students, the need for the right balance of student background and knowledge per group, and the maximization of student's preferences.

For an overview and critical review of collaborative and active learning methods, see Prince (2004).

2.3. Student Population

The exercise is given in various O.R. courses at the Vlerick Leuven Gent Management School and the Ghent University in Belgium. The full-time MBA students at Vlerick Leuven Gent Management School consists of a group that varies between 45 and 55

international students (60% male, 40% female) with an average age of 30 years and a minimum required job experience of three years. The master's students at Vlerick Leuven Gent Management School have no work experience, but had finished their undergraduate studies one year earlier. Group size varies between 55 to 65 students (approximately 60% male, 40% female), with an average age of 23 years. The groups at Ghent University are undergraduate students following economics-business engineering, with a major in operations management. Their group size varies, and the average age of the students is 22 years.

Although the course outline differs among groups, the general purpose of the course is threefold and can be summarized along the following lines. First, students need to get acquainted with the terminology and techniques of O.R., including LP, IP, and nonlinear programming. They are expected to understand the underlying assumptions of each approach and to build (small) models on their own, using often simple and easily accessible software tools such as the Microsoft Excel Solver or any other student-friendly tool. Although many problem formulations can be easily modeled by student-friendly tools such as Solver, it is shown that more sophisticated tools (see, e.g., the AIMMS Figure 3 in §3.2) allow the user to very quickly build a complex large-sized model with a graphical user interface (GUI). Second, students need to get the feeling that any model is a simple and careful abstraction from reality and that final decisions are never made solely on the basis of a computer output. Sensitivity analysis, alternative solution generations, and allowing for subjective feelings and dialogue are a matter of degree and do not harm the usefulness of any model. At the end, people make decisions, not software tools! In §4.1, some often-straightforward model extensions that are frequently proposed during class discussions are given. Last, the students need to be convinced that O.R. is a methodology that can be used in a practical setting. Rather than a black art for mathematicians or scientists, it is a tool used by practitioners who get involved in complex decision-making processes. It is the ultimate goal of the course to convince people that some problems are so complex that they cannot be properly solved without the use of combinatorial optimization models. This complexity issue is often easy to grasp when students try to find solutions for—at first glance—simple problem themselves, quickly realizing that the number of possible solutions grows very quickly with the problem size. A brief summary of the complexity issues discussed in class for the model of §3 is given in §4.2.

The exam cases are selected from different disciplines, such as production, finance, and human resource management. This makes it more likely that

the student preferences are distributed equally over the cases, which leads to a higher solution quality of the model. Experience has shown that when students are asked to gradually build the model in groups, they get a feeling about the complexity of the problem formulation, and the acceptance rate of the solution is high.

3. The Model

3.1. The Mathematical Formulation

The problem formulation depends on the discussion with the students and varies along the background of the students, their constraint formulations, and their objectives. A possible set of problem parameters, the two sets of decision variables, and a problem formulation are given below.

Problem parameters

- S Set of students, index s
- G Set of groups, index g
- C Set of cases, index c
- $w_{s,c}$ Preference weight of student s for case c
- l_c Minimum number of times that case c will be assigned to a group
- u_c Maximum number of times that case c will be assigned to a group
- msg Maximal number of students in one group
- A Set of advanced students

Decision variables

- $x_{s,g,c} = 1$, if student s is assigned to group g to solve case c ,
- $= 0$, otherwise.
- $y_{g,c} = 1$, if the members of group g have to solve case c ,
- $= 0$, otherwise.

$$\text{Minimize } \sum_{s=1}^{|S|} w_{s,c} * x_{s,g,c} \quad (1)$$

$$\text{subject to } \sum_{g=1}^{|G|} \sum_{c=1}^{|C|} x_{s,g,c} = 1 \quad \forall i \in S, \quad (2)$$

$$\sum_{c=1}^{|C|} y_{g,c} = 1 \quad \forall g \in G, \quad (3)$$

$$\sum_{s=1}^{|S|} x_{s,g,c} \leq M * y_{g,c} \quad \forall g \in G, \forall c \in C, \quad (4)$$

