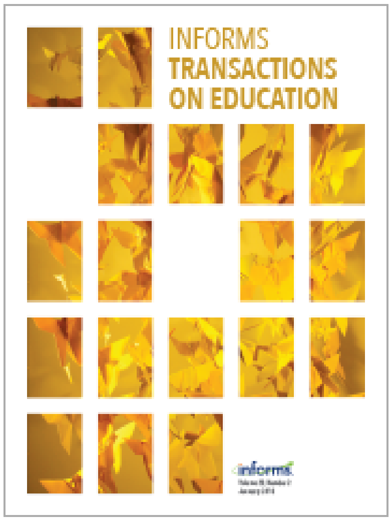

INFORMS Transactions on Education

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Using Mixed-Integer Programming to Win a Cycling Game

Jeroen Beliën, Dries Goossens, Daam Van Reeth, Liesje De Boeck,

Vol. 11, No. 3, May 2011, pp. 93–99
ISSN 1532-0545 | 11 | 1103 | 0093

INFORMS
Transactions on Education

informs®

DOI 10.1287/ited.1110.0062

Using Mixed-Integer Programming to Win a Cycling Game

Jeroen Beliën
Campus Economische Hogeschool, Hogeschool-Universiteit Brussel, B-1000 Brussels, Belgium, and
Department of Decision Sciences and Information Management,
Research Center for Operations Management, Katholieke Universiteit Leuven,
B-3000 Leuven, Belgium, jeroen.belien@hubrussel.be

Dries Goossens
Department of Decision Sciences and Information Management, Katholieke Universiteit Leuven,
ORSTAT, B-3000 Leuven, Belgium, dries.goossens@econ.kuleuven.be

Daam Van Reeth
Campus Economische Hogeschool, Hogeschool-Universiteit Brussel,
B-1000 Brussels, Belgium, daam.vanreeth@hubrussel.be

Liesje De Boeck
Campus Economische Hogeschool, Hogeschool-Universiteit Brussel, B-1000 Brussels, Belgium, and
Department of Decision Sciences and Information Management,
Research Center for Operations Management, Katholieke Universiteit Leuven,
B-3000 Leuven, Belgium, liesje.deboeck@hubrussel.be

This paper presents an application of optimization modeling to the winning of a popular cycling game. The application includes real-life data of contempory cyclists. It also has the potential to motivate students with a competitive but fun "race" for a solution. Because the developed optimization model contains features of knapsack problems, multiperiod inventory problems, and logical constraint modeling, it is perfectly suitable for a concluding case study in an undergraduate operations research/management science course. The application was originally developed for an MBA operations research course focusing on spreadsheet modeling skills, but it can also be used in courses that focus on algebraic modeling of optimization problems.

Key words: cycling; game; spreadsheet modeling; mixed-integer programming; knapsack problems; multiperiod inventory problems
History: Received: May 2009; accepted: August 2010.

1. Introduction and Literature Review

The best way to make business students enthusiastic about operations research/management science (OR/MS) is to let them experience how useful the skills gained can be in their personal lives. Ask your students to model an imaginary business problem and only a minority will be motivated to find the correct solution. Ask your students to model a real-life business problem and maybe, if you are lucky, half of them will make a serious effort. Do the same but use an application that fascinates students, such as popular sports (basketball, football, or baseball in the United States; cycling or soccer in Europe), and the majority will be eager to find the solution. Finally, combine a popular application with a competition aspect (such as playing a game) and almost all of the students will be determined to find the optimal solution (and win the competition).

In a special issue of this journal on "SpORts in the OR Classroom," Cochran (2004) explains how and why sports can help instructors of applied quantitative methods achieve several important pedagogical goals. This special issue presents several articles documenting the successful classroom use of sports contexts, including examples from each of the major North American team sports (baseball, hockey, basketball, soccer, and football). In that issue, Trick (2004) describes how to teach integer programming using sports scheduling applications. As far as we know, our paper is the first OR/MS education paper in the context of cycling.

