Decision Support Systems 52 (2012) 505–513

Contents lists available at SciVerse ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

udpSkeduler: A Web architecture based decision support system for course and classroom scheduling

Jaime Miranda [a,*], Pablo A. Rey [b], José M. Robles [c]

[a] Department of Management Control and Information Systems, School of Economics and Business, Universidad de Chile, Diagonal Paraguay 257, Santiago, Chile
[b] School of Industrial Engineering, Faculty of Engineering, Universidad Diego Portales, Av. Ejército 441, Santiago, Chile
[c] Faculty of Engineering, Universidad del Desarrollo, Av. Plaza 680, Santiago, Chile

ARTICLE INFO

Article history:
Received 6 January 2011
Received in revised form 20 June 2011
Accepted 6 October 2011
Available online 13 October 2011

Keywords:
Education planning
Web-based decision support system
Class scheduling
University timetabling
Operations research
Case study

ABSTRACT

A key process for post-secondary educational institutions is the definition of course timetables and classroom assignments. Manual scheduling methods require enormous amounts of time and resources to deliver results of questionable quality, and multiple course and classroom conflicts usually occur. This article presents a scheduling system implemented in a Web environment. This system generates optimal schedules via an integer-programming model. Among its functionalities, this system enables direct interaction with instructors in order to gather data on their time availability for teaching courses. The results demonstrate that significant improvements over the typical fully manual process were obtained.

1. Introduction

For decades, higher educational institutions have relied on computer-based systems to support a wide range of administrative functions such as course registration, student record management and storage, and personnel and financial management. The use of such systems has significantly improved these institutions' organizational agilities, allowing them to achieve and maintain a considerable degree of administrative and operational efficiency.

A basic recurring process in colleges and university administration is the generation of timetables for course offerings. In its simplest form, this course scheduling process assigns each course to a time slot and a classroom subject to a series of requirements and constraints [28]. These conditions typically include the type and size of the required classroom, instructor time availability, and the avoidance of timetable conflicts for courses students are required to take in the same semester, in addition to any other conditions relating to specific objectives that have been defined by the institution administrators. The scheduling problem created by such a set of circumstances clearly poses an interesting intellectual challenge.

Given the combinatorial nature of the problem and its computational complexity, which has been categorized as NP-hard, manual solution approaches based on trial and error are naturally considered to be inefficient in scenarios with large numbers of courses and classrooms [19]. In real applications, the methods used must address not only the computational complexity but also a series of interconnected factors that vary over time, such as the instructors' time preferences, course prerequisites, and the creation of new courses and study programs.

The problem of generating course schedules and classroom assignments is by no means a new one, having been studied extensively in the literature and addressed through various solution approaches based on operations research [17,27]. In practice, however, only a small number of these studies have been implemented as decision support systems (DSS). A series of interconnected factors underlie this dearth of real-world applications, including organizational resistance to change and the adoption of new technologies [4], organizational attitudes and lack of commitment [11], negative perceptions of the systems' user-friendliness [25], the degree of problem difficulty supported by the DSS [12,13], the need for a multi-disciplinary work team to perform system-related tasks [1,11,26], as well as the required levels of users' experience [22,24], training [12] and participation in the development of the system [11,26].

The present article describes a scheduling decision support system that is based on a Web architecture called *udpSkeduler*. It was implemented in the Faculty of Engineering (hereafter referred to as "the

* Corresponding author. Tel.: +56 2 978 3575; fax: +56 2 635 1679.
E-mail addresses: jmirandap@fen.uchile.cl (J. Miranda), pablo.rey@udp.cl (P.A. Rey), jmrobles@udd.cl (J.M. Robles).

0167-9236/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2011.10.011

Faculty") at Universidad Diego Portales (UDP) in Santiago, Chile, where it has generated a series of benefits that will be described in the following sections. The system uses an integer-programming model in order to define optimal schedules, which simultaneously incorporates all of the problem's requirements and constraints, as well as any additional objectives that are set by the Faculty Board of Directors. Further benefits of the *udpSkeduler* implementation include a reduction in the time that is required for scheduling due to process automation, improvement in schedule quality with fewer human errors and timetable conflicts, the ability to easily and quickly explore and analyze multiple scenarios through a user-friendly interface, and a reduction in work load for planning staff, which frees them for other tasks.

The structure of this article is as follows. Section 2 reviews the literature regarding the role of computer-based support systems in course schedule decision-making at educational institutions. Section 3 introduces the Faculty's timetabling problem and the various stages of the course scheduling process. Section 4 describes the *udpSkeduler* system and its constituent modules. Section 5 presents the results of the implementation, focusing on quantitative evidence of the improvements that are provided by the system. Finally, Sections 6 and 7 discuss the computational experiments that were conducted with the system and present our conclusions.

2. The importance of computer-based systems based on mathematical models in higher education course scheduling

Decision support systems based on mathematical programming models have been supporting course-scheduling processes for more than two decades [3]. The most commonly used solutions employ linear and integer programming techniques. The study by [7], a seminal work in this area, offers a theoretical analysis of a series of optimization problems that have inspired a variety of real-world problem applications. Solution approaches that are built around integer programming are presented in several studies such as [5,6,8,20,21]. Most of these works propose models in which the individual weekly sessions for each course are separately assigned, and the time slot configuration conditions are imposed by applying constraints. In the model to be presented here, the course sessions are assigned simultaneously to a single pattern made up of two components: a set of timeslots and a classroom.

Indeed, most of the above-mentioned studies concentrate on theoretical development and do not analyze an actual computer-based implementation; however, some researchers have integrated their solution approaches into practical applications by including decision support systems for the timetabling process. In 1989, [3] developed a computer-based system for a scheduling problem at an adult education institute that delivers initial solutions with relatively few course conflicts in a matter of minutes which can then be manually fine-tuned. They report that this iterative process produces a satisfactory schedule in no more than an hour.

In 1993, Johnson [16] implemented a computer-based scheduling system at the Loughborough University Business School that efficiently manages course information through a range of queries to a database. One of the principal benefits for the School has been the greater standardization of the process, which has been completely automated. Similar benefits are achieved in the system we propose here through partial structuring and automation of the process and the efficient management of information by means of database filters. These filters organize the information to fit the structure of the pattern assignment variable in the integer programming model, mapping the patterns within a reduced solution space and eliminating all a priori infeasible combinations.

