Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-17 (2022) 61–66

A receding horizon approach for curriculum management in higher education

Thomas N. Larsen **, **Riccardo Busetto** *, **Damiano Varagnolo** **,***, **Simone Formentin** *, and **Adil Rasheed** **

* *Politecnico di Milano, Italy*
** *Norwegian University of Science and Technology, Norway*
*** *University of Padova, Italy*

Abstract: We propose a dynamic approach for curriculum management in university programs, i.e., for deciding which teaching and learning activities should be performed and in which order, as classes are being executed, to better aid the students reach the intended learning objectives. The approach ladders on a continuous-time dynamical model of the learning status of the individual students on the individual skills to be taught during the program. Such a model includes constructivist viewpoints on learning and zone of proximal development effects. Updating the program structure is then cast as an opportune model predictive control task, together with a moving horizon estimator that constantly infers the knowledge status of the class from the assessments performed in class. The proposed closed-loop approach is shown in simulation to significantly outperform the classical open-loop one, i.e., fixing the program structure in advance.

Keywords: Learning analytics, curriculum design, personalized study planning, model predictive control, receding horizon estimation

1. INTRODUCTION

Courses in higher education programs teach a series of topics, each for a certain amount of time and at a certain complexity level. Especially in STEM subjects, the temporal sequences of such topics across the programs often reflect constructivist points of view. Some concepts ladder on others, e.g., Laplace transforms ladder on complex numbers and thus should be taught after them. Using the term Teaching and Learning Activities (TLAs) to indicate lectures, exercises sessions, labs, etc., we may approximate the temporal sequences of topics shown in Fig. 1.

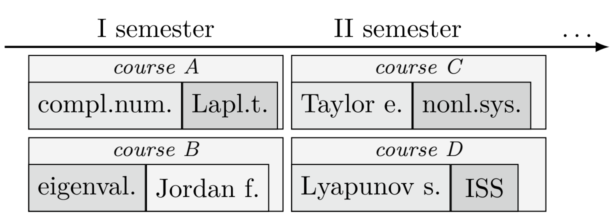

Fig. 1. A schematic representation of a STEM program with two courses in parallel per semester. Each course is composed of a series of TLAs (the various rectangles) dedicated to specific subjects. The length of each rectangle is used to represent how many hours a TLA is executed, while its shade indicates its difficulty level.

⋆ The research leading to these results has received funding from pedagogical funds at Norwegian University of Science and Technology, from the International Federation of Automatic Control, and the European Community through the Erasmus+ project 2019-1-NO01-KA203-060257 "Face It".

Curating the structure of such a program means reorganizing the TLAs logically and temporally, plus deciding their difficult levels so that the restructured program better promotes learning (in a sense to be defined).

Literature review. To the best of our knowledge, updating the curriculum has always been a complex and subjective task. There is some research on how students' engagement and learning are affected by different program designs, e.g., Tight [2012], Ashwin [2014]. There also exist two models that seem to be the most widespread ones: the first is called the *Objectives Model*, which starts from defining the intended learning outcomes as measurable performances, and the second being the *Process Model*, which starts from defining course contents and specifying criteria to assess students' knowledge of these contents. Several variations on these models exist (e.g., Tyler's, Wheeler's, and Kerr's Models) Gatawa [1990].

Likely the most renowned strategy is to follow the *black-box* approach to the sequencing of a curriculum [Crawley et al., 2014], which was initially proposed as an exercise for the teachers to understand better the connections among different parts in higher education programs within the CDIO standard to the management of university programs. This approach starts with the teachers representing every course within a program as a set of inputs (e.g., prerequisite knowledge and skills) and outputs (e.g., contributions to the final learning outcomes). Then the teachers meet to discuss the connections among such elements, and in this way, draw connections and gather intuitions for planning and improving the program. However, this tool

10.1016/j.ifacol.2022.09.225

is still qualitative, not based on quantitative indications, and does not even produce information that is not directly based on personal interpretations.