As has been argued by Sniedovich (2002), games are a valuable source for developing educationally rich material for OR/MS courses. More specifically, games are effective for teaching linear programming (LP) and integer programming (IP) modeling to students who lack the motivation or background to study quantitative methods. For instance, Chlond and Toase (2002) propose several IP models for chessboard placement and closely related puzzles for use in a classroom. Other examples of logic puzzles can be found in Chlond (2008) or in Chlond (2005) in which the popular Su Doku and log pile game are solved using integer programming.

The game presented in this paper is suitable for a mixed-integer programming approach involving three important classes of optimization problems and corresponding modeling techniques: knapsack problems, multiperiod problems, and logical constraint modeling.

1. In many textbooks, the knapsack problem is used to introduce integer programming problems (e.g., Winston 2004) and dates back to the early days of linear programming (e.g., Dantzig 1957). Since then, many papers and even entire books have been written on applications and algorithms for knapsack problems (e.g., Martello and Toth 1990; Kellerer et al. 2005).

2. Multiperiod or dynamic models have also been widely studied. Mixed-integer programming formulations have been proposed, for example, for dynamic inventory (lot sizing) problems (Stadtler 1996), multiperiod financial models (Rohn 1987), and multiperiod work scheduling problems (Franz and Miller 1993).

3. The specific use of integer programming to model logical constraints has been illustrated by Chlond and Toase (2003), who focus on the modeling of logical conditions in puzzles by IP.

The remainder of this paper is organized as follows. Section 2 introduces the Gigabike game and explains which aspects make it interesting for use in a classroom. Section 3 presents the optimization models for solving the Gigabike optimization problem while §4 discusses some classroom experiences. Finally, §5 concludes this paper.

2. The Gigabike Game

2.1. The Origin of Gigabike

Betting on the outcome of future events has always been a popular pastime. These events historically ranged from the totally unpredictable random roll of dice to the brutal and sometimes more predictable outcome of a fight. The outcome of a sports competition is not totally random but depends on the qualities of the athletes and teams involved. As a result, in contrast to taking part in a dice game or a lottery, sports betting allows participants to exploit an information advantage (such as prior results) to predict the final outcome. However, uncertainty remains important enough to create a sufficient difference in opinion among spectators and allow sports bets to take place. It is this unique combination of predictability and uncertainty that makes sports betting particularly appealing.

Halfway through the 1990s, the popular Belgian newspapers *Het Laatste Nieuws* and *Het Nieuwsblad* started to use sports betting games as a marketing instrument to boost sales. Interest in popular sports, especially soccer and road cycling, was combined with a game element of predicting which athletes would do well. Participants in these games typically had to act as "managers" of a team of soccer players or riders selected by themselves. The goal was to select a fixed number of real soccer players (Megascore) or professional cyclists (Megabike) scoring the most points in selected sports events. By paying a small fee, entrants in the game could win interesting (monetary) prizes if they selected the best fantasy team.

For road cycling, newspapers organized games around the popular spring classic races such as the Tour of Flanders and Paris-Roubaix and the widely viewed Tour de France. Unfortunately for die-hard cycling fans, less popular races such as the Giro d'Italia or the autumn classic races were not included in these newspaper games. Peter Samoy and Mark Vanderwegen, two workmates with a common interest in cycling, saw an opportunity and developed a new game concept. In the year 2000, they started "Gigabike," a noncommercial year-long Internet cycling game that includes all major road cycling races (see http://www.gigabike.be). Although no prizes could be won, the game became an instant success amongst Belgian and Dutch cycling fans because of its originality. The number of players in the game rose steadily from 40 in 2000 to 500 in 2010.

2.2. The Principles of the Gigabike Game

The Gigabike game basically operates as follows. At the start of the road cycling season, each Gigabike player selects a team of 30 male riders from a world ranking of all professional road cyclists. In this ranking, each cyclist is assigned a so-called cycling quotient (CQ) value based on his or her performances during the last 12 months (see http://www.cqranking.com). This value reflects the past quality of the rider and might indicate his future performance. The total CQ value of a team must be less than a given CQ budget. Observe that this budget restriction makes it impossible to select only the best riders, that is, the professional cyclists with the highest CQ value. At each of five fixed moments during the season, one can modify the team by substituting at most five cyclists. Hence, the season consists of six periods.