A computer-based system that was implemented by Ferland and Leurent in 1994 [9], known as *SAPHIR*, adopts a heuristic approach to solve an integer-programming model, and provides an interactive optimization procedure that is similar to that in [3].

Stallaert [28] developed a computer-based course scheduling system for Anderson School of Management at the University of California at Los Angeles (UCLA) that generates schedules in two stages. The first stage exclusively deals with the core courses whereas the second schedules all of them. This allows the system to generate multiple versions of the core course schedule reflecting different criteria. A sensibility analysis can then be conducted and the best version can be chosen, with the end result being a better overall schedule. The method also integrates information management and report-generating functions that facilitate the process of fine-tuning the timetables.

Another decision support system for course scheduling, known as *SlotManager* [10], follows an approach that was originally designed by Liang et al. [18] by using a three-component structure that consists of interfaces, databases, and an optimization model. This system features a user-friendly interface and functionalities for obtaining information on course registrations, available classrooms, and course characteristics. The general structure of our *udpSkeduler* system is derived from this work but includes an additional module that supports direct interaction with the teaching staff, capturing instructors' time availability for giving classes and related preferences.

Dimopoulou and Miliotis [8] report on a computer-based system that was developed at the Athens University of Economics and Business that utilizes both a mathematical programming model and heuristic procedures in order to arrive at definitive course and examination schedules. The system is structured around five modules: a data module that manages information, a control system module that includes the user interface and generates data to be fed to the models, an optimization module that solves the mathematical programming model and executes the heuristics, a report generating module that reports on class timetables for use in decision-making, and an evaluation module that examines the quality of the generated schedules. According to the authors, this system provides better timetables for the students, a better use of classrooms, and a satisfaction of instructors' time preferences.

In 2002, Hinkin and Thompson [14] describe a computer-based system they developed, known as *SchedulExpert*, that uses an integer-programming model to generate course schedules for Cornell University's School of Hotel Administration. The principal benefits of this system are that it defines schedules that meet the objectives of the institution's governing body and reduces the time that is needed to produce them. In particular, it eliminates timetable conflicts between required core courses and certain sets of elective courses while, at the same time, minimizes conflicts between certain electives. Finally, this model also supports instructor course assignment decisions.

Miranda [19] presents a computer-based system named *eClasSkeduler* for courses that are given by the Executive Education Unit at the Universidad de Chile. Because of the way courses are taught by the Unit, with different course startup dates and varying course durations, the problem the system must solve requires that each week be individually and simultaneously scheduled. It therefore departs considerably from the problem addressed in the present work in which a single weekly schedule is replicated for an entire semester. Furthermore, although *eClasSkeduler* uses a course pattern assignment variable, the pattern structure differs from the one to be used here in that the former is defined by three components: a set of time slots, a classroom and a period of consecutive weeks over which the course is taught. As for the reported benefits of *eClasSkeduler*, they include a reduction in off-site classroom rental costs, fewer course conflicts, and efficient classroom use.

Finally, we note that all of the surveyed papers of Schaerf [27] and Lewis [17] clearly illustrate the significant benefits of incorporating computer-based systems into the decision-making process for course scheduling.

The *udpSkeduler* contains three important elements that distinguish it from the systems discussed above. The first element has to do with systematizing the capture of instructors' time availabilities for giving classes and associated preferences through a user-friendly interface. The interface allows instructors to interact directly with the system, thus streamlining the process of inputting this key information. The second element is the formulation of an integer programming model with decision variables based on the assignment of courses to patterns for the scheduling problem of an institution's undergraduate programs. Finally, the third distinguishing element is the efficient management of the database storing the system input information.

3. The research problem

The Faculty at *UDP* offers four undergraduate programs: industrial engineering, computer and telecommunications engineering, civil engineering, and civil engineering technology. The first three academic programs are normally completed in 12 semesters and the fourth program in 10. Each engineering program requires the students to take a relatively rigid set of courses that are distributed over the duration of the program according to curricula that define which courses must be offered in each semester and which courses must not be scheduled in the same time slots. Many of the courses are common to more than one curriculum, thus increasing the potential for time conflicts that must be avoided.

Approximately 300 courses must be scheduled each semester. The various sessions (classes) of a single course may be of three types: lectures, laboratory classes and tutorials (recitations), the first two considered as one for scheduling purposes. Lectures and lab classes are taught by instructors previously assigned to courses whereas tutorials are given by teaching assistants. The exact number of courses that are taught from one semester to the next depends on a range of factors, such as course demand, instructor time availability, and the creation of new courses. Each course consists of one, two, or three 90-minute lecture sessions that can be scheduled in four different weekly time slot configurations: 1) one session at any time; 2) two sessions, given on different days at the same time; 3) two consecutive sessions given on a single day; and 4) three sessions, given on three different days at the same time. The sessions for courses following the fourth configuration may be held either on Mondays, Wednesdays, and Thursdays or Tuesdays, Wednesdays, and Fridays.

An interesting element that contributes to the complexity of the problem is that certain multisession courses must alternate between different classroom types. For example, a physics course might have one session in a typical classroom for theory lectures, whereas the second session is held in a laboratory for conducting experiments.

The Faculty's infrastructure consists of 45 classrooms; however, the number that is actually available for courses changes from semester to semester because of the various nonteaching activities such as, seminars, conferences, and workshops, which compete for the use of space. There are five types of classrooms: lecture rooms, physics labs, computer labs, civil engineering labs, process simulation labs, and auditoriums. These spaces may be further differentiated by size and installed equipment.

There are 270 instructors on the Faculty's staff, who are categorized by their full-time or part-time status. Part-time instructors are well-known practitioners from industry who give professionally oriented courses whereas full-time instructors are devoted to teaching and research. Both the number of instructors and their time availabilities vary from semester to semester.

Before the implementation of *udpSkeduler*, the course scheduling process was handled by a team of three planners who generated the timetables and a coordinator who exclusively worked on assigning classrooms. The previous scheduling process was manual and used the schedule from the previous semester as a base format. Errors were frequent, causing numerous scheduling conflicts that were primarily caused by a tendency to assign too many courses to the same time slots, thus, overwhelming the supply of available classrooms. As a result, some courses had to be assigned to external (off-site) classroom space, which presented an awkward situation that led to a series of administrative headaches.

