*International Journal of Computer and Information Technology (ISSN: 2279 – 0764)*
*Volume 06 – Issue 06, November 2017*

# A Model for Study Plan Development with Uncertainty Representation

Hamdy A. Ibrahim
Faculty of computers and information
Menoufia University, Egypt
*Email: hamdy.ibrahim [AT] sait.ca*

*Abstract*— **Study plan development is a key task in academic advising. It is a challenging and time consuming as well it requires too much effort from academic advisors and students. Study plan is characterized by its dynamic nature because of possible change in students' preferences and university regulations. This paper introduced a method for generating a study plan. The proposed method includes two main phases. In the first phase, the settings of study plan development problem are defined and a formal integer programming model is developed. In the second phase, (1) The model is solved using branch-and-bound method to generate an optimal study plan which satisfy university regulations and students' preferences, (2) Searching for alternative study plans which have different structure and similar quality, and (3) Selecting, executing, and evaluating one of the generated study plans. The paper focuses on the first phase and phase 2 - step 1. The proposed method is distinguished from other approaches by representing uncertainty associated with the data (e.g. expected grades) and treating elective courses as if they belong to different academic areas. A real case study is presented to validate the proposed method.**

***Keywords-component; Academic advising; Study plan development; Decision support system; Branch-and-bound; Uncertainty management.***

## I. INTRODUCTION

Academic advising refers to a process in which an advisor assists and guides students in achieving personal, academic, and career goals. It focuses on assisting and guiding students during the process of courses selection and registration. In higher education, academic advising is a significant activity to improve students' educational experiences and also plays a major role in students' academic success and retention. In a survey of 988 (2 and 4 years) colleges, ineffective academic advising was identified as critical factor to student attrition in universities [1].

Study plan development (SPD) is an integral component of academic advising. It refers to a process in which courses required for graduation and not completed are assigned to number of semesters. The study plan represents a roadmap for students to complete degree requirements. Developing an effective study is time consuming and may take substantial effort from advisors and students. The developed study plan must comply with academic regulations and satisfy students' needs and preferences [2].

SPD is associated with many challenges: (1) Study plans are developed for a high number and variety of students with different abilities and skills. (2) SPD is usually running in dynamic environment because of continuous change in both educational institutions (e.g. regulations) and student situation (e.g. personal and/or financial). (3) Study plans have to be adapted to best fit student's evolving goals (4) In SPD, many constraints should not be violated. These constraints are arisen from academic regulations and students' preferences. These constraints vary from student to another depending on student's status and his/her academic progress. (5) Study plans are developed for different academic disciplines with a huge number of different courses which have different attributes (e.g. credit hours, prerequisite courses, fees, etc.) (6) In SPD, dependencies between courses should be taken into account.

Addressing these challenges to make academic advising particularly SPD effective and efficient requires a systematic and well defined method. The main goal of this research is automating academic advising to support advisors and students efficiently and provide them with the required knowledge to make appropriate decisions.

The overall goals of this research are (1) supporting students and academic advisors during the academic advising process and (2) saving the time and effort required to generate a study plan by automating some advising activities. In this paper, a method is proposed to help students and academic advisors when generating a study plan. The proposed method consists of two main phases. In the first phase, the settings of SPD problem are defined and a formal model is developed. In the model, SPD problem is represented as an integer programming problem. In the second phase, SPD optimization problem is solved and a study plan is generated. The contribution of the paper is defining a formal model for SPD problem. This model is distinguished from other models by the following features:

- The model is student-oriented because it takes into account student's preferences, interests, and budget
- The model represents elective courses as if they belong to different academic areas/subjects.

- The model represents uncertainty associated with the expected grades using students' confidence in these grades and displaying their previous grades in similar courses.

The rest of the paper is structured as follows: In section II, the related work and their deficiencies are presented. Section III discusses the proposed SPD method. Section IV presents the formal model of the SPD problem. The algorithm of identifying the first appropriate semester in which a student can take a course is discussed in section V. Section VI presents a case study to validate the proposed method. In section VII, the conclusion and future work are summarized.

## II. Literature Review

Reviewing the technical literature reveals that there are many computer-based academic advising systems which support students and advisors to complete general academic advising tasks. Its main role is facilitating information access to students and advisors and improve academic services. However, this should not replace the advisor role and face-to-face interactions between students and advisors [3]. It can be classified into two categories: advisor-oriented and advisor/student-oriented [4]. The first category only supports advisors and students rely on advisors to complete course registration. However, the second category is more interactive and supports both advisors and students. For example, Indiana University developed INSITE (INdiana Student Information Transaction Environment) designed to help advisors and students to review degree requirements and the student's progress [5]. INSITE generates an advising report which includes information about in-progress courses, grades, and possible future courses. In [6], a web-based expert system which assists fresh high-school graduates to select an academic major. [5] proposed a web-based academic advising tool which recommends a list of future courses based on the student information. It also helps students to complete some tasks without relying on academic advisors. In [7], a spreadsheet-based decision support system for academic Advising was proposed to automate repetitive academic advising tasks. In [8], a rule-based expert system was proposed to guide students to complete routine advising tasks. In [9], InVEStA (Interactive Virtual Expert System for Advising) was proposed to help students and advisors to create accurate and conflict-free schedules.

Moreover, there are few techniques and applications that address and discuss course planning problem. For example, [4] described a flexible recommendation application of course planning. This application defines and customizes recommendations over structured data. In [2], a Petri Net-based model was proposed. The model suggests courses to students taking into account balancing course load, frequency of the course offering, and shortening the path length to graduation. [10] presented a case study where data mining techniques were used to evaluate recommended and personalized study plans in terms of their effect on students' performance. [11] presented a web-based academic advising and planning tool which helps students to select and register courses. The tool is able to generate a study plan for the upcoming semesters. In [12], a technique based on the analytic hierarchy process (AHP) and multichoice goal programming (MCGP) model was proposed to generate a course plan. The research in [13] used two project management tools which help students to graduate sooner. [13] focused on the first tool which presents course sequences using a visualization map tailored for each student. In [14] the problem of elective courses planning was modeled using integer programming and three solutions including Branch-and-Price framework using partial Dantzig–Wolfe decomposition was proposed. In [15], a decision support system for course planning was proposed. The system used a formal model for the course planning problem to generate a set of optimal or near optimal alternative study plans of similar quality and different structures.