However, the sum of the CQ values of the incoming riders cannot exceed the sum of the values of the outgoing riders plus the remaining CQ budget of the preceding period, if any. The winner of the game is the person whose team gains the most points over the course of the whole season, which consists of over 60 races and more than 120 racing days.

Two dynamic characteristics distinguish Gigabike from other Internet or newspaper cycling games. First, the transfer opportunities allow for strategic decision making during the season. Second, the value of the riders changes throughout the year as results from 12 months ago are deleted and replaced by new results. After each race, the riders' CQ values are updated as follows: Each rider loses the CQ points won in last year's edition of that race and gains the CQ points earned in this year's edition. To give an example, suppose a particular rider won last year's Tour of Flanders. This rider will lose the resulting CQ points after the next Tour of Flanders but may win them back if he wins this year's Tour of Flanders. Hence, in the best case, the rider keeps his current value after the Tour of Flanders. At the other extreme, cyclists who scored no CQ points in last year's Tour of Flanders cannot see their CQ value decrease: Any CQ points in this year's Tour of Flanders will increase their CQ value. This dynamic aspect makes the game particularly appealing. Not only luck but also strategic considerations and a good knowledge of the racing schedule of certain cyclists determine the final result of a team. For instance, although a certain cyclist may be a preseason favorite for winning the Tour de France, from a strategic point of view it could be better not to have him on the initial team because his value may decrease before the start of the Tour de France. Therefore, it is not always in a player's best interest to have the best riders on a team right from the start. However, because the number of transfer opportunities is limited, some riders will be on the team for the whole season and a well-balanced starting selection of cyclists for all types of races is crucial in order to have any chance of winning the game.

2.3. Using Gigabike in a Classroom

In the classroom, we focus on a component of the Gigabike game that takes place after all of the races are completed. At the end of each season, some Gigabike players engage in a new competition. Given that all results are now known, they try to find the optimal team, i.e., the team that would have collected the most points if all results had been known (or perfectly forecasted) in advance. As we will show in §3, the problem of finding this optimal team can be modeled as a mixed-integer linear program that has characteristics of several basic problems, i.e., knapsack problems, multiperiod inventory problems, and integer programming modeling.

A spreadsheet (see the file Gigabike_OPT_2008.xls[1]) greatly simplifies the search for the optimal cycling team because it contains all the costs (CQ values) and all the points collected by the riders for each of the six periods. Moreover, given a particular selection, it calculates automatically the budget spent and the number of transfers made in each period. The user only has to indicate, for each of the 6 periods, which 30 riders are selected in the team. Violated constraints (such as budget exceeded, more than 5 transfers between two periods, or not exactly 30 riders) are indicated in red. To reduce the scale of the problem, the spreadsheet only contains the 112 best-scoring riders of the past cycling season. The spreadsheet contains detailed instructions on how to use its information to evaluate a team.

We recommend distributing this spreadsheet file to the students at least one week before the actual lecture on the Gigabike modeling problem and ask them to search for the best possible team. If you promise a small reward for the highest score, many of your students will be further motivated to find a good solution and become familiar with the problem.

Most students will start by experimenting and selecting a team of well-known riders with high CQ points, adding a few low CQ point riders in order to respect the budget. The next step usually involves looking for interesting transfers. Without realizing it, most students develop an heuristic. Therefore, it may be a good idea to begin the lecture with a discussion of simple heuristics based on various approaches to making a selection of riders (e.g., making a selection in a greedy fashion after sorting the riders). These heuristics will make sense to the students because most of them will have tried something similar in their searches for good solutions.

The spreadsheet Gigabike_HEUR_2008.xls illustrates one particular heuristic. The underlying idea is that for each rider and each period, the ratio between the rider's points scored from this period until the end of the season and the rider's CQ value in this period determines how interesting that rider is for each period. For the initial selection, we greedily pick the 30 riders with the highest ratio for period 1. If the budget is violated, we drop the most expensive rider and replace him with the next best rider in the ratio list (and repeat this until the budget is respected). To determine the first transfers, we remove the five riders with the lowest ratio for period 2 and replace them with the riders with the best ratio for period 2 that are not yet on the team. Again, if the budget

[1] This and other supplementary files are available from http://ite.pubs.informs.org/.

constraint is violated, we remove our most expensive rider instead of the best of the five riders with the lowest period 2 ratio. The transfers for the next periods are determined in a similar way. In our experience, about half of the students manage to do better than this heuristic, and students usually quickly come up with ideas to further improve this heuristic (e.g., by increasing the weight of the points scored in the ratio or by squaring them). This experience with heuristics will make it easy to convince the students that it will be useful when the problem is too large and the available solvers (e.g., Frontline's standard Solver© for Microsoft Excel©) are unable to solve it in reasonable computation times.