3.1. The scheduling process and course registration

The scheduling process for a given semester consists of a series of stages that involve the interaction of all of the Faculty's departments and other academic units. The *first stage* is devoted to gathering all of the information that is needed to decide the schedules. It begins with the identification of the available classrooms and the courses that will be offered that semester. Then, for each course, the corresponding academic unit specifies the technology and classroom requirements and the instructors' time availabilities. An estimate of course demand is also made using a procedure that is based on historical registration data and failure rates.

Once all of this information has been collected, the process enters the *second stage*. Here, a set of preliminary schedules is compiled. Each academic unit's schedule planner independently defines the schedules for that unit's courses, incorporating the information on each instructor's time availability. The three planners (one for each program, with civil engineering and engineering technology are considered to be a single unit) and the coordinator frequently meet in order to harmonize their respective schedules, eliminating any classroom "double-bookings" or timetable conflicts for courses that are common to more than one program grid. Most of these scheduling inconsistencies are due to the non-centralized nature of the decision-making process.

The objective of the *third stage* is to capture the real demand for each course. The preliminary course schedule is published and students register in the various offerings during a four-day registration period. This is followed by the *fourth stage*, in which a new and definitive classroom assignment is conducted. All four stages in the timetabling process are shown in Fig. B.1. Stages II and IV are automated by *udpSkeduler*, as described in the next section.

4. Description of the system

udpSkeduler is a decision support system that is based on mathematical programming models [18]. Built in a Web environment, it generates optimal course schedules by taking into account all of the constraints and internal Faculty policy objectives.

The *udpSkeduler* architecture consists of five main modules, which are depicted in Fig. B.2. The first is the user interface module, which is a mechanism for checking and adjusting the system's parameters and options and for managing information. Through this interface, the user can define the program grids and specify each course's characteristics in terms of its technology and infrastructure requirements. Various reports relating to student record management can also be generated.

The second module is the data input module, which stores all of the information that is necessary for generating the course schedules in a relational database. Some examples of these data are instructor time availability, classroom type, program grids, and course characteristics. The schedules for previous semesters are also stored here. Various filters are provided that quickly discard infeasible assignments, thus reducing the computation time that is required to find a solution. One of the filters prevents a course from being assigned to a classroom with insufficient seating capacity, whereas another filter prevents a course from being assigned to a time slot that is outside of the instructor's time availability. Fig. B.3 depicts the structure of the database with its main tables.

The third module is the time availability module. As one of the most important elements of the *udpSkeduler* system, it supports interaction between the instructors and the Faculty's planners. Through a user-friendly interface that can be accessed via the Internet, instructors can enter their personal accounts and specify their time availability. As can be seen in Fig. B.4, the interface for an account displays a table that represents the various time slots. The slots for which the instructor has already indicated availability are displayed in green, and the currently selected ones are shown in red. Timetable preferences can be expressed by ordering the slots by priority from the most preferred to the least preferred (see the dashed-line box in Fig. B.4). If they wish, instructors can also indicate that they have no preferences.

The fourth module is the optimization module, which contains the code for the integer-programming model. The model is solved using the *CPLEX* software package [15]. Finally, the fifth module is the report model, which produces a series of management reports after each execution of the optimization module. These reports present a series of indicators for evaluating the schedules that have been generated by the module, such as the average classroom use, the number of classrooms available for each course session, and course conflicts, as well as model indicators, such as the objective function value, the number of variables and constraints, and the solution computation time.

4.1. The optimization model

The integer-programming model (see Appendix **A**) assigns each course a unique timetable pattern defined by a vector that contains two components, one representing a set of time slots (one for each session) and the other representing a classroom. For example, if course *c* is assigned the pattern (*Monday.A–Thursday.A, FDI315*), it will be given on Mondays and Thursdays in slot A, and both sessions will be held in room *FDI315*. A binary variable x_{cp} is defined to take the value of 1 if course *c* is assigned to pattern *p*, and is 0 otherwise. This use of binary decision variables to assign a course to a unique pattern differentiates the model from those that use binary variables for assigning each course session to a time slot and constraints requiring the individual sessions to be given in predefined sets of time slots [5,6,9,20,28].

The advantage of a pattern-based formulation is that its linear relaxation is tighter [21], thus facilitating the solution of the model by commercial solver packages such as *CPLEX* [15]. The greater tightness is due to the fact that for each linear relaxation solution, an equivalent solution can be constructed that assigns individual course sessions. This relationship does not necessarily work both ways, however, as there exist feasible solutions of the model relaxation that assign sessions but which have no equivalent solutions for the pattern-based model. As an example, if we consider the solution of the linear relaxation of the model in which the variable that assigns course *c* to pattern (*Monday.A–Thursday.A, FDI315*) takes a value of 0.7, the equivalent solution for the relaxation of the individual course session model is constructed by assigning the same value to the variables that assign each session. In other words, a value of 0.7 is assigned to both the variable that assigns course *c* to slot (*Monday.A*) and classroom *FDI315* and the variable that assigns course *c* to slot (*Thursday.A*) and classroom *FDI315*. If, on the other hand, we consider a fractional solution of the individual session model in which the variable that assigns course *c* to slot (*Monday.A*) and classroom *FDI315* takes a value of 0.7 while the variable assigning course *c* to slot (*Thursday.A*) and classroom *FDI315* takes some other value such as 0.2, no equivalent solution for the linear relaxation of the pattern-based model can be found.

For each course *c*, a feasible pattern set exists within the total set of patterns. A feasible pattern for a given course is one for which the previously assigned instructor has time availability and the number of registrants is less than or equal to the classroom capacity. The definition of the feasible pattern subset is determined using the database filters, and the total number of feasible patterns may exceed 100,000 in some cases. For larger instances, however, explicit generation of the set of feasible patterns will produce too many sets and the problem will not be computationally solvable. To address such situations, a dynamic pattern generation approach could be employed by implementing a branch-and-price algorithm [2].

The model's objective function contains five terms that represent the various criteria of concern to the Faculty — see Eq. (A.1) in the appendix. The first term attempts to minimize the use of the Faculty's auditoriums. The second term minimizes the assignment of tutorial sessions outside the preferred times, reflecting the internal Faculty policy of scheduling all tutorials to the best possible extent on a single day of the week. The third term minimizes the number of timetable conflicts in order to ensure more flexibility for students who are repeating courses or taking courses ahead of the normal program grid sequence. The fourth term minimizes the number of external classrooms. Finally, the fifth term attempts to maximize the sum of the time preferences that have been entered by instructors in the time availability module. Each indicated preference is weighted by a parameter that assigns a value of 10 to the most preferred time slots and 1 to the least preferred. A value of 1 is assigned when an instructor indicates no preference.

