

Adaptation and Generation in a Web-Based Tutor for Linear Programming

E. Millán, E. García-Hervás, E. Guzmán, A. Rueda, and J.L. Pérez-de-la-Cruz

Departamento de Lenguajes y Ciencias de la Computación. ETSI Informática.
Universidad de Málaga. SPAIN.
eva@lcc.uma.es

Abstract. We present TAPLI, an adaptive web-based learning environment for Linear Programming. TAPLI is a set of adaptive tools offered in a web-based learning environment: a) an adaptive hypermedia component, that is responsible of presenting the learning contents; b) a testing component, based on Item Response Theory and c) a *drill-and-practice component*, which adaptively generates exercises and coaches the student while solving them.

1 Introduction

The development of adaptive web-based educational systems (AWES) can be of great help in the teaching and learning process, providing world-wide accessibility and timeless availability. Along these lines, the main goal in the research work presented here was to use the web and adaptive technologies to create TAPLI (*Tutorial Adaptativo de Programación Lineal* -Adaptive Tutorial for Linear Programming, in Spanish). The system is available at <http://www.lcc.uma.es/tapli>.

TAPLI is built upon previous work developed by our team for systems like EPLAR [4] (a problem solver for simplex problems), ILESA [3] (a system capable of generating problems at the right level of difficulty based on the student model and also of coaching students while solving a problem) and SIETTE [2] (a web-based testing tool available at <http://www.lcc.uma.es/siette>), that can be used by teachers to edit tests for any subject domain and by students to assess themselves on line).

This paper is structured as follows: in the next section we briefly present an overview of the system. In section 3 we focus on the web issues of TAPLI. The paper finishes with some conclusions and future research directions.

2 The TAPLI System

The TAPLI system (Figure 1) is in fact a set of components that have been integrated in a single learning environment: a) an *adaptive hypermedia component*, that is responsible of presenting the theoretical concepts and examples; b) a *testing component*, that evaluates the student knowledge on each of the topics in which the

domain has been decomposed, and c) a *drill-and practice component*, where students can solve problems (automatically generated and adapted to student's knowledge level) while being coached by the system. Together, all these tools constitute a complete web-based environment for learning Linear Programming. They are coordinated by an *instructional planner*, which is responsible of selecting the next component that will be used by the student in his learning process, by using the information provided by the corresponding *student model*.

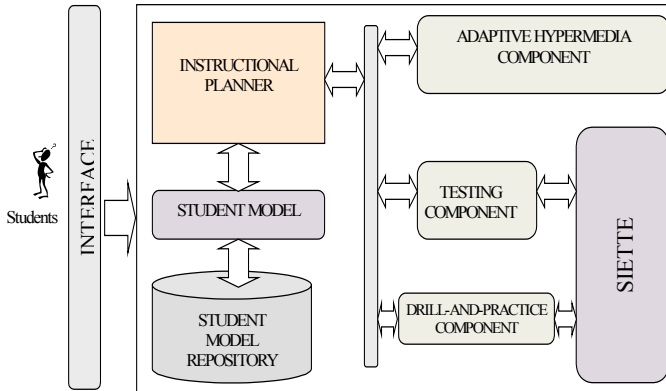


Fig. 1. The architecture of TAPLI.

The testing component can be used at any time during the interaction with the system. The action *Take a test* is also recommended by the system whenever it is advisable. For example, for new students (to check their state of knowledge), when the student reads a page (to infer if the knowledge presented has been acquired), or when the student enters the drill-and-practice environment (to be able to generate problems at the required level of difficulty for student's level of knowledge). Different tests have been defined for each of the six topics the subject has been divided into. After each topic is evaluated, a global grade is computed as a weighted average.

To use the SIETTE system to allow coached problem solving in a drill-and-practice environment, we have made use of the possibility to define questions by means of Java Applets. Generation of problems for web systems is an interesting issue that has already been discussed in [1]. The problem-solving procedure has been decomposed in TAPLI into *basic skills* and *types of problems* and different applets have been defined for each of the basic skills.

3 The Adaptive Hypermedia Component

This component has been designed so students that cannot attend lessons are able to learn the main concepts and techniques for Linear Programming. TAPLI is *adaptable* (some characteristics are adapted according to information provided by the student) and *adaptive* (some others are adapted based on information gathered by the system during student's interactions):

- *Adaptable characteristics.* When the student connects to TAPLI, she has to provide some information (name and password) so the system can create her student model. Also some personal learning characteristics are included in the model like topics of interest (learning goals), and type of knowledge that she wants to acquire (practical vs. theoretical-practical). According to this characteristics, the system will adapt to the student. Learning goals are used to adapt the suggested navigation structure (*i.e.*, the recommendations provided by the system), and also to hide links to tests that evaluate topics that are not learning goals. However, text pages related to topics that are not currently the focus of attention are not hidden. In this way, the student can take a look at them and change his/her learning goals accordingly. Stretch-text techniques have been used so the system can adapt to the type of knowledge required by the student. In this way, the student interested only in problem solving will only see links to the theoretical proofs of the theorems (that can be followed if desired). It is important to say at this point that TAPLI has been implemented as an open environment. This means that it has the ability to recommend students which is the best thing to do at each moment (read a page, solve an exercise, take a test, see an example) but does not force the student to follow the these recommendations. In this way, students can explore the environment with whole freedom.