At this point, we present to the students the optimal solution that is typically 5% better than the best score found by the students. In 2007 and 2008, the optimal score was also better than the best score found by the Gigabike participants who puzzled over it for almost one month. The optimal score in 2008 found by the optimization model in 40 seconds computation time is 37,385 points, while the score obtained by applying the heuristic described earlier is only 26,198 points. The best student found a score of 35,325 points (after one week), which is much better than the heuristic but is still far below the optimal score. The best score found by the Gigabike participants (after four weeks) was 37,375, which is still 10 points less than the optimal score. Once they see the proven power of optimization modeling, students generally become curious about the models.

3. Discussion of Models

This section presents the mixed-integer programming model for the Gigabike optimization game. Because student versions of most optimization packages are limited to a small number of decision variables (up to 200 changing cells in the standard solver), we propose to develop the model for a smaller instance of the problem, i.e., we only consider the 18 best performing riders, a team consisting of 10 riders (instead of 30), and a budget limited to 12,000 CQ points (instead of 20,000). Moreover, at most three (instead of five) transfers are allowed between successive periods; the number of periods remains six. Using these dimensions, the model can be solved using a standard version of solver. Although the model is not difficult, we strongly recommend that participants apply a gradual formulation approach, especially if their quantitative background is limited. After all, the aim of this project is to increase students' modeling skills, not overwhelm them with a brilliant optimization model. Fortunately, the modeling process can be effectively divided into three increasingly difficult stages, each of which corresponds to an important class of optimization problems. In this way, the link between theory and practice becomes clear to students. The three stages are:

1. Knapsack problem modeling.

See "Gigabike_1_template.xls" for a template Excel Solver model.

See "Gigabike_1_solution.xls" for the full model.

2. Multiperiod modeling.

See "Gigabike_2_template.xls" for a template Excel Solver model.

See "Gigabike_2_solution.xls" for the full model.

3. Logical constraint modeling.

See "Gigabike_3_template.xls" for a template Excel Solver model.

See "Gigabike_3_solution.xls" for the full model.

We briefly discuss these three stages and present the corresponding optimization models.

3.1. Knapsack Problem Modeling

To start, assume that the six periods are independent. In this simplified case, for each period an identical, independent problem needs to be solved. Given the costs (CQ values) and the gains (CQ points obtained) of each rider in that period, the problem is to find the 10 riders within a total CQ value of 12,000 (the budget constraint) who collect, in total, the most CQ points. With independent periods, we do not have to bother with the remaining budget (any unused budget is simply lost) or about transfers (we can even select 10 new riders in each period). Each period's problem is a knapsack problem with one extra constraint—the knapsack must consist of exactly 10 items. Let x_{ij} be one if rider i is selected in period j and zero otherwise, p_{ij} be the points obtained by rider i in period j, and c_{ij} be the cost (CQ value) of rider i in period j. Then, the formulation of the first stage of the problem is as follows:

$$\text{maximize} \quad \sum_{i=1}^{18}\sum_{j=1}^{6} p_{ij}x_{ij}, \tag{1}$$

$$\text{subject to} \quad \sum_{i=1}^{18} c_{ij}x_{ij} \le 12{,}000, \quad \forall j=1,\ldots,6, \tag{2}$$

$$\sum_{i=1}^{18} x_{ij} = 10, \quad \forall j=1,\ldots,6, \tag{3}$$

$$x_{ij} \in \{0,1\}, \quad \forall i=1,\ldots,18 \; \forall j=1,\ldots,6. \tag{4}$$

Constraints (2) ensure that the total value of the team in each period does not exceed the budget. Constraints (3) limit the number of selected riders to exactly 10.