$$\sum_{g=1}^{|G|} y_{g,c} \leq l_c \quad \forall c \in C, \quad (5)$$

$$\sum_{g=1}^{|G|} y_{g,c} \geq u_c \quad \forall c \in C, \quad (6)$$

$$\sum_{s=1}^{|S|} \sum_{c=1}^{|C|} x_{s,g,c} \leq msg \quad \forall g \in G, \quad (7)$$

$$\sum_{s=1}^{|A|} \sum_{c=1}^{|C|} x_{s,g,c} \geq \frac{|A|}{|G|} \quad \forall g \in G, \quad (8)$$

$$x_{s,g,c} \in \{0, 1\}, \quad y_{g,c} \in \{0, 1\}, \quad \forall i \in S, \forall g \in G, \forall c \in C. \quad (9)$$

Expression (1) maximizes the total group preferences. A higher preference for a particular case is translated in a higher $w_{s,c}$ value. Constraint set (2) ensures that each student is assigned to a single group/case combination. Constraint set (3) ensures that each group has to solve exactly one single case. Constraint set (4) is a big M constraint that links the $x_{s,g,c}$ and $y_{g,c}$ variables. More precisely, the constraints ensure that students can only be assigned to a particular group/case combination (left-hand side (LHS) of the constraint) if the group is assigned to that particular case (right-hand side (RHS)). The next two constraint sets ensure an equal distribution of cases among the groups and guarantee that each case c is distributed at least l_c times (5) and maximum u_c times (6). Constraint set (7) ensures that each group contains at least msg group members. Constraint set (8) ensures that the advanced modeling students, if any, are equally divided among the groups. Constraint set (9) forces the decision variables to take binary values.

3.2. An Illustrative Example

This section describes a small example that aims to assign 12 student ($|S| = 12$) to three groups ($|G| = 3$). The maximal group size msg equals 4 and the available number of cases equals 3 ($|C| = 3$). The student preferences and their background knowledge (expert in modeling, indicated by yes or no) are summarized in Table 1.

Table 1 Student Preferences and Background Knowledge

S	w_1	w_2	w_3	Adv.
1. Paul	1	2	3	Yes
2. Marie	2	1	3	No
3. Susan	1	3	2	No
4. Peter	2	3	1	Yes
5. John	1	2	3	No
6. Ann	2	3	1	No
7. Clive	1	2	2	No
8. Adam	1	3	2	No
9. Kristen	1	3	2	Yes
10. Heidi	1	2	3	Yes
11. Macie	1	3	2	No
12. Walt	2	1	3	No

Figure 1 Network Representation of the Problem Formulation

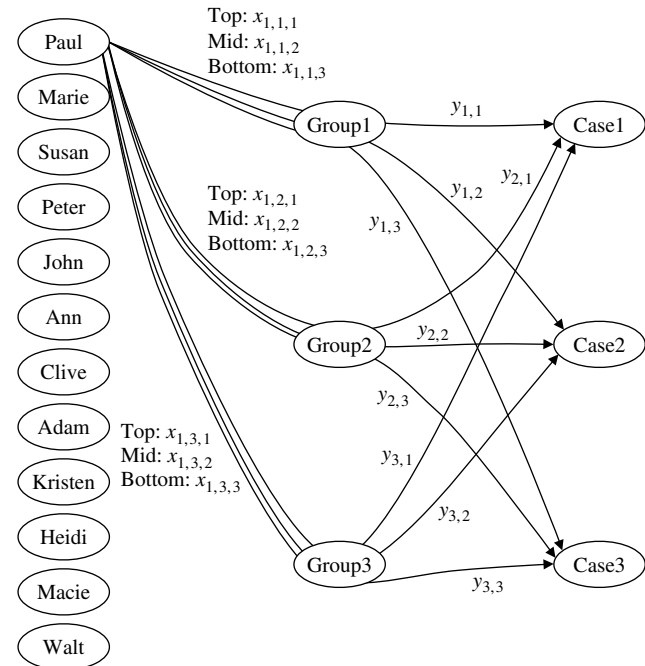


Figure 1 shows a network representation of the assignment problem. The figure shows the $x_{1,g,c}$ variables for Paul and all $y_{g,c}$ variables. In total, the problem contains $12 * 3 * 3 = 108$ $x_{s,g,c}$ variables, $3 * 3 = 9$ $y_{g,c}$ variables, and 36 constraints (excluding the binary constraints given in Equation (9)).