The following constraints are included in the model: each course must be assigned a unique pattern, no courses taught by the same instructor may conflict, no classrooms may be assigned to more than one course in the same time slot, and no courses that are normally taken in the same semester (according to the program grid) may conflict; however, in practice, timetable conflicts cannot be completely eliminated, so they are penalized by the third term of the objective function.

5. Results of the implementation

In this section, we review the results of the Faculty's use of *udpSkeduler*. The system was implemented in two phases over the two semesters of 2010. The last entirely manual scheduling process, which was conducted for the second semester of 2009, will be our point of comparison. Table B.1 summarizes the elements that were involved in the analysis of the system's effectiveness. Table B.2 describes the performance indicators that were used to compare the qualities of the schedules that were generated for the three semesters.

In the first phase of *udpSkeduler* application (semester 2010-01), the course timetables were defined by the Faculty's planners using the traditional manual procedure, with *udpSkeduler* employed only to automate the assignment of classrooms in the final stage (Stage IV) of the process (see Fig. B.1). In the second phase (semester 2010-02), on the other hand, *udpSkeduler* was utilized to define both the course timetables and the classroom assignments, thus automating Stage II as well as Stage IV.

Table B.3 presents the results for the three semesters of our comparative analysis as measured by the aforementioned performance indicators. The results demonstrate that the *udpSkeduler* system produces substantial improvements over the fully manual approach. Furthermore, they show that the amount of improvement grows with the degree of involvement of the system in the scheduling process.

More specifically, a comparison of Case 1 to Case 2 (the last fully manual schedule and the first phase of *udpSkeduler* implementation, respectively) indicates that *udpSkeduler* generated a more efficient classroom assignment. One of its primary successes was to reduce the number of courses for which weekly sessions were assigned to more than one classroom; the incidence of such situations decreased from 49 to 0 (32% of the total) when using *udpSkeduler* instead of manual scheduling. In addition, the use of auditoriums decreased from 8 to 5 and the use of external classrooms decreased from 17

to 11; however, because *udpSkeduler's* involvement was confined to Stage IV, the timetables themselves were still defined manually without regard to classroom availability, and the complete elimination of auditorium and off-site space utilization was not possible. Thus, in certain time slots, classrooms double-booking did occur. As for the improvements in non-preferred tutorials and course conflicts, these were achieved by better course scheduling by the Faculty's planners.

If we now compare Case 1 (fully manual) to Case 3, the second phase of the *udpSkeduler* implementation, in which its use was extended to two stages (II and IV) of the scheduling process, further significant improvements in the performance indicators are evident. The use of auditoriums was completely eliminated, no course had sessions that were assigned to more than one classroom and no external classrooms were needed. This last result was the most important one, because Faculty planners had not managed to assign all courses to rooms on site since 2008. As for the other indicators, non-preferred tutorial times were reduced from 11 to 3, and course conflicts were cut from 21 to 6. These outcomes demonstrate that centralizing decision-making can achieve global optima, with highly efficient assignments of infrastructure resources.

6. Discussion of results

Each of the terms in the optimization model's objective function is weighted by order of importance. The weights themselves were directly decided by the Faculty Board members. The most important objective is the minimization of external classroom use, followed by the minimization of timetable conflicts and then the minimization of auditorium use. The minimization of non-preferred tutorial times and of instructor time preferences was, respectively, fourth and fifth in the importance ordering. The definitive objective function was, therefore, a weighted sum of the objectives that were established by the Faculty Board. The final schedules were validated and approved by the Board, thus completing the integration of the *udpSkeduler* system into the overall schedule generation process.

The process of finding the best configuration for the objective function weights was not explored in this study. Various studies have suggested methodologies determining the optimal objective function weights [23,29], and this continues to be a fertile area for additional research and improvement. The present authors intend to pursue this line of inquiry in future work.

7. Conclusions

A Web-based, course-scheduling and classroom-assignment system using an integer-programming model was developed for the Faculty of Engineering at Universidad Diego Portales. Known as *udpSkeduler*, the system has generated satisfactory course timetables and classroom space assignments on the Faculty's site. The schedules and assignments have fulfilled policy goals such as the use of the same classroom for each weekly session of every course, the elimination of the need for external classroom rentals an end to courses given in Faculty auditoriums.

The results of the *udpSkeduler* implementation demonstrate that integration of the system into the course scheduling process has led to multiple benefits. First, the time required for certain stages of the process was significantly reduced from weeks to a couple of hours. Second, certain stages of the process were successfully automated. Third, course timetable conflicts and other human errors generated by the previous manual process were eliminated. Fourth, the system's mathematical programming model can analyze a large number of solutions and find the optimal solution. Fifth, multiple scenarios can be quickly and simultaneously evaluated in terms of several different objectives by means of simple variations in the system parameters, thus reducing uncertainty in the scheduling decisions and allowing contingency plans to be generated. Sixth, at the organizational level a new communication channel was created between Faculty instructors and planners that accelerates and simplifies the exchange of information, especially in regard to instructor time availability and timetable preferences. Finally, the rapidity of the scheduling process using *udpSkeduler* has allowed planners to devote more time to revising and fine-tuning definitive schedules and other tasks.

Acknowledgments

The authors would like to thank academic and administrative staff members Alejandro León, Claudia Salgado, Claudio Gutiérrez, Ximena Geoffroy, Sheylla Pinto, and Beatriz Pezoa for their assistance in gathering and validating the information that was used in this study. They also owe a debt of gratitude to Pablo Morales for his contribution to building the *udpSkeduler* system interfaces and data models.

This project was partially financed through funds granted by the Faculty of Engineering at Universidad Diego Portales. The first author is grateful for the financial backing received from the Millennium Science Institute "Complex Engineering Systems" (ICM: P-05-004-F), which is a center of research excellence that is supported by government funding.

Appendix A. The mathematical model

We began our exposition of the model by defining and explaining the necessary notation. The set of courses is C, the set of classrooms is S, the set of instructors is L, the total set of time slots is T, a feasible set of timetable patterns is P and the set of such patterns for a given course $c \in C$ is $P(c)$.