There do exist quantitative-based approaches to curate the learning flows within the curricula. Noticeably, several authors propose solving such tasks by starting with curriculum structure and coherence analyses based on opportune graph-based representations of the programs themselves, e.g., Aldrich [2015], Pavlich-Mariscal et al. [2019], Rollande [2015], and Varagnolo et al. [2021]. Such approaches tend to translate topological assessments of opportune networks into indications of what should be taught when (or better in which order). This also gives the possibility of creating individualized study plans for students. However, incorporating the students' current knowledge status into this management process and how to revamp this in a receding horizon fashion is not the focus of the just mentioned papers.

Other works are getting closer to the strategy we propose below. Noticeably, Yeralan and Büyükdağlı [2021] proposes to cast the problem as a linear programming one and develop a numerical tool that helps teachers' decision-making by enabling a quick evaluation of alternative approaches (a study that focuses on and accommodates many "what if" scenarios). Duarte et al. [2021] uses a combination of natural language processing, data visualization, and a classification of learning objectives to construct program plan representations that may be leveraged quantitatively, as we propose here below. Finally, Akbaş et al. [2015] analyzes curricula as directed graphs but adds analyses similar to those cited before with historical data of the students and courses – an approach that comes short of the one proposed here.

To summarize, there seems to be a widespread acceptance that curricula design and modification processes may benefit from data-driven tools that strive to give objective and context-independent information (see also Teixeira et al. [2020]). We claim that these management processes should be supported with up-to-date and objective information about the students current status to be effective [1].

Statement of contributions. To the best of our knowledge, there exists no approach based on a mathematically rigorous definition of the programs management problem [2] that is data-driven (i.e., explicitly using students' performance dynamically and recursively), and that exploits physics-oriented considerations (e.g., natural tendencies to forget subjects, if they are not repeated often enough).

We here instead propose a control-theoretical solution to solve the contents management problem as that of dynamically updating in a receding horizon fashion the temporal sequence, logical order, and difficulty levels of the TLAs while accounting for *a)* the current estimated knowledge levels of a given class of students, *b)* the target knowledge levels that the students should have at the end of the program, and *c)* a dynamical model of how the students nominally uptake and forget the various contents in time.

In this control-oriented framework,

- the dynamics of the uptaking/forgetting of the contents is an opportune ordinary difference equation accounting not only for such effects but also constructivist viewpoints about how the contents build on top of each other;
- the class's knowledge levels are estimated continually through a moving horizon estimator that leverages the model above together with data about the performances of the students on the various concepts they are studying;
- the dynamic update of the program structure is defined as an opportune model predictive control task, again based on the same model and the above moving horizon estimator.

In this paper, we thus discuss the potential benefits and limitations of the proposed strategy through in-silico considerations (while field tests are under execution, and results from these will be reported in an extended version of this manuscript).

Structure. The remainder of the paper is as follows. Section 2 will present a detailed mathematical model of the involved dynamics and the proposed receding horizon strategy. Section 3 will test the given closed-loop approach to curriculum management in a series of simulation case studies. The paper is ended by some concluding remarks.

2. A MODEL OF THE LEARNING PROCESS IN HIGHER EDUCATION

We now describe a simplified model of education that tries to capture the dynamic phenomena that are affected by programs management processes. By doing this, we aim at obtaining a quantitative model that can be used to cast a model-predictive management process. Each subsection below thus focuses on a specific phenomenon, and builds towards the final dynamical model (14) that will then be used to build the proposed model predictive control approach.

2.1 Modeling constructivist viewpoints on a program

We let $\mathcal{X} := \{x_1, \ldots, x_X\}$ be the set of topics taught in the program. Simplifying, a toy program may be

$$\mathcal{X} = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} \text{complex numbers} \\ \text{Fourier transforms} \\ \text{Laplace transforms} \\ \text{eigenfunctions} \end{Bmatrix}. \quad (1)$$

To model constructivist viewpoints on $\mathcal{X}$, we assume that a concept x_a may depend on another x_b by

- **being a prerequisite,** in the sense that x_a is a prerequisite for x_b if learning x_b requires knowing x_a beforehand (Thus, x_a is a necessary but not sufficient

[1] They should also be on labor market conditions and available resources at the institution, as suggested in [Posey and Pitter, 2012]. However, this is currently out of our scope.