Figure 2 shows an optimal solution obtained with a total preference value equal to 31. Seven students

Figure 2 Example Assignment Solution

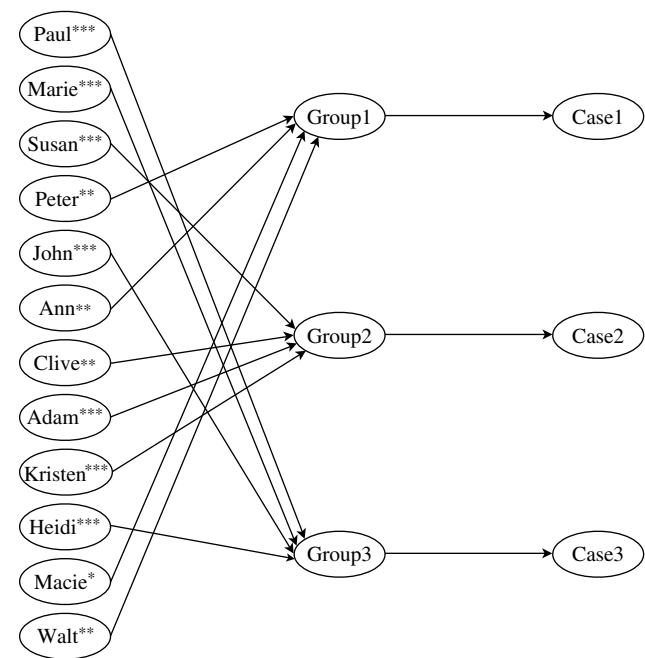
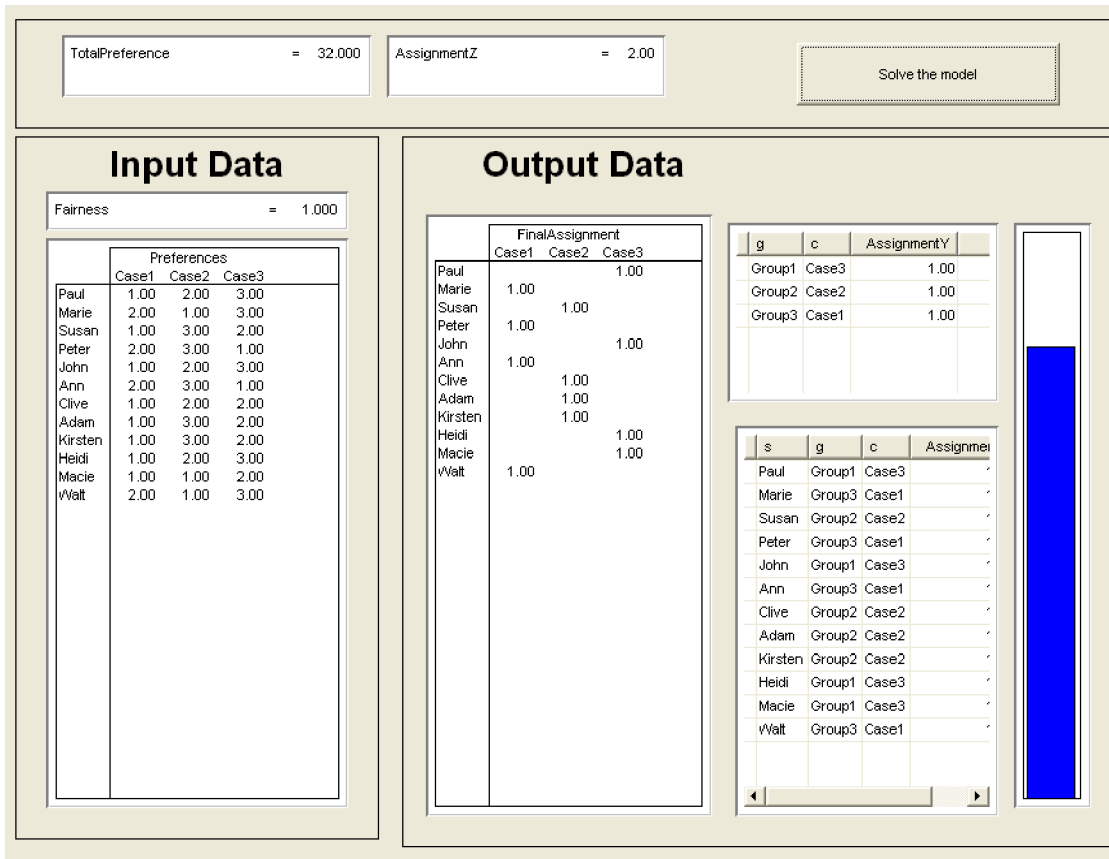