A timetable pattern (hereafter simply "pattern") is a vector containing two components, one representing a set of time slots and the other a classroom where the sessions (classes) of a course are to be given. A feasible pattern p for a course $c \in C$ is one that satisfies the following conditions:

1. The classroom capacity is high enough to accommodate the number of students in the course.
2. The classroom is of the right type for the course (all classroom spaces are classified into lecture rooms, various types of laboratories and auditoriums).
3. The set of time slots follows the weekly time slot configuration defined for the course.

There are two types of course sessions, lectures and tutorials (the former including lab classes), which may be scheduled according to different patterns. Of the three sets of decisions variables in the model, the first two are binary and are activated if the lectures or tutorials of course $c \in C$ are assigned to pattern $p \in P$. More specifically, for each $c \in C$ and pattern $p \in P$ we define the following variables:

$$x_{cp} = \begin{cases} 1 & \text{if course } c \text{ lectures are scheduled according to pattern } p, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$y_{cp} = \begin{cases} 1 & \text{if course } c \text{ tutorials are scheduled according to pattern } p, \\ 0 & \text{otherwise.} \end{cases}$$

The third set of decision variables contains non-negative integer counter variables that register the number of timetable conflicts among groups of courses students are required to take in the same semester. The set of these course groups is denoted R. For each group $r \in R$ we define a counter variable z_{rt} that registers the number of timetable conflicts among the courses in that group for time slot $t \in T$.

The foregoing notation and other considerations are reflected in the final specification of the integer programming model, which is written as follows:

$$\min\ z = C_1 \sum_{p\in P(\text{aud})} \sum_{c\in C} x_{cp} + C_2 \sum_{p\in NM} \sum_{c\in C} y_{cp} + C_3 \sum_{t\in T} \sum_{r\in R} z_{rt} + C_4 \sum_{p\in P(\text{ext})} \sum_{c\in C} \left(x_{cp} + y_{cp}\right) - C_5 \sum_{l\in L} \sum_{c\in C(l)} \sum_{p\in P(c)} g_{lp} x_{cp} \quad \text{(A.1)}$$

s.t.

$$\sum_{p\in P(c)} x_{cp} = 1 \quad \forall c \in C, \quad \text{(A.2)}$$

$$\sum_{p\in P(c)} y_{cp} = 1 \quad \forall c \in C, \quad \text{(A.3)}$$

$$\sum_{c\in C(l)} \sum_{p\in U(c,t)} x_{cp} \le 1 \quad \forall l \in L,\ t\in T, \quad \text{(A.4)}$$

$$\sum_{c\in C} \sum_{p\in Q(c,s,t)} x_{cp} + \sum_{c\in C} \sum_{p\in Q(c,s,t)} y_{cp} \le 1 \quad \forall s\in S,\ t\in T, \quad \text{(A.5)}$$

$$\sum_{p\in U(c,t)} x_{cp} + \sum_{p\in U(c,t)} y_{cp} \le 1 \quad \forall c\in C, \quad \text{(A.6)}$$

$$\sum_{c\in r} \sum_{p\in U(c,t)} x_{cp} + \sum_{c\in r} \sum_{p\in U(c,t)} y_{cp} \le 1 + z_{rt} \quad \forall t \in T,\ r \in R. \quad \text{(A.7)}$$

The objective function of the model is given by Eq. (A.1). The five terms in the function represent five different objectives. The first one minimizes the use of auditoriums in the course schedules. For this purpose we define the set $P(\text{aud})$ to contain all of the patterns that use an auditorium as a classroom. The second term in the function minimizes the assignment of tutorials outside of the preferred time slots. The set of non-preferred slots is denoted NM. The third term in the function minimizes the number of timetable conflicts within each course group r. The fourth term minimizes the use of external classrooms. The set of all patterns assigning an external classroom is denoted $P(\text{ext})$. Finally, the fifth term maximizes the sum of the instructor time preferences over all Faculty instructors. The parameter g_{lp} defines the time preference of instructor l for pattern p and the set of courses taught by instructor l is denoted $C(l)$. Each objective function term has a coefficient that reflects its importance relative to the other terms. The values for these coefficients were defined by the Faculty Board.

Constraints in Eqs. (A.2) and (A.3) respectively impose that for each course $c\in C$, the lectures and tutorials are scheduled using a single pattern. This ensures that every course is or will be assigned a set of time slots and a classroom. The second set of constraints is expressed by Eq. (A.4). It attempts to avoid instructor time conflicts by prohibiting the assignment of more than one lecture to an instructor in a single time slot. Note that this constraint set does not apply to teaching assistants (responsible for tutorials and lab classes), who are not chosen until after the final timetable schedules are generated. The third set of constraints is defined by Eq. (A.5). Its role is to prevent the assignment of a classroom more than once to the same time slot. The set of feasible patterns for course $c\in C$ that assigns classroom $s\in S$ to time slot $t\in T$ is denoted $Q(c,s,t)$. The fourth set of constraints is given by Eq. (A.6). It imposes that the lectures and tutorials for course $c\in C$ may not be assigned to the same time slot $t\in T$. The set of feasible patterns for course $c\in C$ that contain time slot $t\in T$ is denoted $U(c,t)$. The fifth and last set of constraints is specified by Eq. (A.7). It imposes that to the extent possible, the courses students are required to take in the same semester are not scheduled in the same time slot $t\in T$. If such a conflict cannot be avoided, the z_{rt} term in the objective function is activated and penalized.

