Bridging the gap between industry, science and education: Industry experience to students, computer science methods to administration

Bert Van Vreckem, Dmitriy Borodin, Wim De Bruyn, Victor Gorelik, Sergey Zhdanov, Alexander Rodyukov

Abstract
In this paper, we consider two topics in higher education. The first one relates to the gap between education and the professional field. We propose to incorporate constantly changing industry experience into relevant courses. We illustrate this with an example of a curriculum audit in the faculty the authors are affiliated with. In the second topic, we focus on computer science techniques and business methodologies for automating the planning of course programs that can be used in software for course administration. We present mathematical models for planning courses in a bachelor program, and topics within courses, alongside techniques to implement these models and solve arising practical issues, a.o. analytical hierarchy processes, pairwise comparisons, approximation, and optimization. To conclude, we think it is important to realize that technological, social and cultural changes in the world, and particularly in higher education, necessitates rethinking the principles of planning course programs, and updating the ways for administrating education using up-to-date technologies.

Keywords: education, planning, mixed-integer programming, science, ICT

Introduction

Aligning education with industry
Aligning higher education with industry needs is an ongoing concern for colleges and universities, especially in fields that evolve rapidly, e.g. Information and Communication Technologies (ICT). This ‘gap’ between education and industry can be characterized from different points of view, and we illustrate this with a few examples and points for further discussion in this paper. First, we see a significant discrepancy between competences demanded by the labour market on the one hand and those acquired by college/university graduates on the
other (Sajid Sheikh, 2009, Froeschle, 2010). Additionally, some of the existing practically useful scientific research results rarely or even not all are applied in the real life (Dosi et al., 2006). In this paper, we share our experience with regard to bridging the gap between science, higher education, and industry. We present a case of planning an industry-friendly study programme in the field of ICT, and propose how to use computer science-based approach to automate this planning process.

Development of a study programme that reflects the industry needs with regard to graduates’ skills and competences has been widely studied and discussed (Liu Hong and Yunzhong Jiang, 2001, Nancy Lynch et al., 2007, Reif, 2007), and different solutions were suggested. Some companies started their own initiatives intended to educate specialists according to industry needs. Well-known examples are certification programmes like Microsoft Certified Systems Engineer (MCSE) or Cisco Certified Network Associate (CCNA). However, many industry professionals and educators alike question the quality of brand-aligned certifications, and these programmes do not train for more general competences required from employees (e.g. teamwork, communication, etc.).

**Responding to changing needs**

Teaching staff in rapidly evolving disciplines have to be on the lookout for relevant changes continuously, while keeping a delicate balance between following the latest trends and sticking to stable technologies that have proved their value. Methods, technologies, or paradigms that are state-of-the-art now may be obsolete within a few years by the time a student graduates. However, today’s hype will not necessarily ascend to tomorrow’s mainstream. In any case, regular updates to the curriculum are necessary in order to remain relevant.

Another issue in planning study programmes is global changes in systems of higher education, for example the Bologna declaration. For every institution this means that every single study programme has to be audited, and if necessary adapted, in order to get accredited under the new system.

Planning and organizing courses is tedious work and rapidly becomes intractable for larger institutions or departments. People responsible for this task do a terrific job with the limited resources available to them. More often than not, planning is largely a manual process, often based on implicit knowledge or “intuition”. This motivated extensive scientific research on
automating the planning process in education on different levels (see review Zhdanov et. al. 2009). The study programme planning problem was formulated as a mathematical model by one of the authors of this paper. It was developed for a Russian university, taking into account the peculiarities of the Russian local educational system, which currently does not completely comply with the Bologna agreement (Borodin, Tokarev, 2008, Borodina et.al., 2011). Moreover, it was suggested to research the programme planning problem both on the course level (strategic level) and deeper, on the competences level (tactics level) taking into account the relationships between competences, courses and course topics (Borodin, Tokarev, 2008). In this paper, an example of the strategic level model is provided together with a practical application.

The rest of the paper is structured as follows. Section 1 describes a case of converting the study program to fit the Bologna declaration and at the same time incorporating current industry requirements. In section 2 a mathematical model for planning a study programme is presented as a case of applying the results of a scientific research for the administration aid purposes; the practical contribution and particular features of the model and of the case are discussed. The conclusions propose statements for the future work and provide some open questions for discussion.