No doubt the above research attempts contributed in addressing a set of challenges associated with academic advising. However, the following deficiencies and limitations have been identified in these techniques:

1. Approaches that formally model and tackle the SPD problem do not have supporting tools. On the other hand, approaches supported by tools do not formally model the SPD problem.
2. Most approaches are advisor-oriented not student-oriented and its main focus is respecting academic regulations. These approaches do not significantly take into account students' preferences and satisfaction.
3. Failing to consider uncertainty possibly associated with the SPD problem (e.g. expected grade in a course).
4. Failing to consider different types of dependency between courses.
5. Most if not all approaches treat elective courses as one set and ignore the fact that elective courses may belong to different academic areas.

## III. Proposed SPD Method

Figure 1 depicts the proposed SPD method which consists of two phases.

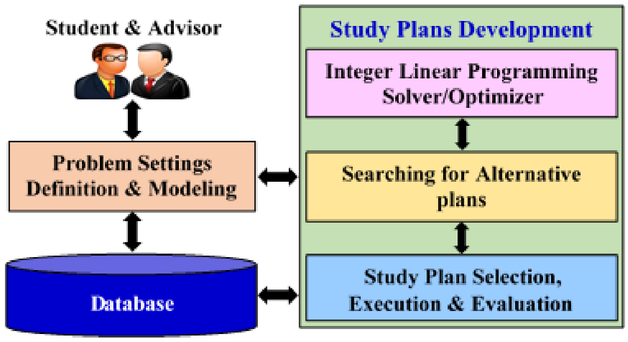


Figure 1. Proposed SPD Method

### A. ***Problem Settings Definition & Modeling***

In this phase, SPD problem is formally defined and modeled as integer programming problem because all of decision variables are integer (see section IV). This includes defining an objective function and a set of constraints. The objective function aims at maximizing students' satisfaction about the developed study plan and achieving a high GPA. The constraints represent academic regulations and students' preferences. Student and academic advisor specify values of different decision variables. These values represent student's preferences and vary from student to student. Student's preferences include (1) Which elective courses s/he prefers to take, (2) In which semesters s/he prefers to take different courses, (3) Expected grades for unfinished courses, (4) his/her confidence level in the expected grades, (5) his/her budget for each semester, etc.

Data can be retrieved from database and used as guiding values during the problem settings definition. This data includes student's performance in completed courses which belong to the same academic area. Data provided by academic advisor includes (1) Credit hours and fees per course, (2) Course prerequisites, and (3) Semesters in which courses are offered.

### B. *Study Plan Development*

In this phase, the integer programming problem defined and modeled in the previous phase, is solved using branch-and-bound methods. Practically, branch-and-bound methods always outperform implicit enumeration methods. The paper used a software tool called LINGO [16]. In particular, LINGO branch-and-bound solver has been used to find the optimal solution for SPD optimization problem. LINGO is selected because (1) It provides a full featured environment for efficiently and easily developing and solving integer programming models, (2) It includes a set of fast and built-in solvers such as direct, linear, nonlinear, and branch-and-bound solvers. Built-in solvers mean there is a direct link between the modeling and solver components. Therefore, the compatibility problems between the two components are minimized, (3) Easy model expression: Its scripting language allows modeling an integer programming problem using standard mathematical notation, and (4) It was used by other researchers [15, 17, 18].

For the defined problem, LINGO generates only one study plan which achieves the objective function (i.e., finding its optimal value) and meets the problem constraints. Solving SPD problem using LINGO includes two steps:

(1) Creating a LINGO SPD optimization model in the LINGO model window. This step includes expressing three components:
   a. The objective function that describes what the model should optimize.
   b. A set of decision variables representing the values that can be changed to generate the optimal value of the objective function.
   c. Constraints which define the limits on the values of the decision variables.

Section IV discusses the above three components of SPD optimization model.

(2) Using branch-and-bound solver to solve the LINGO SPD model that has been created in step 1.

## IV. The Model For SPD

As discussed in section III, SPD problem is modeled as integer programming problem by defining an objective function and a set of constraints. Two key objects (courses and students) are considered in the SPD model. This section discusses the decision variables, Constraints related to these objects, and objective function.

### A. *Courses*

In most of higher educational institutions, degree requirements include completion of all compulsory courses $C_C$ and a number of elective courses $C_E$. Elective courses can belong to different study areas. In some universities, students may need to complete minimum credits from all elective courses regardless the study areas. However, other universities may require students to complete minimum credits from elective courses belonging to different study areas (e.g. at least 2 courses from study areas 1 and 2). The courses component is represented as follows:

- Assume C refers to a set of Z courses. The set C is divided into two sets: $C_C$ with n compulsory courses and $C_E$ with (z-n) elective courses.

$$C_C = \{c_1, c_2, ..., c_n\} \quad (1)$$

$$C_E = \{c_{n+1}, c_{n+2}, ..., c_z\} \quad (2)$$

Where:

$$\mathrm{C} = \mathrm{C_C} \cup C_E$$

$$\mathrm{C_C} \cap C_E = \phi$$

- The study plan represents an organized schedule where uncompleted courses are assigned to a specific number of semesters (T). In the study plan, a course $c_i$ is assigned to only one semester t. Course assignment is represented using a set of decision variables $A_i$ defined as follows:

$$A_i = \{a_{i,t} : i = 1..z, t = 1..T\} \quad (3)$$

The decision variable $a_{i,t}$ is set to 1 if the course $c_i$ is assigned to semester t. It is defined as follows:

$$\forall c_i : a_{i,t} = \begin{Bmatrix} 1 & \text{if } \mathrm{c_i} \rightarrow \mathrm{t} \\ 0 & \text{otherwise} \end{Bmatrix} \quad (4)$$

For example, if the course $c_1$ is assigned to the 3rd semester, $a_{1,3}$ will be set to 1 and ($a_{1,1}$, $a_{1,2}$, $a_{1,4}$...etc.) will be set to 0. Accordingly, $A_1 = \{0,0,1,0,\ldots\}$.