Figure 3 The Student/Group/Case Assignment Problem Using the AIMMS Interface



have their first-choice case (indicated by ***), whereas Peter, Ann, Clive, and Walt have their second-choice case (**). Macie is the unluckiest of the group because she has received her last-choice case (*).

Thanks to the small size of the problem, it is easy to see that a similar total preference can be obtained by switching, e.g., Macie and Marie, to obtain an alternative optimal solution. The preference of Macie goes up with one star whereas the preference of Marie goes down by one star, still reaching the total preference value of 31. However, for larger instances, these alternative solutions are hard to find. Note that the group of Marie can probably live with the switch, because the original group contained only three-stars assignments. Switching people from groups to obtain a more fair case assignment will be discussed in §4.1.

Figure 3 shows the GUI that is projected in class during the construction of the model. This interface can be easily used to guide the students in their problem formulation and allows a simple translation of the decision variables, constraint definition, and objective formulation in a software tool. The GUI is prepared and shows the preferences of the students (input data) received a couple of days earlier. The model is built during the class discussion, and a solution is shown at the end of the discussion.

4. Teaching Experience

The problem description is fairly easy to understand and immediately gets attention when students hear that this class is about an exam case assignment. The group discussion consists of three parts. First, §4.1 shows that the model formulation has a focus on both the definition of objective and constraints, as well as the translation of these ideas into formulas. Second, once the model has been formulated, it easily can be shown that the complexity of these problem types quickly goes up as the size of the class increases. Section 4.2 shows how to open a discussion about hints and tricks for how to reduce problem complexity. Moreover, once the solution is shown to the audience, people always comment on issues and quickly see room for improvement. It can easily be shown that optimization can be seen as a dynamic process where there is room for subjectivity by adding or deleting extra constraints. This also gives the opportunity to show how a near-optimal solution can be obtained quickly, without losing too much quality. Finally, §4.3 illustrates that a discussion of real-life applications of similar models is a necessary and often requested conclusion that enables many students to translate their specific work situation (in case of MBA students) to these types of problem descriptions.

4.1. Model Formulation

4.1.1. Choice of Decision Variables. One of the most animated and probably most important discussions is the choice of the decision variables $x_{s,g,c}$ and $y_{g,c}$ in the model presented in §3. From a student's point of view, an obvious choice of variables would be to split the student/group assignment and group/case assignment into two binary variables, x_{sg} and y_{gc} , respectively. However, when confronted with the problem objective and/or the constraints, people quickly multiply the variables leading to quadratic problem formulation or run into trouble during constraints formulations. These discussions give the instructor an opportunity to start a discussion on the difference between LP, IP, and non-linear programming and illustrate the importance of careful decision variable definition.

4.1.2. Choice of Objective Function. The formulation of the objective function is mostly a straightforward choice. However, other objectives might lead to alternative or completely different solutions, satisfying other needs of the group. An interesting discussion is the difference between a global group optimum and a more individual-oriented optimum.

A *global group* optimum objective is given in (1) and searches for the maximum total preference of the group without considering variation between individuals. However, people are often collegial to each other and prefer a fair assignment above an assignment with many happy people and some frustrated people that received their last-choice case. This leads to a more *individual-oriented optimum* approach that tries to minimize the number of unlucky students that received their last-choice case.