Appendix B. Tables and figures

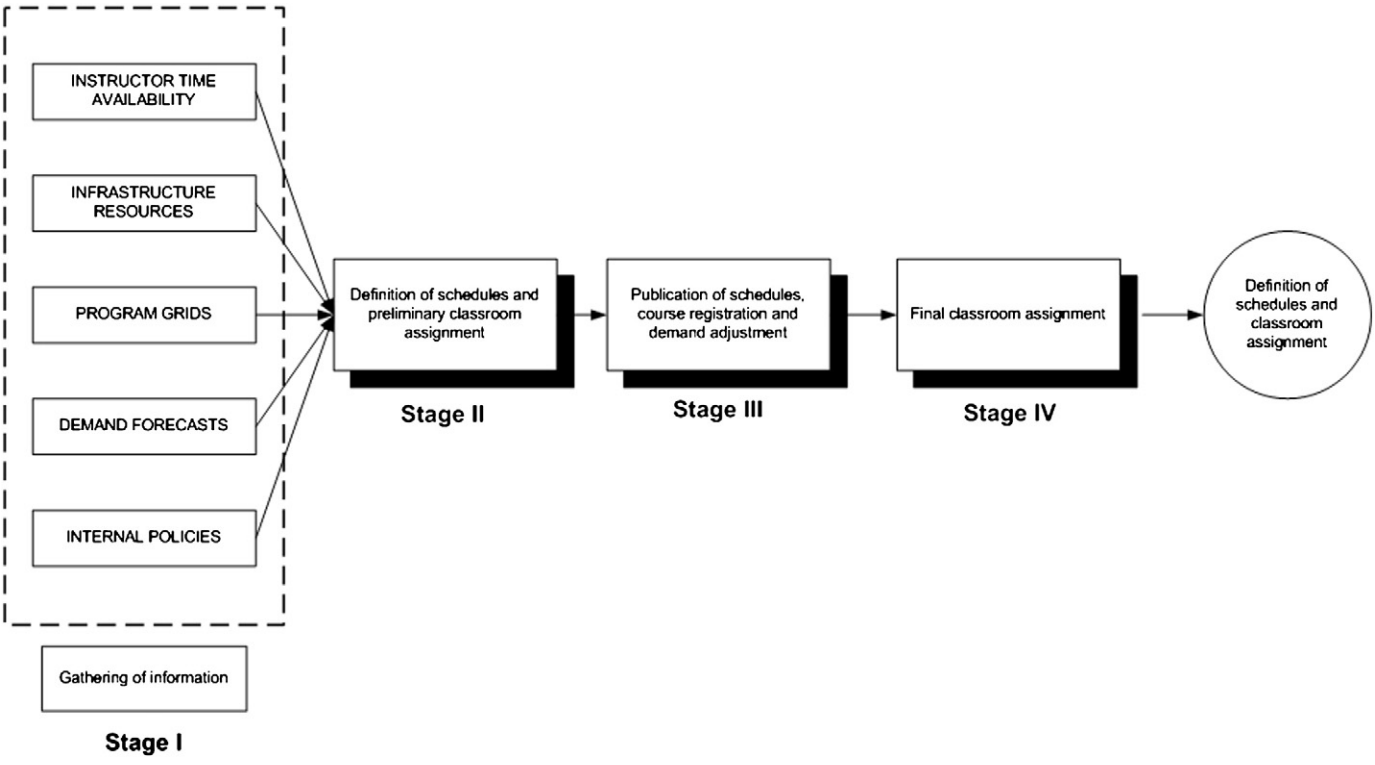

Fig. B.1. The four stages of the course scheduling process.

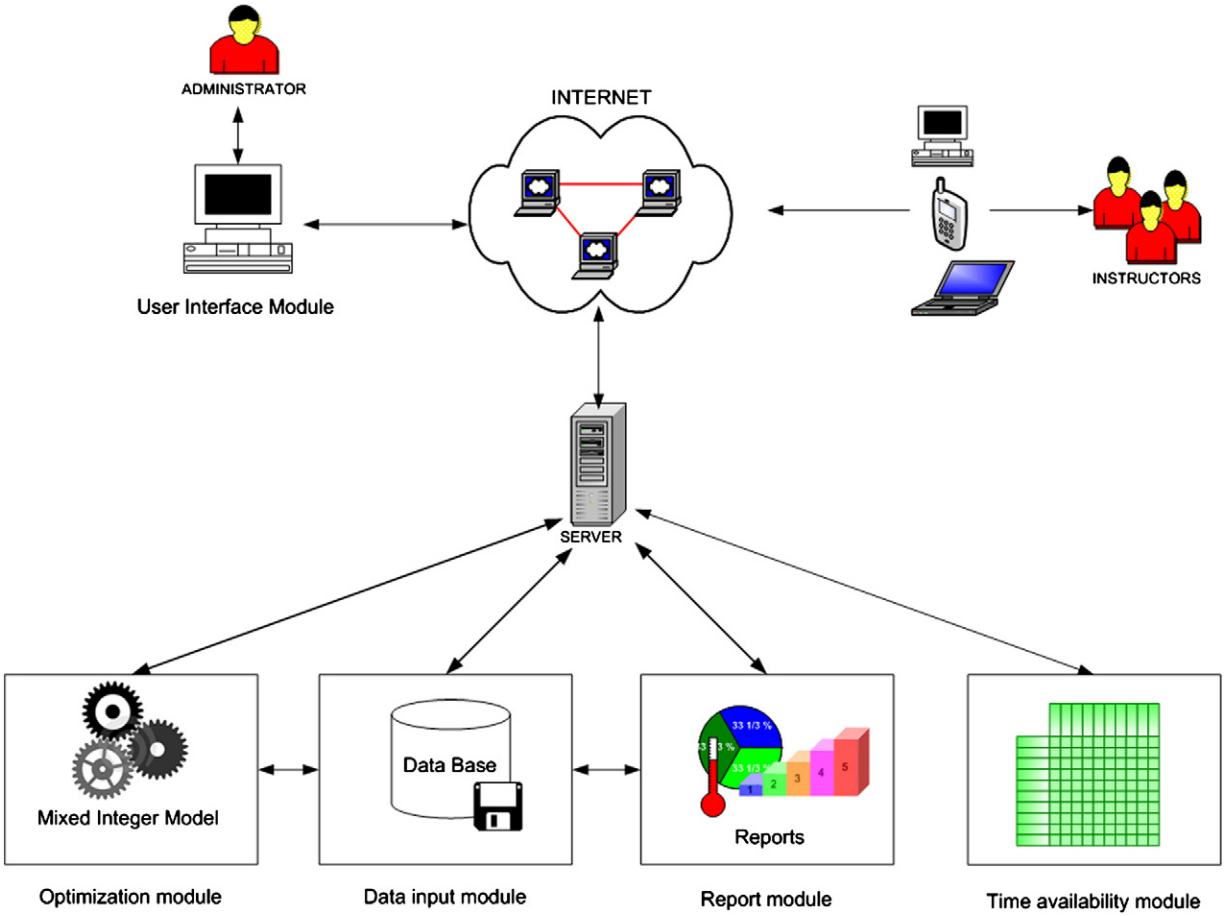

Fig. B.2. General architecture of the *udpSkeduler* system.