- Students have to complete a minimum credit hours (H) to get an academic degree. For any semester except first semester, a student already finished a number of courses and did not finish other courses. Accordingly, courses can also be divided into finished (*fin*) and unfinished (*unfin*) sets. Assuming $r_i$ is credit hours for course $c_i$, the following constraint must be met to get the academic degree.

$$\sum_{c_i \in fin} c_i \cdot r_i + \sum_{c_j \in unfin, t=1..T} c_j \cdot r_j \cdot a_{j,t} \geq H \quad (5)$$

The study plan must include all compulsory courses and some elective courses excluding all finished courses. A course $c_i$ is also assigned to only one semester in the study plan. The sum of the decision variables $a_{i,t}$ equals 0 if the elective course is not included in the study plan. It equals 1 if the elective course is included in the study plan.

Therefore, the following constraint must be met.

$$\forall c_i : \sum_{t=1}^{T} a_{i,t} \begin{cases} =1 & \text{if } c_i \in C_C \cap unfin \\ \leq 1 & \text{if } c_i \in C_E \\ =0 & \text{if } c_i \in fin \end{cases} \quad (6)$$

- The academic regulations may specify some elective courses as must-take courses. The must-take elective courses are treated as compulsory courses. Suppose that $C_{E\_musttake}$ refers to the set of must-take elective courses. The generated study plan will satisfy the following constraint:

$$\forall c_i \in C_c \cup C_{E\_musttake} : \sum_{t=1}^{T} a_{i,t} = 1 \quad (7)$$

- An elective course ($c_i \in C_E$) may belong to one of K-study areas $\{S_1, S_2, \ldots, S_K\}$. A decision variable $S_{i,j}$ is used to indicate that a course ci belongs to study area $S_j$ and defined as:

$$S_{i,j} = \begin{Bmatrix} 1 & \text{if } c_i \in S_j \\ 0 & \text{if } c_i \notin S_j \end{Bmatrix} \quad (8)$$

- The educational institution may require to complete minimum ($min_j$) and maxmium ($max_j$) credit hours from a study area $S_j$. This is modeled as follows:

$$\forall S_j : \min_j \leq \sum_{i=1}^{z} S_{i,j} \cdot r_i \cdot \sum_{t=1}^{T} a_{i,t} \leq \max_j \quad (9)$$

- The total credits of all elective courses from different study areas must be also between minimum ($min_h$) and maximum ($max_h$) credit hours. This is modeled as follows:

$$\min_h \leq \sum_{j=1}^{k} \sum_{i=1}^{z} S_{i,j} \cdot r_i \cdot \sum_{t=1}^{T} a_{i,t} \leq \max_h \quad (10)$$

- A limit (i.e. minimum and maximum) may be specified for total credit hours that a student can register in a semester. This limit vary depending on (1) whether the student is a full-time or part-time, (2) high GPA versus low GPA, and (3) fall and winter versus spring and summer semesters. For a semester t, the limit is defined by $min_t$ and $max_t$. This is represented using the following constraint:

$$\forall t : \min_t \leq \sum_{i=1}^{z} r_i \cdot a_{i.t} \leq \max_t \quad (11)$$

- ***Course dependency*** refers to a relation between two or more courses. It often reflects the academic enrollment requirements (e.g., prerequisite courses) that must be satisfied prior registering in a course. Accordingly, courses can be dependent or independent. Dependent courses represent the fact that (1) prerequisite courses for a course must be finished before registering this course, and (2) advanced courses requires completion of introductory courses before registering these advanced courses. In a study plan, dependent courses cannot be assigned to the same semester. However, independent courses are courses which are not related to each other. Independent courses can be taken concurrently. Independent courses can be assigned to the same semester in the study plan. Table I summarizes the two types of dependency used in this model.

TABLE I. TYPES OF COURSE DEPENDENCY

| Dependency | Description | Example |
|---|---|---|
| **Concurrency** | Both courses must be assigned to the same semester. | $c_k$ and $c_l$ are assigned to the $1^{st}$ semester. |
| **Precedence** | Both courses must be assigned to different semesters. | • $c_k$ is assigned to semester $t_1$.<br>• $c_l$ is assigned to semester $t_2$.<br>• $t_1$ precedes/before $t_2$ |

Suppose that there are 2 courses $c_k$ and $c_l$ which are assigned semesters $t_1$ and $t_2$ respectively. The dependency between $c_k$ and $c_l$ can be represented by calculating the difference ($d_{k,l}$) between $t_1$ and $t_2$. If the $d_{k,l}$ = 0, the dependency is concurrency ($t_1 = t_2$). However, if $d_{k,l} \geq 0$, the dependency is precedence. When $d_{k,l}$ = 1, the two courses are assigned to consecutive semesters (e.g., $t_1 = 2$ and $t_2 = 3$).

$$d_{k,l} = t_1 - t_2 \quad (12)$$

$$d_{k,l} = \begin{Bmatrix} \geq 1 & a_{k,t_1} = 1, a_{l,t_2} = 1, t_1 \neq t_2 \\ 0 & a_{k,t_1} = a_{l,t_2} = 1, t_1 = t_2 \end{Bmatrix} \quad (13)$$

- The educational institution may only offer some courses in specific semesters. Suppose $c_i$ is only offered in semester t. Therefore, the decision variable $a_{i,t}$ will be pre-set to 1 and to 0 for other semesters.

$$\begin{aligned} &\forall c_i : a_{i,t} = 1 \\ &\forall c_i : a_{i,s} = 0, s \in T, s \neq t \end{aligned} \quad (14)$$

## *B.* ***Students***

When students use the proposed method, they are required to determine the following:

- ***Maximum number of courses per a semester:*** A student can determine the maximum number of courses ($\sigma_t$) s/he can take per semester (t). This is modeled using the following constraint:

$$\forall t : \sum_{i=1}^{z} a_{i,t} \leq \sigma_t \quad (15)$$

In circumstances where a student's GPA is lower than a threshold value, the academic regulations may also restrict the maximum number of courses s/he can take.

- ***Student's preference for courses and semesters:***

Students must take all compulsory courses. Therefore, student's preference determines semesters in which s/he prefers to take these courses. However, they take some elective courses. Therefore, student's preference determines which elective courses s/he prefers to take and in which semesters.