[2] Note that the proposed methodology is also suitable for curating the contents of single courses. When we thus say "program contents management problem," we refer to whole programs and single courses.

element for learning x_b). For example, complex numbers are prerequisites for Laplace transforms;

- **providing complementing knowledge,** in the sense that x_a complements x_b if knowing x_a helps learning x_b (even if x_a is neither a necessary nor sufficient condition for learning x_b). For example, the concept of eigenfunctions complements (and thus is useful for understanding) Fourier transforms, and vice versa.

The relations of what is a prerequisite for what may be collected in a matrix $\mathcal{P} \in \{0,1\}^{X \times X}$ where the a-th element in the b-th column of $\mathcal{P}$ indicates whether x_a is a prerequisite for x_b or not [3]. Similarly, the relations of what complements what may be collected in the matrix $\mathcal{C} \in \{0,1\}^{X \times X}$. For example, for the program in (1) meaningful choices for $\mathcal{P}$ and $\mathcal{C}$ would be

$$\mathcal{P} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \mathcal{C} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \tag{2}$$

(e.g., both complex numbers and Fourier transforms are prerequisites for Laplace transforms, and eigenfunctions provide complementing knowledge for learning Laplace transforms). In this paper we assume that the teachers of the program have already agreed on the structure and values of $\mathcal{X}$, $\mathcal{P}$, and $\mathcal{C}$.

2.2 Modeling the knowledge levels of the students

We consider a time-invariant class of students $\mathcal{S} := \{s_1, \ldots, s_S\}$ (e.g., a class of three students, $s_1 = $ ann, $s_2 = $ bob, and $s_3 = $ coy). With a slight abuse of notation, we indicate with $x_i^s(t) \in [0,1]$ the knowledge status of the individual student s about the specific topic x_i at time t. This enables indicating with the column vector $x^s(t) \in [0,1]^X$ the knowledge level of s about the whole program. For example, with this notation the knowledge level of $s = $ bob about $\mathcal{X}$ in (1) at time t may be

$$x^{\text{bob}}(t) = [0.9 \quad 0.8 \quad 0.2 \quad 0.0]^T . \tag{3}$$

Symmetrically, the column vector $x_i(t) \in [0,1]^S$ is used to indicate the knowledge level of the whole class about a specific topic i at time t. For example, the knowledge level of the whole class (Ann, Bob, and Coy) about $x_3 = $ Laplace transforms at time t may be

$$x_{\text{Lap.transf.}}(t) = [0.1 \quad 0.6 \quad 0.2]^T . \tag{4}$$

The knowledge level of the whole class on the whole program at time t (in the following, the "whole state" at time t) is then the column vector $x(t) \in [0,1]^{S \cdot X} := \left[x_1^T(t), \ldots, x_X^T(t)\right]^T$.

Note that the per-student and per-topic knowledge levels vectors $x^s(t)$ and $x_i(t)$ (and thus the whole state $x(t)$) are latent variables that need to be estimated from students' assessments data. We postpone describing how to represent such assessments formally below.

It would then be natural to assume that the ideal target of the teachers is to organize the program so that it makes the whole state $x(t)$ increase component-wise in time until it becomes a vector of ones as fast as possible. In the proposed approach we though let the ideal target that of maximizing some opportune statistics of such knowledge vector *at the end* of the program. Our intuition is indeed that if sacrificing some amount of knowledge before the program ends may help getting more students more knowledgeable at the end of the program, then this strategy shall be preferred. This means that we define the end goal of the program management strategy that of maximizing the L1 norm of the "end state", i.e., of $x(t_{\text{end}})$, where t_{end} is the time marking the end of the program [4].