Obviously, different approaches to set up a more fair objective function can be discussed in class, and experience shows that students quickly come up with proposed changes in the weights of the preferences, without changing their relative rankings, to stimulate a more fair distribution of cases among students. An extended objective function is shown in class that aims to avoid low student/case preferences. The introduction of an extra decision variable z that finds the worst assigned student/case preference (worst is equal to the lowest assigned $w_{s,c}$ preference value for a student s that needs to solve case c) by means of the following set of constraints:

$$z \leq \sum_{g=1}^{|G|} \sum_{c=1}^{|C|} w_{s,c} * x_{s,g,c} \quad \forall s \in S. \quad (10)$$

When this variable is added to the objective function (eventually weighted to give more importance to a fair case distribution), the model will try to avoid bad

preference case assignments to students. In the example of §3.2, this will lead to the more fair case distribution as discussed by switching two people.

It is easy to open a discussion to show that the more individual-oriented objective can never be better than the group optimum in terms of total preferences. In class, both approaches are discussed and compared, and it is often the case that the total preference is equal for the two approaches, which illustrates the presence of alternative optimal solutions for the problem. Moreover, it demonstrates that optimization is often a dynamic process where there is room for subjectivity, discussion, and changes on request of the user(s).

4.1.3. Maximal Group-Size Constraint. The most difficult part of the formulation is the big M constraint set to link the $x_{s,g,c}$ and $y_{g,c}$ variable s . During the construction of this constraint set, the big M is introduced as a "big" number. This raises the question of "what is big enough?" It can be shown to students that the M needs to be big enough to avoid eliminating optimal solutions, but that smaller M values are preferable to tighten the RHS of the constraint. When students are asked to determine the minimal value of the big M , many respond that the minimal value must be equal to the size of the student set, because the LHS of this constraint is the sum of all students (i.e., $M = |S|$). This is not an easy exercise, because only a few students can further reduce the big M value to $M = msg$ and can even delete an important constraint from the formulation. In doing so, the original constraints given in constraint sets (4) and (7) can be reduced to a single constraint set:

$$\sum_{s=1}^{|S|} x_{s,g,c} \leq msg * y_{g,c} \quad \forall g \in G, \forall c \in C. \quad (11)$$

An alternative approach is to replace the constraint set of Equation (4) by a constraint for each $x_{s,g,c}$ variable as given below. The different approaches can be compared and discussed in class:

$$x_{s,g,c} \leq y_{g,c} \quad \forall i \in S, \forall g \in G, \forall c \in C. \quad (12)$$

4.2. Model Complexity

Once a model formulation has been selected, a discussion can be started about the balance between problem complexity and solution quality. It can be easily shown that when the class size increases, it is often hard to find an optimal solution in a reasonable time. Section 4.2.1 shows that the problem complexity can often be reduced by adding or changing constraints or by fixing the values of some of the decision variables. Section 4.2.2 gives an example of how a near-optimal search often reduces the computational effort dramatically, without much impact on solution quality.

Additional information, including supplemental material and rights and permission policies, is available at <http://ite.pubs.informs.org>.

4.2.1. Problem Complexity. To further reduce the complexity, students need to become aware of alternative problem formulations. More precisely, when they are shown that the group number does not matter to each student, it can be easily shown that the number of $y_{g,c}$ variables can be reduced dramatically.

The information contained in (5)–(6) allows one to reduce the number of decision variables and constraints. In the example of §3.2, the lower value l_c and upper value u_c on the number of times each case needs to be assigned to groups allows the user to preassign a number of cases to groups, without losing optimality. Indeed, each case needs to be solved exactly once, and hence there will be no need for constraint sets (5)–(6) when the groups are preassigned to the cases as $y_{1,1} = y_{2,2} = y_{3,3} = 1$ and $y_{1,2} = y_{1,3} = y_{2,1} = y_{2,3} = y_{3,1} = y_{3,2} = 0$ (see Figure 4).

However, it is important to show to students that not all $y_{g,c}$ variables can be preassigned. An easy example with three cases and 10 groups and $l_c = 3$ and $u_c = 4$ values (which often corresponds to the problem size of the MBA classes) shows that fixing $y_{1,1} = y_{2,1} = y_{3,1}$ to 1 for the first case and, likewise, fixing $y_{4,2} = y_{5,2} = y_{6,2}$ to 1 and $y_{7,3} = y_{8,3} = y_{9,3} = 1$ for the two other cases still guarantees finding the optimal solution. However, the assignment for the last group, i.e., $y_{10,c}$, cannot be fixed because there is one case that will be assigned four times instead of three times, and the model needs to determine which case this will be.