For an elective course $c_i$, the student assigns an interest value ($I_i$) which represents the importance of the elective course to the student and amount of his/her interest to take it. A 1-5 scale (see Table II) is used to represent the interest value. 5 means an elective course is very important and the student has a very high interest in the course. However, 1 means not at all important. As a result, elective courses are prioritized based on interest values.

$$\forall c_i \in C_E : I_i \in \{1,2,3,4,5\} \tag{16}$$

TABLE II. SCALE FOR INTEREST VALUE

| | |
|---|---|
| 1 | Not at all important |
| 2 | Slightly important |
| 3 | Important |
| 4 | Fairly important |
| 5 | Very important |

An elective course with interest value 5 will be included in the study plan instead of an elective course with interest value 1.

Suppose that a course $c_i$ is offered in fall (*f*), winter (*w*), and spring (*s*) semesters and its prerequisite courses were finished. If the student prefers to take $c_i$ in winter only then $a_{i,w}$ will be pre-set to 1. Suppose that a course $c_i$ is offered in many semesters. The student can also assign a preference value ($\rho_{i,t}$) to each semester t. $\rho_{i,t}$ represents how preferable to take a course $c_i$ in a semester t. A 1-5 scale is used for the preference value $\rho_{i,t}$. 5 means it is highly preferred to take $c_i$ in semester t. However, 1 means it is not at all preferred to take $c_i$ in semester t.

$$\forall c_i : \rho_{i,t} \in \{1,2,3,4,5\} \tag{17}$$

A course will be assigned to the most preferred semester (i.e., the semester with the highest preference value) provided the prerequisite courses were finished before this semester. A course may be assigned to a semester with the less preference value if this assignment has a better influence on GPA, student's satisfaction, and constraints' satisfaction.

As soon as preference values are assigned to semesters, the algorithm described in section V will be applied to calculate the first appropriate semester to which a course could be assigned and compare it to the most preferred semester. Accordingly, the course will be assigned to the most preferred semester or later. Suppose that a student prefers to take the course $c_i$ in the $4^{th}$ semester (i.e., $4^{th}$ semester has the highest preference value). By applying the algorithm, it is found that the course $c_i$ cannot be taken before the $5^{th}$ semester. Accordingly, the course will be assigned to the $5^{th}$ semester or later.

- ***Expected grade for each course***

Assuming $g_i$ is the expected grade for a course $c_i$. Considering the developed study plan, the total grade points ($G_t$) for a semester t is calculated as follows:

$$G_t = \sum_{i=1}^{z} g_i \cdot r_i \cdot a_{i,t} \tag{18}$$

The expected grades entered by students could be uncertain. This uncertainty and associated consequences can be mitigated by retrieving grades obtained by students in the past courses belonging to the same academic area. The retrieved grades reflect student's performance in similar courses and can be used to as a basis for identifying the expected grades. For example, if a student will enter his expected grade for a database course, grades for the past database courses will be retrieved and displayed to the student.

Another way to mitigate the uncertainty and its consequences is allowing students to enter their confidence level ($\omega$) in the expected grades. Confidence value ranges from 0 to 1 and shown in Table III.

TABLE III. CONFIDENCE VALUE DESCRIPTION

| Confidence value | Description |
|---|---|
| 0 | Student is totally uncertain |
| 1 | student is totally certain |
| $0 \le \omega \le 1$ | Different levels of uncertainty |

The equation used to calculate $G_t$ is modified to include the confidence level.

$$G_t = \sum_{i=1}^{z} g_i \cdot r_i \cdot \omega_i \cdot a_{i,t} \tag{19}$$

Where $\omega_i$ : confidence level in the expected grade for a course $c_i$.

The grade point average (GPA) reflects how students perform in their academic study. Students must maintain a satisfactory GPA (e.g. 2.0 or higher) to avoid academic probation, program withdraw, or dismissal. A key characteristic of a good study plan is helping students to achieve a higher GPA. Achieving a higher GPA requires that the study plan to include courses in which students expect a higher grade and they are more interested in it. The GPA ($\delta_t$) for a semester t is calculated as follows:

$$\forall t : \delta_t = \frac{G_t}{\sum_{i=1}^{z} r_i \cdot a_{i,t}} \tag{20}$$

Where the denominator represents the total amount of credit hours.

Students with unsatisfactory performance are at risk of dismissal. Their GPA for a semester t must be higher than a threshold value ($\lambda_t$) to avoid dismissal. This is modeled using the following constraint:

$$\forall t : \delta_t \ge \lambda_t \tag{21}$$

The SPD model is considered as a nonlinear model because there is a decision variable in the denominator of the GPA equation.

- ***Budget***

  Students pay two types of fees: tuition (e.g. $\nu_i$: tuition fees for a course $c_i$) and general fees ($\tau_t$). Tuition fees include course registration. General fees include health plan, U-pass, recreation, students' union, student services, etc. The model allows students to enter their budget for each semester. For semester t, total fees should not exceed student's budget ($\beta_t$). This is modeled as follows:

$$\forall t: \sum_{i=1}^{z} \nu_i \cdot a_{i,t} + \tau_t \leq \beta_t \tag{22}$$

- ***Leave of absence:***

  In some circumstances, a student may request a leave of absence for one or more semesters. Suppose that a student will take a leave of absence in a future semester t. Therefore, decision variables related to the student such as budget and maximum number of courses per semester are set to 0. Furthermore, no courses will be assigned to the semester t. This is modeled using the following constraint:

$$\forall t: \sum_{i=1}^{z} a_{i,t} = 0 \tag{23}$$

  When the student takes a leave of absence for many semesters, the above constraint will be applied to these semesters.

### *C.* ***Objective Function***

A key step in integer programming problem is defining an objective function which represents a set of goals to be optimized while meeting a set of constraints. Two goals were identified for study plan development:

(1) Maximizing students' satisfaction which depends on their interest to take specific elective courses (i.e., $I_i$) and their preferences about in which semesters they will take courses (i.e., $\rho_{i,t}$). Additionally, satisfying students related constraints will increase their satisfaction.

(2) Achieving a high GPA which depends on expected grades ($g_i$).