Conversion of the study programme “Graduaat Toegepaste Informatica” into “PrBach Applied Computer Science”

The Faculty of Business Information and ICT of University College Ghent (HoGent) organises two Bachelor-grade study programmes, viz. Applied Computer Science and Office Management. A few years ago, in light of the Bologna process to standardise higher education degrees, the old programme, called “Graduaat Toegepaste Informatica” (Graduate Applied Computer Science), had to be thoroughly reviewed and audited in order to have it accredited under the new system. This resulted in the programme “Professional Bachelor Applied Computer Science”, with a revised and updated curriculum, and detailed course descriptions compliant with the European Credit Transfer and Accumulation System (ECTS). In this section, we discuss the methodology used, focusing on how to systematically take industry needs into account in the curriculum.
In order to allow comparison of study programmes between institutions, a common reference level framework for competences and qualifications is needed, expressed as learning outcomes in terms of knowledge, skills, and competences (Winterton, et al., 2005). To prepare the new curriculum, we first looked at which competences an IT professional needs. We distinguish between following types:

- **general competences**, e.g. “being able to reflect critically”, “being able to acquire and process data”, “being able to communicate effectively”, “an attitude towards life-long learning”, “creativity”, etc. In Flanders, these competences are goals set by the government in legislation concerning higher education (Flemish Government, 2003)
- **general professional competences**, e.g. “being able to work in a team”, “being solution-oriented”, etc. These are also specified in Flemish legislation.
- **specific professional competences**, e.g. “development of computer programmes”, “maintaining database systems”, “system and network administration”, etc. These competences were not set by law, but compiled by the department and teaching staff.

In Flanders, several governmental and professional organisations have been gathering information on necessary competences for specific vocations. We used following sources to select the professional competences for our study programme:

- The “Competence and Occupations Directory for the Labour Market” (COBRA) of the Flemish Employment Service (VDAB)\(^1\)
- Professional competence profiles compiled by the Social and Economic Council of Flanders (SERV)\(^2\)
- Professional profiles from Agoria ICT, a section of the federation of the technological industry in Belgium
- The social network of teaching staff, specifically alumni, contacts with companies that accommodate internships for our students, and the “resonance committee” (an advisory

---


board in the department consisting of ICT professionals that provides continuous feedback on the curriculum, and industry needs)

All competencies gathered in this way were compiled in a list that was then refined by small task teams consisting of teaching staff, according to their expertise, and expanded. For every selected competence, we specified a number of indicators that show how the acquisition of the competence can be assessed (see Table I). This makes the assessment process verifiable and focused. Finally, all competences were assigned to courses, about three to four per course. In some cases, this resulted in redirecting focus of courses to more topical subjects, and even the introduction of new courses or removing courses from the programme.

Table I. Competences and indicators for the course Network and System Administration.

Some competences are evaluated through a classic written examination (a.o. basic knowledge), others through lab exercises (a.o. network server installation, and documenting the process). Also, a few times per semester students get a group assignment where they have to build a complex network (incl. services like e-mail) in team in the course of a day.

<table>
<thead>
<tr>
<th>Core competence 1. Installation, configuration, and security of networks and operating systems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Indicators</strong></td>
</tr>
<tr>
<td>● Knowing the basics of network and system administration</td>
</tr>
<tr>
<td>● Knowing the basics of network and system security, and being able to apply these</td>
</tr>
<tr>
<td>● Being able to track down and solve security risks</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Core competence 2. Configuration, administration and testing of complex network, telecom and server systems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Indicators:</strong></td>
</tr>
<tr>
<td>● Being able to map out, analyse, configure and test a network infrastructure in a mixed environment (Windows, Linux, Cisco).</td>
</tr>
<tr>
<td>● Being able to automate system administration tasks using scripts</td>
</tr>
<tr>
<td>● Being able to document configurations and procedures</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>General competence. Executing assignments with perseverance, responsibility, and stress resistance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Indicators:</strong></td>
</tr>
<tr>
<td>● Being able to systematically analyze and solve network problems (troubleshooting).</td>
</tr>
<tr>
<td>● Being able to set up a complex network infrastructure quickly and in team</td>
</tr>
</tbody>
</table>

Automated planning of a study programme

After incorporating some of the industry needs to the course list, we consider the problem of planning all the courses over the study period.
The goal is to automate the creation of a study programme from scratch taking into account all the available data about each course. In the more strict way, the general study programme planning problem can be formulated as follows. All the courses must be assigned to all the semesters in the way that all the requirements and limitations are satisfied and some target indicators are achieved. Depending on the system of education, the following requirements and limitations could be applied:

- number of credits per semester and per academic year
- number of hours per semester and per academic year
- precedence relations between courses - prerequisites for each course as described in the previous section

Examples of target indicators:

- Sum of all distances between interdependent courses should be minimized
- Total buffer for all courses should be minimized in the way that the courses that are not on the "critical path" could be shifted
- Number of credits per semester should be as close as possible to 30

To model this assignment problem, we make use of an Integer Programming (IP) formulation (Williams, 2009). IP is a mathematical method to determine the best outcome (e.g. maximum profit or minimum cost) of a certain linear objective function, given a number of constraints expressed as linear relationships. For reasonably sized problem instances, efficient algorithms exist that find an optimal solution, i.e. an assignment of values to all decision variables in the objective function and constraints, in reasonable time.