Table B.1
Characteristics of the three analyzed semesters.

	Semester		
	Case 1 2009-2	Case 2 2010-1	Case 3 2010-2
Classrooms	45	44	44
Courses	267	315	285
Course sessions	738	867	794
Instructors	248	259	255

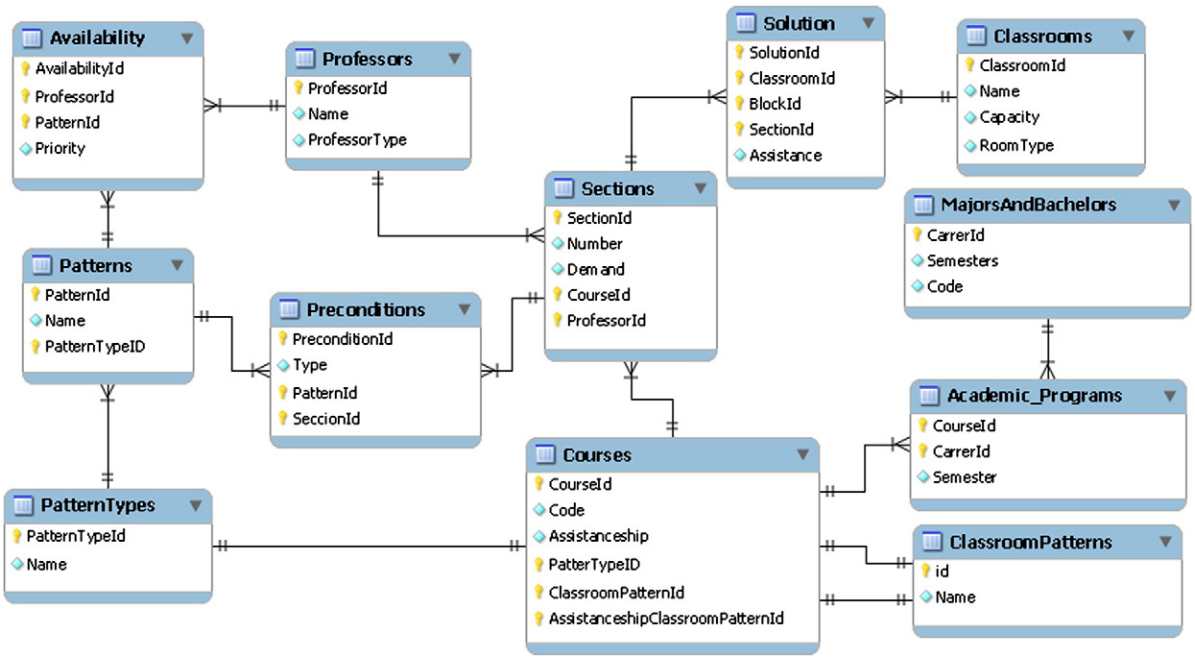

Fig. B.3. Architecture of the *udpSkeduler* database.

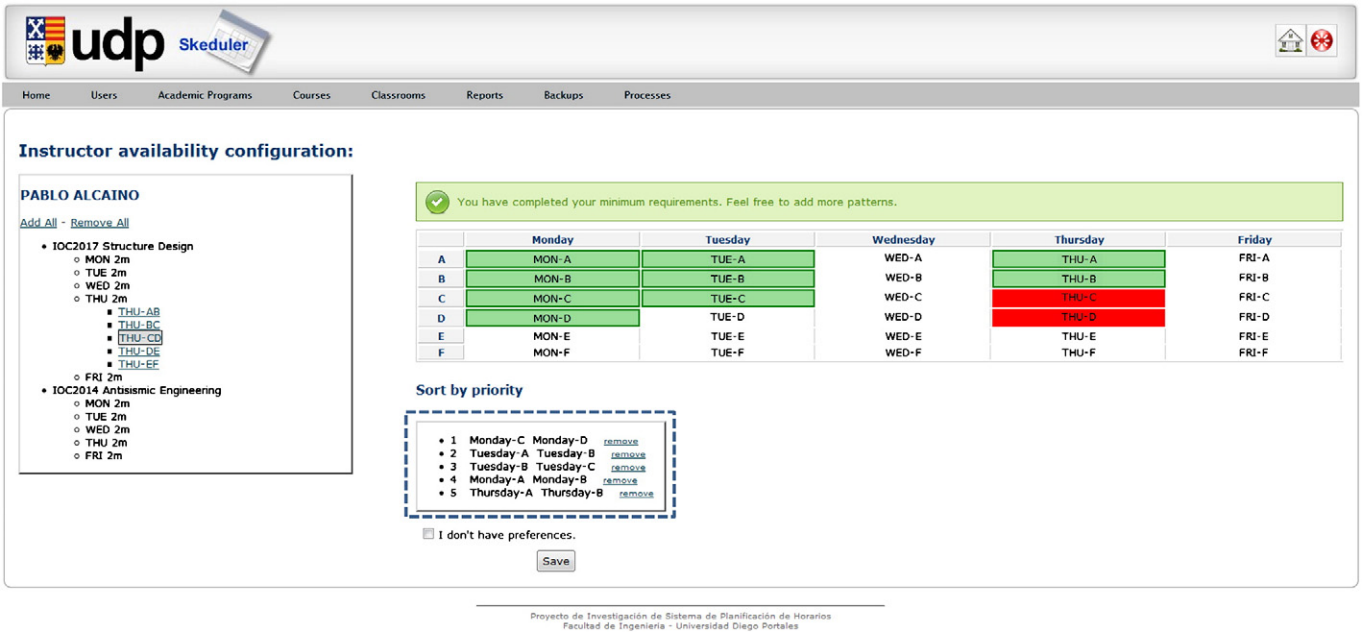

Fig. B.4. Screen shot of the instructor account interface for the *udpSkeduler* time availability module.

Table B.2
Scheduling performance indicators.

Performance indicator	Description
Use of auditoriums	Number of courses that are assigned to a Faculty auditoriums
Use of external classrooms	Number of courses that are assigned to external classrooms
Non-preferred tutorials	Number of courses with tutorials that are outside of preferred times
Different classrooms	Number of courses with sessions that are assigned to different classrooms
Course conflicts	Number of conflicts between courses that are normally taken in the same semester according to the program grid

Table B.3
Number of undesirable situations in semester course schedules.