2.3 Modeling the TLAs executed during the program

We let a program be a series of TLAs (e.g., a specific lecture, in-class exercise, lab assignment, etc.). We then represent a generic TLA as a $X \times 2$ dimensional matrix

$$a_j := [w_j,\ h_j], \qquad w_j \in \{0,1\}^X ; \quad h_j \in [0,1]^X \tag{5}$$

where the two column vectors w_j and h_j represent respectively which topic is involved in this assessment (thus the i-th element of w_j indicates whether the i-th topic is involved in the TLA a_j or not) and how advanced the TLA is (thus the i-th element of h_j indicates at which taxonomy level the i-th topic is being involved in a_j). E.g., let a_j be an exercise related to our toy program $\mathcal{X}$ in (1). Let a_j involve remembering the definition of Laplace transforms plus manipulating some complex numbers. Assume the teachers of the program $\mathcal{X}$ agree that the difficulty levels of such manipulations and of defining Laplace transforms are respectively [5] 0.5 and 0.2. a_j may then be represented as

$$a_j = [w_j, h_j] = \begin{bmatrix} 1 & 0.5 \\ 0 & 0 \\ 1 & 0.2 \\ 0 & 0 \end{bmatrix} . \tag{6}$$

Importantly, including the taxonomy levels h_j's within the description of activity a_j enables accounting quantitatively for the so-called *Zone of Proximal Development effects*, i.e., assume that a student s has a given pre-knowledge level $x_i^s(t)$ about concept x_i at time t, and that the teacher has to choose which TLA to execute so to help student gain further knowledge about x_i. Intuitively, if the TLA is too easy or too difficult for the student, then its usefulness for the learning purposes is suboptimal. Assuming proximal development effects means then assuming that to maximize learning there is the need of choosing activities "with the right difficulty at that specific time" (and since the knowledge of the student should hopefully increase in time, then such difficulty should also increase).

[3] See also Molontay et al. [2020] for alternative strategies on how to define such a matrix.

[4] One may want prefer to maximize some other norm, e.g., L2, elastic net, or even some weighted norm promoting a higher knowledge about some specific topics than others. This is however more a designer choice than a framework structuring one. So, without loss of generality, in this paper we keep the final goal that of maximizing the L1 norm of the end state.

[5] We remark that the process of defining such difficulty levels is a non-trivial task. These numbers should indeed be a sufficiently accurate description of how complex an activity is. Verifying this is though well beyond the scope of this paper, and left as a future work.

2.4 Modeling the program as a set of TLAs

We assume that the teachers have in the years prepared a set of TLAs, $\mathcal{A} \coloneqq \{a_1, \ldots, a_A\}$, where each element a_j is as above (i.e., $a_j = [w_j, h_j]$). Given this set, a program can be modelled as in Fig. 1 essentially the sequence $\boldsymbol{a} \coloneqq [a(1), a(2), \ldots]$ of which TLAs are to be done, potentially with replacement.

As a shorthand we will use the notation $\boldsymbol{a}(1 : k) \coloneqq [a(1), \ldots, a(k)]$. Note that we will use k to denote the sequencing index, and this is in contrast with the subscript j, that we instead use to indicate a specific TLA within the set of potential activities $\mathcal{A}$. This means that to each $a(k)$ corresponds a specific j so that $a(k) = a_j$. Moreover we assume the duration of the program to be fixed and known in advance (in other words, the total number of activities to be executed is given and fixed).

Note that the purpose of this paper is to devise a control-theoretic algorithm for the program contents management problem, i.e., designing and updating the sequence $\boldsymbol{a}$ starting from opportune estimates of the current latent class knowledge status $x(t)$ in a receding horizon fashion.

2.5 Modeling the assessments executed during the program

To formulate the contents management problem in a control fashion there is the need to close the loop. To do so, we assume that at the end of each TLA $a(k) = [w(k), h(k)]$ every student executes a quantitative assessment activity, and that this measurement is available for such control purposes[6]. Thus, after executing $a(k)$ the system collects a vector $y^s(k) \in [0, 1]$, $s = 1, \ldots, S$ of individual assessments (one for each student) where each scalar $y^s(k)$ indicates which score student s got after participating in $a(k)$ (0 indicating no success, 1 total success).