Now we provide a mathematical model of the problem. We use the following notation in the model:

- \( P_{i,k} = \begin{cases} 
1, & \text{if course } k \text{ is a soft prerequisite for course } i, \\
0, & \text{otherwise} 
\end{cases} \)

  \( Course \ k \ is \ a \ soft \ prerequisite \ for \ course \ i \) iff course \ k \ must be studied either in an earlier or in the same semester as course \ i ;

- \( H_{i,k} = \begin{cases} 
1, & \text{if course } k \text{ is a hard prerequisite for course } i, \\
0, & \text{otherwise} 
\end{cases} \)
- Course *k* is a **hard prerequisite** for course *i* iff course *k* must be studied in an earlier semester than course *i*;
- *c<sub>i</sub>* denotes the number of credits for course *i*;
- *C* is a constant that denotes the number of credits per academic year *y* (we assume an academic year consists of two semesters);
- *S* is a constant that denotes overall number of semesters in the programme;
- *N* is a constant that denotes overall number of courses in the programme.

We define the following decision variables:

\[
a_{i,j} = \begin{cases} 
1, & \text{if course } i \text{ is assigned to semester } j, \\
0, & \text{otherwise}
\end{cases}
\]

where *i* = 1..*N* and *j* = 1..*S*.

And the objective function:

\[
\text{Minimize } \sum_{j=1}^{S} \left( \frac{C}{2} - \sum_{i=1}^{N} a_{i,j} c_i \right)
\]

Subject to:

\[
\forall i = 1..N \quad \sum_{j=1}^{S} a_{i,j} = 1 
\]

\[
\forall i, k = 1..N \quad P_{i,k} \left( \sum_{j=1}^{S} j a_{i,j} - \sum_{j=1}^{S} j a_{k,j} \right) \geq 0 
\]

\[
\forall i, k = 1..N \quad \left| \sum_{j=1}^{S} j a_{i,j} - \sum_{j=1}^{S} j a_{k,j} \right| \geq H_{i,k} 
\]

\[
\forall y = 1..3 \quad \sum_{i=1}^{N} a_{i,2y+1} c_i - \sum_{i=1}^{N} a_{i,2y+2} c_i = C 
\]

The objective function (1) assures that credit assignments are balanced between semesters. Constraint (2) ensures that each course is assigned to exactly one semester. A course can’t be given before its prerequisite courses (soft prerequisites), which is expressed in constraint (3). In its turn, constraint (4) encodes hard prerequisites. Finally, constraint (5) fixes the number of credits per academic year.
Case study

In order to validate the proposed model, we applied it to the curriculum of the study programme “Professional Bachelor Applied Computer Science” mentioned above. The programme consists of 34 courses taught over three academic years. The total number of credits per academic year is 60. An academic year consists of two semesters, which we will refer to as the fall (September-February) and spring semester (February-July), respectively. The academic year starts the last week of September, so the first, third, and fifth semesters are fall semesters, the second, fourth, and sixth are spring semesters.

The model was implemented in CPLEX, a well-known commercial solver for Mixed Integer Linear Programming (MILP) problems developed by IBM. We took the prerequisites for each course that were specified in the ECTS sheets and encoded them in the model. It typically takes less than 30 seconds to find an optimal solution. The study programme proposed by the algorithm can be found in Figure 1 and Table III. The four first semesters contain courses each totaling 30 credits. Only the last two semesters, it was not possible to assign 30 credits to each. During the final semester, students do not have regular classes anymore, but instead do an internship of three months in a company. The course corresponding to internships (denoted STAG in the figure, short for “stages”, i.e. internships in Dutch) has 27 credits, and all other courses are hard prerequisites. Consequently, it must be assigned to a semester by itself, and the fifth semester will have 3 credits extra. In this proposed solution, 26 of the 34 courses were assigned to the same semesters as currently in reality. The changes are fairly reasonable. In fact, credits are spread out more evenly as in reality (See Tables II and III).

---

3 IP is a special case of MILP
Figure 1. Study programme proposed by the algorithm. Edges denote prerequisite relations between courses, as specified in the ECTS sheets. Edges with dashed lines denote courses that can be assigned to the same semester even though one is a prerequisite of the other (soft prerequisite). Columns denote the 6 semesters in the programme. Some edges are not shown for clarity. The course “STAG” (representing internships, “Stages” in Dutch) has all other courses as a (hard) prerequisite. Also, some superfluous prerequisites were removed from the graph by transitive reduction. For example, prerequisite TWI1 -> OPON is already expressed by prerequisites TWI1 -> TWI2 and TWI2 -> OPON.

Table II. Current course assignments. This table shows the courses as they are currently assigned in reality, including credits per course and totals per semester.