This model ignores three aspects of the real game. (1) It does not take into account the fact that any unused budget from the first period can be used in the second period while any unused budget in the second period can be used in the third period, etc. (2) It assumes a budget constraint of 12,000 CQ points in each period, which is not entirely correct because in the real game a team can have a larger value than 12,000 CQ points after the first transfer moment (this will be the case when the selected riders increase their total CQ value, i.e., if they perform better than the previous year during the completed period(s)). (3) It allows for a solution with more than three transfers between two periods. We address the first two aspects by extending the model to multiple dependent periods.

3.2. Multiperiod Modeling

To allow for future use of the remaining budget and to have a correct calculation of the available budget in the periods after period 1, balance constraints must be added to the model. Balance constraints are typical of multiperiod problems such as dynamic inventory, work scheduling, or investment problems. A balance constraint links two consecutive periods by setting the output of one period equal to the input of the next period.

Let the variable r_j denote the remaining budget of period j:

$$r_j \geq 0, \quad \forall\, j = 1, \ldots, 6. \tag{5}$$

Replace constraint set (2) by the following constraints:

$$\sum_{i=1}^{18} c_{i1} x_{i1} + r_1 = 12{,}000, \tag{6}$$

$$r_{j-1} + \sum_{i=1}^{18} c_{ij} x_{i,\,j-1} - \sum_{i=1}^{18} c_{ij} x_{ij} = r_j, \quad \forall\, j = 2, \ldots, 6. \tag{7}$$

Equation (6) calculates the remaining budget in period 1. Note that the budget cannot be exceeded because the r_j variables are required to be nonnegative and a budget violation corresponds to a negative remaining budget ($r_j < 0$). Equation (7) define the balance restrictions, saying that the remaining budget from the previous period plus the incomes of the riders sold minus the costs of the new riders selected must be equal to the remaining budget that can be carried over to the next period. Note that riders are always bought (sold) for their value at the moment of purchase (sell). Hence, it is possible to sell a rider at a price higher than the price paid when the rider was bought. This will be the case if the rider performed better than he did in the preceding year during the periods under consideration (between time of buy and the time of sell). The opposite is also possible. Observe that for riders who are not transferred, i.e., those riders who are on the team in period $j-1$ and also in period j, the net contribution to the budget equals zero. Indeed, these riders have a positive as well as a negative contribution to the left-hand side of Equation (7) and at the same cost, i.e., at their current value c_{ij}.

3.3. Modeling of Logical Constraints

The optimal solution to the previous model does not necessarily satisfy the restriction on the maximal number of transfers between two periods. If fewer than seven riders are common to two succeeding periods, then more than three transfers took place, which is not allowed. In order to model the transfer restriction, we need a new binary decision variable t_{ij}, which equals one if rider i is transferred into the team in period j and zero otherwise:

$$t_{ij} \in \{0, 1\}, \quad \forall\, i = 1, \ldots, 18\ \forall\, j = 2, \ldots, 6. \tag{8}$$

The model can then be extended by adding the following constraints:

$$t_{ij} \geq x_{ij} - x_{i,\,j-1}, \quad \forall\, i = 1, \ldots, 18\ \forall\, j = 2, \ldots, 6 \tag{9}$$

$$\sum_{i=1}^{18} t_{ij} \leq 3, \quad \forall\, j = 2, \ldots, 6. \tag{10}$$

Having introduced the binary variable t_{ij}, it is easy to state the transfer restriction by requiring that the sum of the t_{ij} variables must be less than or equal to three in each period via inequalities (10). We only have to make sure that t_{ij} equals one for each new rider i, that is, for each rider who is selected for the team in period j but was not yet on the team in period $j-1$. Modeling this kind of conditional relation is typical for integer programming formulations. Constraints (9) ensure that this relation holds by requiring t_{ij} to be one only if the rider is on the team in period j (i.e., when $x_{ij} = 1$) and not in period $j-1$ (i.e., when $x_{i,\,j-1} = 0$). This final model is the complete mixed-integer programming model for the smaller instance of the ex post Gigabike optimization problem. The same model can be used to find an optimal solution for the Gigabike problem when it includes all the riders. Only the dimensions of the problem will change.