Figure 4 Simplified Network Representation of the Problem Formulation

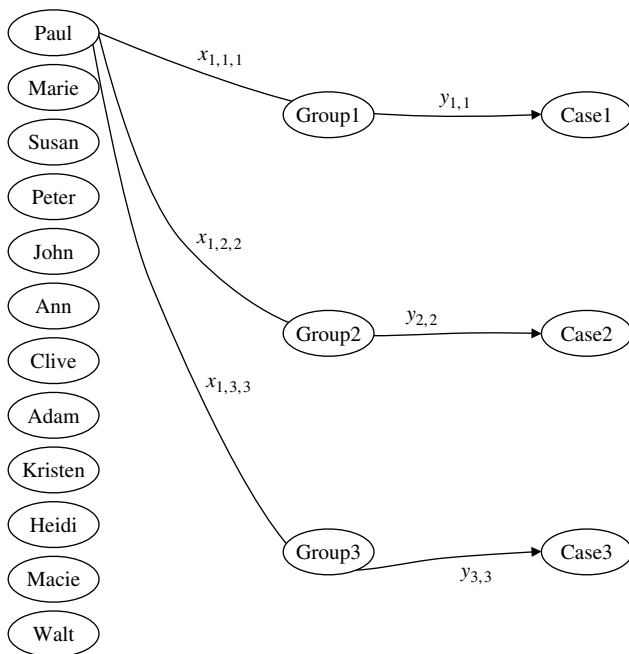
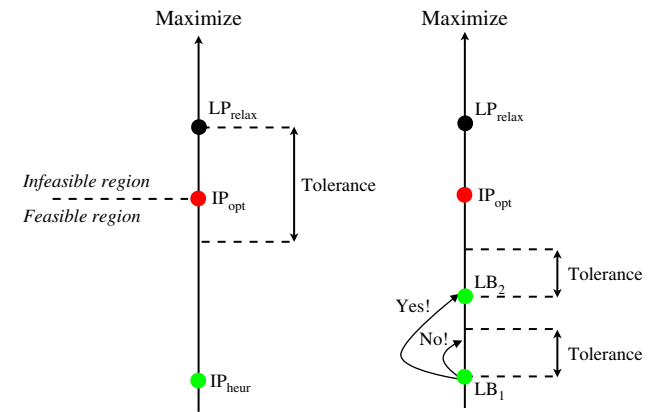


Figure 5 Truncated Search to a Heuristic Solution



4.2.2. Solution Quality. Even with the preassignments discussed in the previous section, finding the optimal solution often requires a lot of time. Consequently, when the problem is solved in real time in class, we often set a tolerance to allow near-optimal solutions.

Figure 5 shows the two space reduction rules that are used in light of the case study of this paper. It explains how lower and upper bounds are used in a search process, how node reduction rules work in a branch-and-bound search, and how this often has little or no effect on the solution quality but dramatically reduces the solution time.

LP/IP Gap. The gap between the LP relaxation and the optimal IP solution can be used to show the difference between the two techniques and the use of lower and upper bounds in a branch-and-bound search process (left part of the figure). This process can be accelerated by setting a tolerance equal to (no effect) or larger than (search space reduction) this gap. Obviously, people quickly come up with the correct idea that if the tolerance has been set too small, this approach has no effect on the solution time. Moreover, this tolerance gives a deviation from the LP-relaxed upper bound (in case of a maximization problem) and gives hardly any information on the maximum deviation between the heuristic solution found and the (unknown) optimal solution.

Optimality Tolerance. This space reduction approach is less obvious to understand and requires a basic understanding of the branch-and-bound solution methodology. A tolerance can be set as a maximum allowable deviation between the optimal (unknown) solution and the solution the students want to obtain. This tolerance defines a zone around the dynamic best-known solution (BKS) in which the branch-and-bound approach can truncate nodes. The right part of the figure shows that the first BKS, denoted by LB_1 , is set with a $x\%$ tolerance such that only solutions that are $x\%$ better than the current BKS are worth investigating. Once a new BKS is found (LB_2), this tolerance

threshold dynamically shifts and the search continues in the same way.