Accordingly, the objective function F(x) is formulated as follows to include the above goals.

$$F(x) = \sum_{i=1}^{z} \left( (g_i)^{\theta} \cdot I_i \cdot \sum_{t} \rho_{i,t} \cdot a_{i,t} \right) \tag{24}$$

Where Ɵ refers to a parameter which is specified by students to determine the relative influence of the expected grade $g_i$ on the value of F(x). Ɵ takes values from 0.5 to 2.0. The value 0.5 means that $g_i$ has a very low impact and 2.0 means its impact is very high. The value assigned to Ɵ varies from a student to another. It is recommended to assign a high value to Ɵ. Therefore, elective courses with higher expected grades will be included in the study plan. Consulting students and academic advisors suggests setting Ɵ to 2.

## V. . Identification of The First Appropriate Semester

The first appropriate semester (FAS) refers to the first possible semester to which a course is assigned. The course cannot be assigned to a semester before FAS because the prerequisite courses will not be finished before it. The algorithm shown in Figure 2 identifies the appropriate semester to which a course can be assigned. It is a modified version of the one used in [15]. For each course $c_i$, the algorithm is applied in two main steps:

(1) Taking prerequisite courses into account, determining the first appropriate semester ($FAS_i$) to which $c_i$ could be assigned. At this semester, all prerequisite courses will be finished and $c_i$ can be taken in $FAS_i$ or after it.

(2) Comparing $FAS_i$ to the most preferred semester ($MPS_i$). If $FAS_i \leq MPS_i$ then $c_i$ will be assigned to $MPS_i$. Otherwise, the student will be notified to change his/her preference and accept the assignment of $c_i$ to $FAS_i$ or after it.

```
START
  FOR c_i ∈ unfin
      Read preference values for all semesters
      Save preference values into prefArray
      Save semesters into semArray
      Set max = Max(prefArray)  // Find the the highest preference value
      Set j = index of the highest preference value
      Set MPS_i = semArray[j]
      calculatePathLength(c_i,0)
      Set FAS_i to max(pathLengthArray)
      IF FAS_i <= MPS_i
         Assign c_i to MPS_i
      Else
        Assign c_i to FAS_i
      END IF
   END FOR
END
void calculatePathLength(c, count)
{
    FOR c_p ∈ {prerequisite_courses(c)}
        Set count_p = count
        IF (c_p has prerequisite courses & c_p ∈ unfin)
           count_p++
           calculatePathLength(c_p, count_p)
        ELSE add count_p to pathLengthArray
        END IF
    END FOR
}
```

Figure 2. FAS Estimation Algorithm

The algorithm retrieves student's preference values assigned to semesters. Preference values and corresponding semesters are saved into prefArray and semArray respectively. Afterwards, the algorithm finds the semester with the highest preference value and save it into $MPS_i$. The function calculatePathLength is then called to check all prerequisite courses ($c_p$) for $c_i$ and determine the FAS for each prerequisite course by recursively calling the function. The recursive call continues till reaching a finished prerequisite course or a course which does not have prerequisites. The loop in the function traverses all prerequisite courses and a counter value is estimated for each prerequisite course path. The counter variable represents the number of unfinished prerequisites along a specific path. When the recursive call stops, the counter is saved into pathLengthArray and another loop iteration starts till all prerequisites are checked.

Afterwards, the maximum counter value which represents the longest path of prerequisites is identified and saved in $FAS_i$. Finally, $FAS_i$ is compared to $MPS_i$ to determine to which semester $c_i$ will be assigned. Figure 3 shows an example for $C_7$ course prerequisites. A student must finish $C_4$, $C_5$, and $C_6$ courses before taking $C_7$.

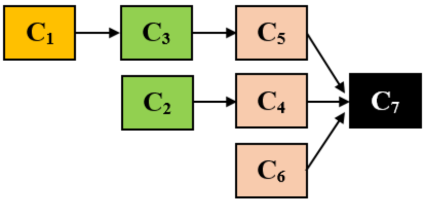


Figure 3. Course Prerequisites

$C_5$ has 2 prerequisite ($C_1$ and $C_3$). $C_4$ has one prerequisite ($C_2$). However, $C_6$ does not have prerequisites. Suppose that all courses in Fig. 3 are not finished. Identifying FAS for C7 requires checking three prerequisites paths ($C_1$ to $C_5$, $C_2$ to $C_4$, and $C_6$). The length of $C_1$ to $C_5$ Path is 3 because 3 semesters are required to finish its prerequisites. The length of $C_2$ to $C_4$ Path is 2 because 2 semesters are required to finish its prerequisites. The length of $C_6$ Path is 1 because only 1 semester is required to finish $C_6$. Accordingly, the longest path is the first path and $C_7$ can be only registered after 3 semesters. Suppose that $C_1$ will be registered in the $1^{st}$ semester. Therefore, $C_7$ can be registered in the $4^{th}$ semester (i.e. FAS = 4) or after it. Suppose that the student assigned the highest preference value to the $3^{rd}$ semester (i.e. MPS = 3). According to the algorithm shown in Figure 2, $C_7$ will be registered in the $4^{th}$ semester or later.

# VI. CASE STUDY

## A. *Overview*

This section validates the proposed method using real scenarios from the faculty of engineering where 45 courses are offered to students. This includes 32 ($c_1$ to $c_{32}$) compulsory courses and 13 ($c_{33}$ to $c_{45}$) elective courses. Graduation requires that a student must complete all 32 compulsory courses and 12 credit hours from the 13 elective courses. A leave-of-absence may be granted to a student for a period of up to 52 weeks upon his/her request. The leave-of-absence is requested based on a semester by semester basis. During the duration of leave-of-absence, there is no tuition fees and the student cannot register courses.

A student must maintain a good standing by getting a good GPA to avoid dismissal. Otherwise, an academic warning is issued when the student earns a GPA of 1.00–1.99 for any semester. The student is dismissed from the faculty if s/he receives an academic warning for three consecutive semesters.

A student must register at least 12 credits per term to maintain full-time status. A student with a satisfactory standing can register between 8 and 19 credits per semester. While, a student who received two or more academic warnings may register maximum of 14 credits per semester. Furthermore, a student whose GPA is at least 3.5 can register 24 credits.