$y^s(k)$ is thus a random variable whose distribution depends on the student knowledge state $x^s(k)$. To explicit this distribution it is useful to define the *skill deficiency factor*, i.e., an indication of how ill-prepared student s is when participating in $a(k)$. To define such number, recall that $w(k)$ describes which topics are involved in the activity, while $h(k)$ at which difficulty levels these are involved. The *individual skill deficiency factor* may be then defined as

$$\max\left(h_j(k) - x_j^s(k), 0\right). \tag{7}$$

E.g., if $a(k)$ involves Laplace transforms at a taxonomy level 0.7 and Bob has a current knowledge state about this topic of 0.5, then Bob has a skill deficiency factor about Laplace transforms in that activity of 0.2. If instead $x^{\text{ann}}_{\text{Lap.Transf.}}(k) = 0.8$, then Ann has no deficiencies about Laplace transforms for that activity.

The factor (7) may be defined so to be multi-dimensional, i.e.,

$$\max\left(h(k) - x^s(k), 0\right) \tag{8}$$

represents the vector of individual deficiency factors obtained considering the max operator as component wise. Note that if some elements of $h(k)$ are be zero (i.e., the activity does not involve some specific topic) then this does not disrupt the meaning of the vector in (8), that shall be intended as a quantification of how unprepared a student is to participate in a specific TLA.

(8) may then be used to build the scalar

$$\frac{\left|\max\left(h(k) - x^s(k), 0\right)\right|_1}{\left|w(k)\right|_1}, \tag{9}$$

that scalarizes the skills deficiency vector above into a mean deficiency (with the mean being computed over how many topics are actually involved in $a(k)$). Given this mean deficiency, we assume the performance of student s at time k when participating in the activity $a(k)$ to be the random variable

$$y^s(k) = 1 - \frac{\left|\max\left(h(k) - x^s(k), 0\right)\right|_1}{\left|w(k)\right|_1} \tag{10}$$

Note that with this formulation, the skills deficiency score approximately corresponds to how much the assessment of a student will deviate from the perfect score 1.

2.6 Modeling the effect of participating in a specific activity on the knowledge status

Participating in activity $a(k)$ should ideally lead to improved knowledge. The zone of proximal development intuitions states that if a student has a given knowledge level $x_i^s(k)$ about topic x_i, then better to make them perform an activity $a(k)$ whose difficulty level $h_i(k)$ does not differ too much from $x_i^s(k)$. In other words, the smaller $|h_i(k) - x_i^s(k)|$, the higher the potential of the activity.

Thus, we model the per-topic per-student potential improvement brought by $a(k)$ as

$$\frac{\beta\varepsilon}{\beta + |h_i(k) - x_i^s(k)|}, \tag{11}$$

so that the perfect alignment $x_i^s(k) = h_i(k)$ leads to the best potential improvement of ε, a user-defined hyperparameter. Conversely, perfect misalignment ($x_i^s(k) - h_i(k) = \pm 1$) leads to a potential improvement of $\beta\varepsilon/(\beta + 1)$, with β as another hyperparameter.

However, the potential improvement shall also account for the following phenomena.

- If the student is deficient in prerequisite topics, the potential improvement factor shall diminish accordingly (since participating in an activity for which one does not know the prerequisites is less useful). Thus, the potential improvement shall also depend on the current knowledge status of the topics implicitly defined by the matrix $\mathcal{P}$.
- If the student is knowledgeable in complementing topics, the potential improvement factor shall increase accordingly (since participating in an activity

[6] This assumption is actually a big simplification: our anecdotal experience suggests that it is actually rare that a teacher collects and stores quantitative data about the performance of each student during the various lectures. It is even more rare that this data, if collected by an individual teacher, is stored and available for long term planning purposes[7]. In any case we are interested in understanding the value of such information – in other words, we want to check whether in this idealized scenario the value of such information is negligible or not. In the first case, then it would have no sense to try to modify the system. If instead this information seems crucial for solving effectively the programs management problem, then this result may drive the process of changing the system.

for which one has complementing knowledge is more useful). Thus, the potential improvement shall also depend on the states defined by the complementing matrix $\mathcal{C}$.