4. Classroom Experiences

Since 2007, the Gigabike game has been used to teach mixed-integer programming modeling to students of both the commercial engineering program and the MBA program at University College Brussels. We have received a lot of positive reactions from this class. To evaluate the motivational and instructional effectiveness of the Gigabike game in a more formal way, we composed a questionnaire consisting of eight questions. The first five questions measure to what

Table 1 Gigabike Evaluation Questionnaire

	Responses (%)				
Question	Strongly agree	Agree	?	Disagree	Strongly disagree
(1) The Gigabike game has increased my interest in operations research.	15	70	5	5	5
(2) Thanks to the Gigabike game, I'm more convinced that optimization modeling techniques can be applied to real-life problems.	50	40	10	0	0
(3) If the Gigabike case had been taught before I had to choose my thesis subject, I would have taken into consideration a thesis subject situated within the field of operations research.	20	25	20	25	10
(4) If I had the opportunity to specialize in a particular subject (via long or short courses), then the Gigabike game would contribute to the fact that I am more inclined to choose a direction related to operations research.	15	35	30	15	5
(5) The Gigabike game has increased my interest in a job (academic, consultancy, business…) within the field of operations research.	5	30	35	20	10
(6) The Gigabike game has improved my modeling skills related to knapsack problems.	25	40	20	15	0
(7) The Gigabike game has improved my modeling skills related to dynamic (multiperiod) problems.	25	40	20	15	0
(8) The Gigabike game has improved my modeling skills related to integer programming problems (more specifically, modeling if-then constraints).	20	55	10	15	0

extent the Gigabike game increased students' interest in operations research. The last three questions measure to what extent the Gigabike game helped to improve the students' modeling skills. The questionnaire was sent to 27 final-year students in the master's course in commercial engineering at University College Brussels who took the class on the Gigabike game in November 2009. The questionnaire was completed by 20 out of the 27 students. The questions together with the response scores are reported in Table 1.

The most remarkable results can be summarized as follows. The majority (85%) of the students became more interested in operations research (Question 1) and no less than 90% of them confirm that the Gigabike game increased their belief in real-life applications of optimization modeling techniques (Question 2). Almost half of the students (45%) would have considered thesis subjects related to operations research if they had taken the Gigabike class before they had to choose thesis subjects (Question 3). More than one-third of the students (35%) became more interested in jobs related to operations research because of the Gigabike game (Question 5). Almost two-thirds of the students believe that their modeling skills with respect to knapsack problems, dynamic problems, and integer programming modeling have improved thanks to the Gigabike game (Questions 6–8). These results confirm our belief in the power of the Gigabike game as a useful educational tool to teach operations research.

Besides the reported results of the questionnaire for the Gigabike class in November 2009, we received much (informal) positive feedback from many other students (commercial engineers as well as MBA students) in earlier years (2007 and 2008). First of all, thanks to the game and sports element, students are much more motivated to find the optimal solution to the problem (and beat their opponents). Consequently, most students make a serious effort to understand the characteristics of the Gigabike game and, particularly, the optimization problem. Many students have asked us for more time to tackle the Gigabike puzzle, a request we do not often get when students have to model a (fictitious) business case. Some students even immediately started to formulate the problem as a mixed-integer programming (MIP) model though we only asked them to find the best possible solution they could by trial and error. The Gigabike example, therefore, provides a good illustration of how a structured 'model formulation' approach eventually outperforms a quick trial-and-error approach that focuses on immediate success. This is an important lesson for students who might have to make important business decisions in their future careers.

5. Conclusion

Recently, (mixed) integer programming has gained in popularity (Laundry et al. 2009). Indeed, advances in solution methods as well as computer hardware and software have considerably decreased the computing times needed to solve these models. As a result, the importance and relevance of (M)IP has also increased for solving real-life managerial-related problems. However, getting students really excited about (M)IP is not always an easy task. One way to overcome this is to let students experience how (M)IP can be used in applications taken from their own fields of interest. In this paper, we use an ex post game component of a popular cycling game (the Gigabike game) that involves real-life data of all current cyclists. Furthermore, we stimulate the students' interest by adding a competitive aspect: the week before the solution is presented and discussed in class acts as a real race among all the students for winning the game. A spreadsheet file facilitates this search process.