The problem formulation discussed in this paper serves as an ideal tool to elaborate on the complexity dimension of various problem formulations in terms of computational difficulty. Illustrating the need for dedicated solution procedures or heuristic search procedures is not key in the discussion with the students but can be done at later sessions when the students are more familiar with various optimization techniques. Moreover, the need for heuristic solution approaches can be further discussed in the light of other heuristics, such as greedy optimization, truncated branch-and-bound, metaheuristic search, and many more.

4.3. Usefulness in a Business Environment

One of the major advantages of the problem description is how easy it is to understand the problem and the direct implication on the daily life of the student. A second advantage lies in the broad application domain of similar model descriptions and hence the feeling that these O.R. models are applicable outside the class room. At the end of the teaching sessions, students are asked to formulate similar problem descriptions, that occur in their direct environment. Experience has shown that people quickly come up with problem descriptions from their own experience, and MBA students often come back with a similar problem description applicable to their own business or service environment. Undergraduate students often refer to timetabling problems, exam scheduling problems, and assignment problems where students receive locations for their next Erasmus year. In an Erasmus year, students study one semester or a complete academic year abroad at another university. They express their preferences for these foreign universities but may be assigned to a location which was not their first preference.

This session is often extended with cases and problem descriptions from literature or references coming from my own experience. One of the most popular extensions to students are all kinds of staff-scheduling problems that are basically similar assignment problems within the presence of numerous constraints. These constraints are often dictated by internal company policies, which renders them useable to start a real-life discussion from different angles. The nurse-scheduling problem case at the University Hospital, Ghent, Belgium (Maenhout and Vanhoucke 2008b) and the crew-scheduling case at Brussels Airlines, Brussels, Belgium (Maenhout and Vanhoucke 2008a) are used as illustrative material and are easily understood by people because they tackle problems with which they are familiar.

5. Conclusion

In this paper, a simple yet applicable assignment model has been presented to divide a class of students in groups such that each group receives exactly one exam case to solve. The problem formulation can be set up in class during a discussion round and can be easily incorporated in a software tool without excessive use of time.

The advantages of using this model description for teaching purposes are threefold. First, the simple problem formulation is easy to understand and has direct implications on the students because it deals with their final exam assignment of the course. This makes the problem particularly interesting to them, which obviously leads to a high class participation. Second, the model can be used to illustrate many O.R.-related issues, such as the need for heuristic search of a solution, the necessary dynamic adaptations during problem formulation, or the tricks and hints to reduce the problem complexity. Finally, the problem formulation can be easily translated to other settings and environments, which makes it valuable for MBA students to translate this model to their business or service environment.

References

- Aggarwal, R., I. Khera. 1978. Short cases for teaching management science: Their role in closing the gap between theory and practice. *Interfaces* 9 90–94.
- Baker, K., S. Powell. 2002. Methods for assigning students to groups: A study of alternative objective functions. *J. Oper. Res. Soc.* 53 397–404.
- Beheshtian-Ardekani, M., M. Mahmood. 1986. Development and validation of a tool for assigning students to groups for class projects. *Decision Sci.* 17 92–113.
- Belton, V., J. Scott. 1998. Independent learning and operational research in the classroom. *J. Oper. Res. Soc.* 49 899–910.
- Böcker, F. 1987. Is case teaching more effective than lecture teaching in business administration? An exploratory analysis. *Interfaces* 17 64–71.
- Bodily, S. 1996. Teachers' forum: Teaching MBA quantitative business analysis with cases. *Interfaces* 26(November–December) 132–138.
- Borsting, J., T. Cook, W. King, R. Rardin, F. Tuggle. 1988. A model for a first MBA course in management science/operations research. *Interfaces* 18(September–October) 72–80.
- Carraway, R., D. Clyman. 1997. Managerial relevance: The key to survival for OR/MS. *Interfaces* 27 115–130.
- Chen, M. 1981. Teaching OR/MS as a service course in business/management schools. *Interfaces* 11 75–79.
- Corner, J. 1997. Teaching decision analysis. *Interfaces* 27(November–December) 131–139.
- Corner, J., P. Corner. 2003. Teaching OR/MS using discussion leadership. *Interfaces* 33(May–June) 60–69.
- Donohue, J., J. Fox. 1993. An investigation into the people-sequential heuristic method of group formation. *Decision Sci.* 24 493–508.
- Eaves, B. 1997. Learning the practice of operations research. *Interfaces* 27(September–October) 104–115.
- Evans, J. 1992. Creativity in MS/OR: Improving problem solving through creative thinking. *Interfaces* 22(March–April) 87–91.
- Franz, L. 1989. Integrating analysis and communication skills into the decision science curriculum. *Decision Sci.* 20 830–843.