## B. *Validation Scenarios*

Two scenarios of students are used in the validation. The first scenario is for a student whose GPA>2.0. However, the second is for a student at dismissal risk because of his/her low performance. In both scenarios, input data is provided by university administration and students. The data provided by university (see Tables IV and V) is common between all students. For example, prerequisites and credit hours per course are same and do not change from a student to another student. However, data provided by students may change from a student to another student. Tables IV-VII and Figures 4-5 represent a sample of this data. Students should not violate university regulations during their data entry. For example, a student cannot choose spring semester to take a course although it is not offered in spring. Another example, a student cannot register less than 8 credit hours per semester.

TABLE IV. EXAMPLES FOR INPUT DATA

| University | Student |
|---|---|
| • List of compulsory and elective courses | • Interest values assigned to elective courses |
| • Course prerequisites | • For each course, preference values assigned to semesters |
| • Credit hours per course | • Expected grades for unfinished courses |
| • Tuition fees per each course | • Confidence level in the expected grades |
| • Semesters in which a course is offered | • Maximum number of courses per semester |
| • Total credit hours required for elective courses | • Total credit hours per semester |
| • Threshold GPA value for warning | • Budget per semester |
| • Minimum and maximum credit hours per semester | • Number of semesters towards graduation |
| | • First and last semesters per study plan |

TABLE V. DATA ENTERED BY UNIVERSITY

| $C_i$ | Type | Credit | Fees | Offered in | | Prerequisites |
|---|---|---|---|---|---|---|
| | | | | Fall | Winter | |
| $C_5$ | Compulsory | 3 | 240 | × | ✓ | $C_4$ |
| $C_9$ | | 5 | 364 | ✓ | ✓ | - |
| $C_{18}$ | | 5 | 364 | ✓ | ✓ | $C_7$,$C_{13}$ |
| $C_{25}$ | | 6 | 372 | ✓ | ✓ | $C_{19}$ |
| $C_{29}$ | | 3 | 240 | × | ✓ | $C_{10}$ |
| $C_{30}$ | | 3 | 240 | ✓ | ✓ | $C_{27}$ |
| $C_{39}$ | Elective | 3 | 240 | ✓ | × | $C_5$ |
| $C_{40}$ | | 3 | 240 | × | ✓ | $C_{18}$ |
| $C_{44}$ | | 3 | 240 | ✓ | ✓ | $C_{30}$ |
| $C_{45}$ | | 3 | 240 | ✓ | ✓ | $C_{29}$ |

In the $1^{st}$ scenario (see Table VI), the student finished 8 compulsory courses. He wants to develop a study plan for 6 ($T_1$-$T_6$) semesters without a leave-of-absence. At the end of $T_6$, s/he will finish 24 compulsory courses and 4 elective courses.

Therefore, s/he will complete all degree requirements. S/He is very interested to take the elective courses $c_{39}$, $c_{40}$, $c_{44}$, and $c_{45}$.

TABLE VI. SAMPLE DATA FOR STUDENT WITH GPA ≥ 2.0

| $C_i$ | Expected Grade | Confidence Value | Interest Value | Most Preferred Semester | | |
|---|---|---|---|---|---|---|
| $c_5$ | B+ | 0.85 | - | $T_1$ | | |
| $c_9$ | C+ | 0.67 | - | $T_4$ | | |
| $c_{18}$ | B- | 0.7 | - | $T_5$ | | |
| $c_{25}$ | A | 1 | - | $T_3$ | | |
| $c_{29}$ | A | 0.95 | - | $T_1$ | | |
| $c_{30}$ | B+ | 0.5 | - | $T_4$ | | |
| $c_{39}$ | A- | 1 | 4 | $T_2$ | | |
| $c_{40}$ | B- | 0.88 | 5 | $T_5$ | | |
| $c_{44}$ | B | 0.9 | 5 | $T_5$ | | |
| $c_{45}$ | A | 1 | 3 | $T_4$ | | |
| Study plan period | 6 semesters | | | | | |
| | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ |
| Total credits | 21 | 19 | 21 | 21 | 21 | 24 |
| No. of Courses | 5 | 4 | 5 | 5 | 5 | 6 |
| Threshold $\lambda_t$ | 2.7 | 2.7 | 2.7 | 2.7 | 2.7 | 2.7 |
| Budget | 1600 | 1600 | 1600 | 1600 | 1600 | 1600 |
| Leave-of-absence | None | | | | | |
| No. of finished compulsory courses | 8 | | | | | |

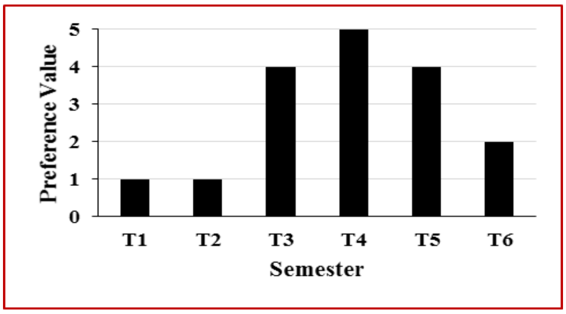


Figure 4. C9: Semesters' Preference Values for Student with GPA ≥ 2.0

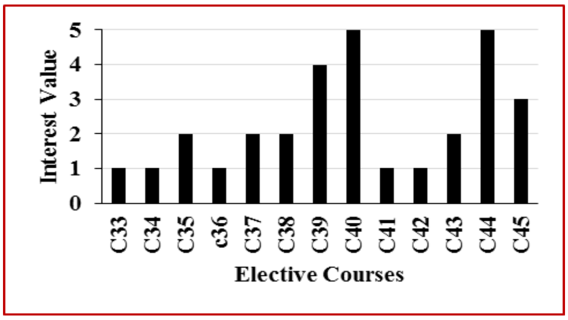


Figure 5. Elective Courses' Interest Values for Student with GPA ≥ 2.0

In the 2nd scenario (see Table VII), the student finished 10 compulsory courses. He received two academic warnings since his/her GPA is less than 2.0. S/He will be dismissed if he receives a 3rd warning. S/He also did not finish 12 credits from the 13 elective courses. S/He is very interested to take the elective courses $c_{33}$ and $c_{37}$. The prerequisites for $c_{33}$ and $c_{37}$ are $c_{11}$ and $c_9$ respectively. S/He needs to generate a study plan for 5 semesters ($T_1$, $T_2$, $T_4$, and $T_5$) and takes a leave-of-absence in semester $T_3$.