	Semester		
Performance indicator	Case 1 2009-2	Case 2 2010-1	Case 3 2010-2
Use of auditoriums	8	5	0
Use of external classrooms	17	11	0
Non-preferred tutorials	11	8	3
Different classrooms	49	0	0
Course conflicts	21	13	6

References

[1] D. Bajwa, A. Rai, I. Brennan, Key antecedents of executive information system success: a path analytic approach, Decision Support Systems 22 (1998) 31–43.
[2] C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, P.H. Vance, Branch-and-price: column generation for solving huge integer programs, Operations Research 46 (1998) 316–329.
[3] N. Chantal, D. de Werra, Extension of coloring models for scheduling purposes, European Journal of Operational Research 40 (1989) 32–37.
[4] R. Cooper, R. Zmud, Information technology implementation research: a technological diffusion approach, Management Science 36 (1990) 123–139.
[5] S. Daskalaki, T. Birbas, Efficient solutions for a university timetabling problem through integer programming, European Journal of Operational Research 160 (2005) 106–120.
[6] S. Daskalaki, T. Birbas, E. Housos, An integer programming formulation for a case study in university timetabling, European Journal of Operational Research 153 (2004) 117–135.
[7] D. de Werra, An introduction to timetabling, European Journal of Operational Research 19 (1985) 151–162.
[8] M. Dimopoulou, P. Miliotis, Implementation of a university course and examination timetabling system, European Journal of Operational Research 130 (2001) 202–213.
[9] J. Ferland, C. Fleurent, SAPHIR: a decision-support system for course scheduling, Interfaces 24 (1994) 105–115.
[10] L. Foulds, D. Jonhson, SlotManager: a microcomputer-based decision support system for university timetabling, Decision Support Systems 27 (2000) 367–381.
[11] M. Ginzberg, Early diagnosis of mis implementation failure: promising results and unanswered questions, Management Science 27 (1981) 459–478.
[12] T. Guimaraes, M. Igbaria, M. Lu, The determinants of DSS success: an integrated model, Decision Sciences 23 (1992) 409–424.
[13] T. Guimaraes, Y. Yoon, A. Clevenson, Factors important to expert system success: a field test, Information Management 30 (1996) 119–130.
[14] T. Hinkin, G. Thompson, SchedulExpert: scheduling courses at Cornell University School of Hotel Administration, Interfaces 32 (2002) 45–57.
[15] ILOG, User's Manual CPLEX 9.0, ILOG S.A., 2003
[16] D. Johnson, A database apprroach to course timetabling, The Journal of the Operations Research Society 44 (1993) 425–433.
[17] R. Lewis, A survey of metaheuristics-based techniques for university timetabling problems, OR Spectrum 30 (2008) 167–190.
[18] T. Liang, C. Lee, E. Turban, Model management and solvers for decision support, Handbook on Decision Support Systems, Volume 1, Springer-Verlag, 2008, pp. 231–258.
[19] J. Miranda, eClasSkeduler: a course scheduling system for the Executive Education Unit at the Universidad de Chile, Interfaces 40 (2010) 196–207.
[20] S. MirHassani, A computational approach to enhancing course timetabling with integer programming, Applied Mathematics and Computation 175 (2006) 814–822.
[21] A. Qualizza, P. Serafini, A column generation scheme for faculty timetabling, in: E. Burke, M. Trick (Eds.), Proceedings of the 5th International Conference on Practice and Theory of Automated Timetabling (PATAT 2004), LNCS, 3616, Springer-Verlag, 2004, pp. 161–173.
[22] L. Raymond, Organization characteristics and mis success in the context of small business, MIS Quarterly 9 (1985) 37–52.
[23] G. Reeves, E.P. Hickman, Assigning MBA students to field study project teams: a multicriteria approach, Interfaces 22 (1992) 52–58.
[24] G. Sanders, J. Courtney, A field study of organizational factors influencing DSS success, MIS Quarterly 9 (1985) 77–91.
[25] K. Sandhu, Automating class schedule generation in the context of a university timetabling information system, Ph.D. thesis, School of Managementm, Griffith University, 2001.
[26] R. Santhanam, T. Guimaraes, J. George, An empirical investigation of ODSS impact on individuals and organizations, Decision Support Systems 30 (2000) 51–72.
[27] A. Schaerf, A survey of automated timetabling, Artificial Intelligence Review 13 (1999) 87–127.
[28] J. Stallaert, Automated timetabling improves course scheduling at UCLA, Interfaces 27 (1997) 67–81.
[29] D. Teodorović, E. Krčmar-Nožić, Multicriteria model to determine flight frequencies on an airline network under competitive conditions, Transportation Sciences 21 (1989) 14–25.

Jaime Miranda is an Academic of the Department of Management Control and Information Systems at the Universidad de Chile. He has a Bachelor's degree in Industrial Engineering and a Master in Operations Management at the Universidad de Chile. Currently he is a PhD candidate for the program of Complex Engineering Systems at the same University. Jaime Miranda's research interests include: scheduling the timetables of the different courses of the Faculty, sports programming, staff and location and vehicle routing problems. Professor Miranda designed and implemented the computer-based information system DSS (Decision Support Systems) in various companies and universities in Chile.

Pablo A. Rey is an Assistant Professor in the School of Industrial Engineering at the Universidad Diego Portales in Chile. He has a Bachelor's degree in Mathematics from the Universidad Nacional de Córdoba in Argentina and a PhD. degree in Electrical Engineering from the Pontifícia Universidade Católica do Rio de Janeiro in Brazil. Among his research interests are the application of integral programming techniques and simulation techniques for scheduling and timetabling, vehicle routing and, other transportation planning problems. Dr. Rey has collaborated with the design and the implementation of operations research based on methodologies in some Brazilian and Chilean organizations.

José M. Robles is an Academic and the Dean of the Faculty of Engineering at the Universidad del Desarrollo in Chile. He holds a Bachelor's degree in Industrial Engineering from the Pontificia Universidad Católica de Chile and a Ph.D. from the University of California, LA. He has been an Academic in the Department of Industrial and Systems Engineering at the Pontificia Universidad Católica de Chile. Professor Robles has a vast consulting experience besides participating in the development of computing systems in the areas of operations, logistics and simulation. Furthermore, he has advised the World Bank and the BID (Banco Interamericano de Desarrollo) in the field of education in many Latin American countries. His research interests include finance, optimization, operations research and technology.