- Grenander, U. 1965. An experiment in teaching operations research. *Oper. Res.* **13**(September–October) 859–862.
- James, J. 1988. Teaching operational research using IBM-PC compatibles. *J. Oper. Res. Soc.* **39**(August) 779–789.
- Kao, C., Q.-J. Yeh, Y.-C. Tsai. 1997. Teaching undergraduate operations research in some of the business schools in Asia. *Interfaces* **27**(September–October) 93–103.
- Lasdon, L., J. Liebman. 1998. The teacher's forum: Teaching nonlinear programming using cooperative active learning. *Interfaces* **28** 119–132.
- Liberatore, M., R. Nydick. 1998. Issue in education: Modeling software and the MBA OR/MS course. *OR/MS Today* **25**(4) 8.
- Liebman, J. 1994. New approaches in operations research education. *Internat. Trans. Oper. Res.* **1** 189–196.
- Liebman, J. 1998. Teaching operations research: Lessons from cognitive psychology. *Interfaces* **28**(March–April) 104–110.
- Maenhout, B., M. Vanhoucke. 2008a. Automatic personalized crew rostering at Brussels Airlines. Technical report, Ghent University, Ghent in Flanders, Belgium.
- Maenhout, B., M. Vanhoucke. 2008b. The impact of incorporating nurse-specific characteristics in a cyclical scheduling approach. Technical Report, Ghent University, Ghent in Flanders, Belgium.
- Mingers, J., F. O'Brien. 1995. Creating student groups with similar characteristics: A heuristic approach. *Omega Internat. J. Management Sci.* **23** 313–321.
- Miser, H. 2000. The easy chair: What OR/MS workers should know about the early formative years of their profession. *Interfaces* **2** 99–111.
- Miyaji, I., K. Ohno, H. Mine. 1987. Solution method for partitioning students into groups. *Eur. J. Oper. Res.* **33** 82–90.
- Muller, T. 1989. Assigning students to groups for class projects: An exploratory test of two methods. *Decision Sci.* **20** 623–634.
- Powell, S. 1995. The teachers' forum: Teaching the art of modeling to MBA students. *Interfaces* **25**(May–June) 88–94.
- Powell, S. 1997. The teachers' forum: From intelligent consumer to active modeler, two MBA success stories. *Interfaces* **27** 88–98.
- Powell, S. 1998. The teachers' forum: Requiem for the management science course? *Interfaces* **28** 111–117.
- Prince, M. 2004. Does active learning work? A review of the research. *J. Engrg. Ed.* **93** 223–231.
- Robinson, S., M. Meadows, J. Mingers, F. O'Brien, E. Shale, S. Stray. 2003. Teaching OR/MS to MBAs at Warwick business school: A turnaround story. *Interfaces* **33**(March–April) 67–76.
- Sampson, S., J. Freeland, E. Weiss. 1995. Class scheduling to maximize participant satisfaction. *Interfaces* **25** 30–41.
- Stallaert, J. 1997. Automated timetabling improves course scheduling at UCLA. *Interfaces* **27** 67–81.
- Strasser, S., C. Ozgur. 1995. Undergraduate business statistics: A survey of topics and teaching methods. *Interfaces* **25** 93–103.
- Tavares, L. 1994. Education in OR/MS at a glance: Results of the first IFORS survey. *Internat. Trans. Oper. Res.* **1** 389–394.
- Tingley, G. 1987. Can MS/OR sell itself well enough? *Interfaces* **17** 41–52.
- Winston, W. 1996. The teachers' forum: Management science with spreadsheets for MBAs at Indiana University. *Interfaces* **26** 105–111.
- Zahedi, F. 1985. MS/OR education: Meeting the new demands on MS education. *Interfaces* **15** 85–94.