TABLE VII. SAMPLE DATA FOR STUDENT WITH GPA < 2.0

| $C_i$ | Expected Grade | Confidence Value | Interest Value | Most Preferred Semester | |
|---|---|---|---|---|---|
| $c_5$ | C | 0.75 | - | $T_2$ | |
| $c_9$ | A | 0.95 | - | $T_1$ | |
| $c_{18}$ | C+ | 0.7 | - | $T_2$ | |
| $c_{25}$ | B | 1 | - | $T_5$ | |
| $c_{30}$ | D | 0.65 | - | $T_1$ | |
| $c_{33}$ | B | 0.85 | 5 | $T_2$ | |
| $c_{37}$ | C+ | 0.6 | 5 | $T_4$ | |
| Study plan period | 5 semesters | | | | |
| | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
| Total credits | 20 | 19 | 0 | 16 | 20 |
| No. of Courses | 4 | 4 | 0 | 3 | 3 |
| Threshold $\lambda_t$ | 2.0 | 2.0 | 0 | 2.0 | 2.0 |
| Budget | 1400 | 1000 | 0 | 1000 | 1200 |
| Leave-of-absence | $T_3$ | | | | |
| No. of finished compulsory courses | 10 | | | | |

## C. ***Results Discussion***

Tables VIII-IX show the generated study plans for the two scenarios respectively. In these tables, the suffix (s), associated with the course, represents its credit hours. As shown in these tables: (1) The generated study plans satisfied most of the constraints such as student's budget and maximum credit hours that can be registered by students, (2) The expected GPA exceeds the threshold value ($\lambda_t$), (3) For the student with GPA < 2.0, the expected GPA for all semesters is increased to above 2.0, (4) Courses have not been assigned to $T_3$ as the student requested a leave-of-absence in this semester, (5) Elective courses with the highest interest values and highest grades have been included in the study plans, (6) Courses have not been assigned to the most preferred semesters because its prerequisites were not completed. For example, the student requested to take $c_{30}$ in $T_1$ (see Table VII). Table XI shows that $c_{30}$ has assigned to $T_2$ (i.e., $FAS_{30}$) because the student must take $c_{27}$ first, and (7) The study plans satisfy the maximum number of courses that a student wish to register in a semester

TABLE VIII. STUDY PLAN FOR STUDENT WITH GPA ≥ 2.0

| | | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ |
|---|---|---|---|---|---|---|---|
| **Finished Courses** | | $c_{4(3)}$ | $c_{5(3)}$ | $c_{39(3)}$ | $c_{9(5)}$ | $c_{8(3)}$ | $c_{40(3)}$ |
| $c_1$ | $c_{14}$ | $c_{7(4)}$ | $c_{19(5)}$ | $c_{25(6)}$ | $c_{30(3)}$ | $c_{18(5)}$ | $c_{16(6)}$ |
| $c_2$ | $c_{21}$ | $c_{10(3)}$ | $c_{29(3)}$ | $c_{27(5)}$ | $c_{45(3)}$ | $c_{44(3)}$ | $c_{24(5)}$ |
| $c_3$ | $c_{23}$ | $c_{22(5)}$ | $c_{11(6)}$ | $c_{17(5)}$ | $c_{12(3)}$ | $c_{32(4)}$ | $c_{26(3)}$ |
| $c_6$ | $c_{13}$ | $c_{15(5)}$ | | | $c_{20(5)}$ | $c_{28(5)}$ | $c_{31(4)}$ |
| No. of Courses | | 5 | 4 | 4 | 5 | 5 | 5 |
| Total Credits | | 20 | 17 | 19 | 19 | 20 | 21 |
| Expected GPA | | 3.0 | 2.95 | 3.10 | 2.74 | 3.0 | 2.88 |
| Total Cost | | 1510 | 1216 | 1340 | 1448 | 1510 | 1518 |

## D. ***Key strengths***

- Generating the study plan using the proposed method will save the time and effort of students and academic advisors to manually generate it.
- The defined model is a realistic and comprehensive model because it represents a set of realistic university regulations. Furthermore, the model was not developed based on unrealistic assumptions.

TABLE IX. STUDY PLAN FOR STUDENT WITH GPA < 2.0

| | | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
|---|---|---|---|---|---|---|
| **Finished Courses** | | $c_{4(3)}$ | $c_{5(3)}$ | - | $c_{18(5)}$ | $c_{11(6)}$ |
| $c_1$ | $c_{12}$ | $c_{9(5)}$ | $c_{30(3)}$ | - | $c_{19(5)}$ | $c_{13(4)}$ |
| $c_2$ | $c_{14}$,$c_{17}$ | $c_{7(4)}$ | $c_{10(3)}$ | - | $c_{37(3)}$ | $c_{25(6)}$ |
| $c_3$ | $c_{21}$ | $c_{27(5)}$ | $c_{33(3)}$ | - | | |
| $c_6$,$c_8$ | $c_{23}$ | | | - | | |
| No. of Courses | | 4 | 4 | 0 | 3 | 3 |
| Total Credits | | 17 | 12 | 0 | 13 | 16 |
| Expected GPA | | 2.28 | 2.53 | - | 2.15 | 2.44 |
| Total Cost | | 1270 | 960 | 0 | 968 | 1046 |

- The model can be adapted (e.g. changing, adding, or removing constraints) to represent different regulations and preferences. The adaptability feature helps in tackling the dynamic nature of study-plan development problem. The dynamic represents possible changes in university or student constraints.
- The proposed method is a student-oriented method because it considered students' preferences, interests, and budget. The method allows students to accept or reject the generated study plan. In case of rejecting the plan, it allows to modify the constraints to generate another plan.
- The model does not treat elective courses as one academic set and consider it as if they belong to different academic areas.
- The uncertainty is represented by considering students' confidence in the expected grades and displaying their previous grades in similar courses.