The game is suitable for an undergraduate OR/MS course as a case study to conclude an (M)IP module. Indeed, the model can be gradually built up in three distinctive phases, each phase covering an important class of (M)IP optimization methods. In the first phase, we consider the model as made up of independent knapsack problems (one for each period). The second phase introduces dependency between the periods, turning it into a multiperiod model. Finally, the third phase adds a typical logical constraint in order to obtain the correct final model. The models of all phases can be solved using Frontline's standard version of Excel Solver©, thus making the models accessible to every student. Class experiences with the game foster our belief that using applications from students' own worlds of interest is valuable in getting students excited about what they tend to perceive as dull theoretical models (in this case, (M)IP models). The competitive component as an introduction to the game has impressively stimulated students' interest, and this was visible during classes and in students' answers to the questionnaires. We hope that, in this way, students learned about the power such models can provide and, more importantly, started to get a feeling about the relevance these models have in real-life management applications.

Supplementary Material

Files that accompany this paper can be found and downloaded from http://ite.pubs.informs.org/.

Acknowledgments

We are extremely grateful to Peter Samoy and Mark Vanderwegen for developing the unrivalled Gigabike game. Apart from providing years of game pleasure, Gigabike was the authors' inspiration for the educational tool presented in this paper. They thank their colleague Mark Corner for correcting this paper with respect to the English language. Finally, the authors thank the anonymous referee and associate editor for their useful comments on improving the presentation and content of this paper.

References

Chlond, M. J. 2005. Classroom exercises in IP modeling: Su doku and the log pile. *INFORMS Trans. Ed.* **5**(2) 77–79. http://www.informs.org/Pubs/ITE/Archive/Volume-5.

Chlond, M. J. 2008. Puzzle—O.R. with the fairies. *INFORMS Trans. Ed.* **8**(2) 100–102. http://www.informs.org/Pubs/ITE/Archive/Volume-8.

Chlond, M. J., C. M. Toase. 2002. IP modeling of chessboard placements and related puzzles. *INFORMS Trans. Ed.* **2**(2) 1–11. http://www.informs.org/Pubs/ITE/Archive/Volume-2.

Chlond, M. J., C. M. Toase. 2003. IP modeling and the logical puzzles of Raymond Smullyan. *INFORMS Trans. Ed.* **3**(3) 1–12. http://www.informs.org/Pubs/ITE/Archive/Volume-3.

Cochran, J. J. 2004. Introduction to the special issue: SpORts in the OR classroom. *INFORMS Trans. Ed.* **5**(1) 1. http://www.informs.org/Pubs/ITE/Archive/Volume-5.

Dantzig, G. B. 1957. Discrete variable extremum problems. *Oper. Res.* **5**(2) 266–277.

Franz, L. S., J. L. Miller. 1993. Scheduling medical residents to rotations: Solving the large-scale multiperiod staff assignment problem. *Oper. Res.* **41**(2) 269–279.

Kellerer, H., U. Pferschy, D. Pisinger. 2005. *Knapsack Problems.* Springer Verlag, Berlin/Heidelberg.

Martello, S., P. Toth. 1990. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, New York.

Rohn, E. 1987. A new LP approach to bond portfolio management. *J. Financial Quant. Anal.* **22**(4) 439–467.

Sniedovich, M. 2002. OR/MS games: 1. A neglected educational resource. *INFORMS Trans. Ed.* **2**(3) 86–95. http://www.informs.org/Pubs/ITE/Archive/Volume-2.

Stadtler, H. 1996. Mixed integer programming model formulations for dynamic multi-item multi-level capacitated lotsizing. *Euro. J. Oper. Res.* **94**(3) 561–581.

Trick, M. 2004. Using sports scheduling to teach integer programming. *INFORMS Trans. Ed.* **5**(1) 10–17. http://www.informs.org/Pubs/ITE/Archive/Volume-5.

Winston, W. 2004. *Operations Research: Applications and Algorithms*. Thomson Learning Brooks/Cole, London.