<table>
<thead>
<tr>
<th>Semester</th>
<th>$c_i$</th>
<th>Semester</th>
<th>$c_i$</th>
<th>Semester</th>
<th>$c_i$</th>
<th>Semester</th>
<th>$c_i$</th>
<th>Semester</th>
<th>$c_i$</th>
<th>Semester</th>
<th>$c_i$</th>
<th>Semester</th>
<th>$c_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARC</td>
<td>6</td>
<td>ALGO</td>
<td>5</td>
<td>A&amp;O1</td>
<td>6</td>
<td>BENW</td>
<td>5</td>
<td>A&amp;O2</td>
<td>4</td>
<td>STAG</td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMDB</td>
<td>5</td>
<td>CMIT</td>
<td>4</td>
<td>BEPR</td>
<td>5</td>
<td>DB</td>
<td>5</td>
<td>CVFE</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table III. Proposed course assignments.
This table shows the course assignments as proposed by the algorithm, including credits per course and totals per semester. Differences are emphasized.

<table>
<thead>
<tr>
<th>Semester 1</th>
<th>$C_i$</th>
<th>Semester 2</th>
<th>$C_i$</th>
<th>Semester 3</th>
<th>$C_i$</th>
<th>Semester 4</th>
<th>$C_i$</th>
<th>Semester 5</th>
<th>$C_i$</th>
<th>Semester 6</th>
<th>$C_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMDB</td>
<td>5</td>
<td>ALGO</td>
<td>5</td>
<td>A&amp;O1</td>
<td>6</td>
<td>BENW</td>
<td>5</td>
<td>A&amp;O2</td>
<td>4</td>
<td>STAG</td>
<td>27</td>
</tr>
<tr>
<td>FOOO</td>
<td>5</td>
<td>BMGT</td>
<td>4</td>
<td>BEPR</td>
<td>5</td>
<td>DB</td>
<td>5</td>
<td>CVFE</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOOP</td>
<td>6</td>
<td>CMIT</td>
<td>4</td>
<td>CARC</td>
<td>6</td>
<td>DWAP</td>
<td>5</td>
<td>DBDM</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTBE</td>
<td>5</td>
<td>OOPR</td>
<td>4</td>
<td>FEPR</td>
<td>5</td>
<td>KITP</td>
<td>3</td>
<td>DCNW</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TFE1</td>
<td>5</td>
<td>PRJ1</td>
<td>7</td>
<td>INTT</td>
<td>4</td>
<td>PRJ2</td>
<td>7</td>
<td>ICTM</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TWI1</td>
<td>4</td>
<td>TWI2</td>
<td>6</td>
<td>OPON</td>
<td>4</td>
<td>TFE2</td>
<td>5</td>
<td>OPTA</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Totals:</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>33</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Discussion

An important consideration for administrators is assignment of teaching staff to courses each academic year. Courses are added or removed, as dictated by industry needs. Organizational changes within the institution induce changes in the programme. The number of students enrolled in each course fluctuates each year, either because of new enrollments, drop-outs, or students retaking courses they failed. Some courses require a more intensive guidance from teaching staff than others, so more staff should be assigned per capita. Teaching staff tend to specialize in specific domains. Also, large differences in load between the fall and spring semester should obviously be avoided at all cost. The model could be extended to address these issues, but adding these factors to the equation requires a large amount of additional data. In order to know how many teachers must be assigned to each course, depends on expected number of enrolled students, and the “intensity” of the course (which can be expressed as a scalar value). Adding data and constraints to the model has its limits, though. At a certain point, refining the model has no more additional value. Finding a solution will take much longer, and it may be
unusable. For example, teaching staff tend to specialize in specific domains, or may have particular preferences with regard to course assignments. A mathematically calculated reassignment of courses could cause teachers to lose their favorite course(s). This is a sensitive matter that would result in tension among the staff members. These human factors can best be solved by negotiation, so we see the solution found by the algorithm as a starting point for discussion among staff, rather than being etched in stone.

References
Borodin, D. & Tokarev, A. “Mathematical Models for Educational Planning”, Quality Journal. 8, 8, p. 5-14. 10 p. (in Russian)
Masurel, E. & Nijkamp, P., 2009. "Bridging the gap between institutions of higher education and small and medium-size enterprises", Serie Research Memoranda 0037, VU University Amsterdam, Faculty of Economics, Business Administration and Econometrics.


University College Ghent (2008) Zelfevaluatiertapport van de opleiding professionele bachelor in de toegepaste informatica (Self-Assessment Report of the Professional Bachelor in Applied Computer Science), Department of Business Information & ICT and Department of Business Administration, University College Ghent, November 2008


List of Tables and Figures

Table I. Competences and indicators for the course Network and System Administration.

Table II. Current course assignments.

Table III. Proposed course assignments.

Figure 1. Study programme proposed by the algorithm.