### *E. __Weaknesses__*

- The proposed method is used to generate only one study plan. Its flexibility can be improved by generating several study plans plan. These plans satisfy the constraints, achieve the objective function, and have different structures. The method should be able to find not only the optimal study plan but also the near-optimal study plans. Then students with the help of academic advisors analyze these plans and select the plan that best fits with their needs. Ensuring the quality of the generated plans may require adding a condition specifying that the value of the objective function must exceed a threshold value (e.g., 97%) of the optimal objective function found by solving the optimization model. In this paper, the method modules that are responsible for (1) searching for alternative study plans and (2) study plan selection, execution, and evaluation were not implemented.
- The proposed method is not supported by a software tool. Therefore, it is important to develop a software tool which supports students and academic advisors during the method application. This tool will improve the method usability.

## VII. CONCLUSION AND FUTURE WORK

This paper proposed a method which helps students and academic advisors to generate study plans. It aims at automating some of the academic advising activities to save the time and effort of students and academic advisors without replacing face-to-face interaction between them. The proposed method represents the SPD problem as optimization problem. The core of the method is a formal integer programming model which represents the SPD problem. The model includes a set of constraints and decision variables which represent university regulations as well student preferences and interests. The proposed method is able to suggest a study plan that satisfy student's needs and university regulations. It also addresses a number of key limitations of previous research work. It is characterized as a student-oriented method because the student participate in the overall process and it considers his/her preferences and interests in terms of what courses would he like to take and when. Two real scenarios was used to illustrate and validate the method. These scenarios show promising results.

Future research focuses on improving the flexibility of the proposed method and extending its capability by:

- Modifying it to generate several study plans of the same quality and with different structures.
- Developing a module that helps students and academic advisors to analyze the generated study plans and select the most effective and convenient plan.
- Developing a software tool that supports the method. This includes implementing a branch-and-bound method designing the database component.

## ACKNOWLEDGMENT

The author wish to thank Faculty of Computers and Information, Menoufia university, Egypt for their support.

## REFERENCES

[1] P. E. Beal, L. Noel, "What Works in Student Retention," *American College Testing Program*, Iowa City, Iowa, 1980.
[2] R. R. Hashemi, J. Blondin, "SASSY: a Petri Net based student-driven advising support system," *The Seventh International Conference on Information Technology, Las Vegas*, Nevada, pp. 150-155, April 2010.
[3] D. Yarbrough, "The engagement model for effective academic advising with undergraduate college students and student organizations," *Journal of Humanistic Counseling, Education and Development*, vol. 41 Issue 1, pp. 61-68, 2002.
[4] R. Shatnawi, Q. Althebyan, B. Ghalib, M. Al-Maolegi, "Building A Smart Academic Advising System Using Association Rule Mining, "*arXiv preprint arXiv:1407.1807*, July 2014.
[5] O. Marques, Xundong Ding and S. Hsu, "Design and development of a Web-based academic advising system," *31st Annual Frontiers in Education Conference. Impact on Engineering and Science Education. Conference Proceedings (Cat. No.01CH37193)*, Reno, NV, vol. 3, pp. 6-10, 2001.
[6] F. H. Grupe, "Student advisement: applying a web-based expert system to the selection of an academic major," *College Student Journal*, vol. 36 Issue 4, 2002.
[7] M. Talal Al-Nory, "Simple Decision Support Tool for university academic advising," *2012 International Symposium on Information Technologies in Medicine and Education*, Hokodate, Hokkaido, pp. 53-57, 2012.

[8] M. Patankar, "A rule- based expert system approach to academic advising," *Innovations in Education and Training International*, Vol. 35, Issue 1, pp. 49–58, 1998.
[9] D. Pokrajac, M. Rasamny, "Interactive Virtual Expert System for Advising (InVEStA)," *36th Annual Frontiers in Education Conference*, California, pp. 18–23, 2006.
[10] M. A. Siddiqui and S. Gemalel-Din, "Evaluation of Academic Plans of Study Using Data Mining Techniques," *2013 IEEE 13th International Conference on Advanced Learning Technologies*, Beijing, pp. 224-228, 2013.
[11] M. S. Laghari, G. A. Khuwaja, "Student advising and planning software," *International Journal on New Trends in Education & their Implications (IJONTE)*, Vol. 3, Issue 3, pp. 158–175, July 2012.
[12] Safak Kiris, "AHP and multichoice goal programming integration for course planning," *International Transactions In Operation Research*, Vol. 21, Issue 5, pp.819-833, September 2014.
[13] V. Gonzalez, D. Esparza, "Work in progress: advising tool to improve the time for graduation and the transfer of students from a community college to engineering school," *The 40th IEEE Frontiers in Education Conference (FIE'10)*, Washington, D.C., 2010.
[14] Simon Kristiansen, Matias Sørensen, Thomas R. Stidsen, "Elective Course Planning," *European Journal of Operational Research*, Vol. 215, Issue 3, pp. 713-720, , 16 December 2011.
[15] Abdallah Mohamed, "A decision support model for long-term course planning," *Decision Support Systems*, Vol. 74, pp. 33-45, June 2015.
[16] LINDO Systems Inc., Available: http://www.lindo.com August 2017.
[17] M. L. Cheong, R. Bhatnagar, S. C. Graves, "Logistics network design with supplier consolidation hubs and multiple shipment options," *Journal of Industrial and Management Optimization*, Vol. 3, Issue 1, pp. 51-69, 2007.
[18] V. Jayaraman, A. Ross, "A simulated annealing methodology to distribution network design and management," *European Journal of Operational Research*, Vol. 144, Issue 3, pp. 629-645, 2003.

## Authors' Profiles

**Dr. Hamdy Ibrahim** is an assistant professor, Department of Information Systems, Menoufia University, Egypt. He is currently working as a postdoctoral fellow at university of Calgary, Alberta and instructor at SAIT Polytechnic, Alberta, Canada. He received his

Ph.D. in the field of Software Engineering, University of Calgary, Alberta, Canada, April 2011. He received bachelor´s degrees in Electronic Engineering from Menoufia University, Egypt. He received his master degree in computer electronics and communication from faculty of Engineering, Cairo university. His research interests include Intelligent transportation systems, Pattern recognition, Decision support for academic advising, Uncertainty management, Decision support under uncertainty, and Software release planning.