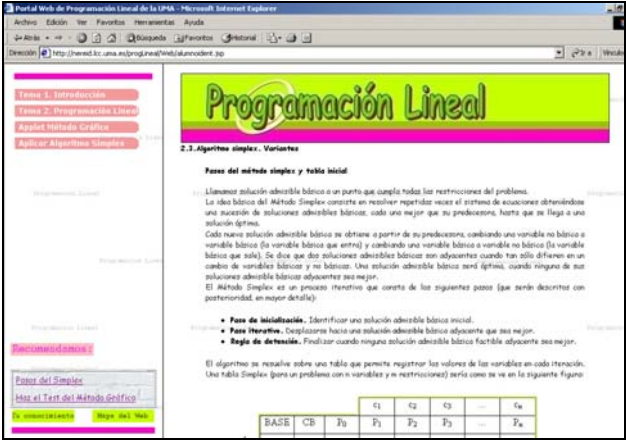


Fig. 2. Snapshot of TAPLI interface

- *Adaptive characteristics.* Some features are adapted according to relevant information inferred by the system, as the list of visited pages and student’s knowledge level. The list of visited pages is used to suggest which page the student should see next. Student’s knowledge level is estimated by the SIETTE system after the student takes a test or solves a coached exercise, and kept in the student model. It is used to suggest which kind of action the student should take next (take a test/solve an exercise). Once the student has visited all the pages, the next page to be visited is selected according to the information contained in the student model. As an example, we show in Figure 2 a snapshot of TAPLI interface during a session. The interface is divided into two different parts: on the right there is a frame that presents; the curriculum structure, a window for recommendations and two

informative buttons: *Tu conocimiento*, that allows students to inspect their student model and change his/her learning goals; and *Mapa del web*, which shows a statistics of visited pages. For this particular student, the system offers two recommendations: what to *read* next (*Pasos del Simplex*, i.e. Simplex steps), which is based on the visited pages; and what to *do* next (*Test del método gráfico*, i.e., take a test about the graphic method), which is based on the current estimation of student's knowledge level. On the left side, the learning contents of the selected topic of the curriculum are shown.

The communication of this component with SIETTE is accomplished through a pre-established protocol, based on URL calls from one system to the other with the proper parameters. The testing component sends to SIETTE the set of topics to be assessed, the current estimation of the student's knowledge, the URL to which the results will be returned and some additional parameters. Once the evaluation has finished, SIETTE will invoke to the URL indicated, passing the new estimated knowledge levels of the student.

4 Conclusions and Future Work

In this paper we have presented an adaptive web-based learning environment for Linear Programming. This system has three different educative components: an adaptive hypermedia component, a testing component, and a drill-and-practice component. The TAPLI system presents different kinds of adaptation: adaptive navigation by direct guidance (recommendations) and by link hiding, adaptive presentation by hiding parts of the text, adaptive testing and finally adaptive problem generation and personalized assistance. A fully working version of TAPLI is currently available at <http://www.lcc.uma.es/tapli>. However, further work is needed in two different directions: first, we would like to increase the range of adaptive techniques. Second, we plan to make a formative evaluation with students in order to see if TAPLI does indeed increase the rate of effective learning.

References

- [1]. Belmonte, M. V., Guzmán, E., Mandow, L., Millán, E., & Pérez de la Cruz, J. L. (2002). Automatic generation of problems in web-based tutors. In R. J. a. J. L. C. Howlet *Virtual Environments for Teaching and Learning* (pp. 237–277). World Scientific.
- [2]. Conejo, R., Guzmán, E., Millán, E., Pérez, J. L., & Trella, M. (To appear). SIETTE: A WebBased Tool for Adaptive Testing. *International Journal of Artificial Intelligence in Education*.
- [3]. López, J. M., Millán, E., Pérez J.L., & Triguero F. (1998). ILESA: A web-based Intelligent Learning Environment for the Simplex Algorithm. *Proceedings CALISCE'98*.
- [4]. Millán, E., Mandow, L., & Rey, L. (1999). EPLAR: Un entorno para la enseñanza de la Programación Lineal. En *Desarrollo profesional y docencia universitaria: Proyecto de innovación en la Universidad de Málaga*. Servicio de Publicaciones de la Universidad de Málaga.