To discount the potential improvement depending on $\mathcal{P}$, we multiply (11) by

$$\prod_{j|\mathcal{P}_{ji}=1} x_j^s(k) \tag{12}$$

so that the closer the prerequisite knowledge to zero, the smaller the uptake about x_i can be. Ideally, knowing all the prerequisites perfectly leads to no reduction of the potential improvement factor.

To accelerate the potential improvement as a function of $\mathcal{C}$, we multiply (11) by

$$\prod_{j|\mathcal{C}_{ji}=1} \left(1 + \frac{x_j^s(k)}{\gamma C_i}\right) \tag{13}$$

with C_i counting how many complementing topics x_i has, i.e., $C_i := |\mathcal{C}_i|_0$, $\mathcal{C}_i$ being the i-th column of $\mathcal{C}$. Null knowledge about such complementing topics leads thus the potential improvement factor to remain as initially defined, while full knowledge about all these topics leads this factor to be $\left(1 + \frac{1}{\gamma C_i}\right)^{C_i}$. Note that here γ is another user-definable hyperparameter.

Finally, there is the need for multiplying everything by a factor $w_i(k)$, i.e., a number that is one if activity $a(k)$ involves topic x_i, or zero otherwise. To summarize, the temporal dynamics associated with the per topic knowledge levels of a generic student are

$$x_i^s(t+T) = e^{\alpha T} x_i^s(t) + w_i(k) \cdot \frac{\beta \varepsilon}{\beta + |h_i(k) - x_i^s(k)|} \cdot \prod_{j|\mathcal{P}_{ji}=1} x_j^s(k) \cdot \prod_{j|\mathcal{C}_{ji}=1} \left(1 + \frac{x_j^s(k)}{\gamma C_i}\right) \tag{14}$$

where $e^{\alpha T} x_i^s(t)$ accounts for memory decay over time, acting upon all skills.

2.7 *A receding horizon approach to solve the curriculum management problem*

Given models (14) and (10), the problem of estimating the learners' knowledge status may be solved using off-the-shelf Moving Horizon Estimator (MHE) approaches. Given an estimate $\hat{x}$, finding the currently optimal TLA sequence can also be solved using off-the-shelf Model Predictive Control (MPC) approaches. Note that the estimation approach establishes the class's knowledge estimates on a per-student basis. However, solving the MPC problem on a per-student basis would result in individualized courses; in this case, we thus consider the "average student" knowledge since considering the constraint that the class shall follow a unified program.

3. NUMERICAL RESULTS

We consider the problem of curating an individual course consisting of 60 hours of frontal lectures, executed on Mondays, Wednesdays, and Fridays for 2 hours each, teaching 5 skills (one may indeed let the individual skills as conceptually large as wanted, e.g., "linear algebra." One may thus intend these 5 skills as separate parts of the course). Such skills are so that the prerequisite and complementing matrices $\mathcal{P}$ and $\mathcal{C}$ defined as in (2) are

$$\mathcal{P} = \begin{bmatrix} 0&1&1&1&1\\ 0&0&0&1&1\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{bmatrix} \qquad \mathcal{C} = \begin{bmatrix} 0&0&0&0&0\\ 0&0&1&0&0\\ 0&1&0&0&0\\ 0&0&0&0&1\\ 0&0&0&1&0 \end{bmatrix}. \tag{15}$$

We finally assume that this course has a database of 500 TLAs, so that the first 100 are so that $w_j = [1,0,0,0,0]$, and $h_j = [\mathcal{U}(0,1),0,0,0,0]$, i.i.d., with $\mathcal{U}(0,1)$ the uniform distribution in $[0,1]$. The second 100 TLAs are instead so that $w_j = [1,1,0,0,0]$, and $h_j = [\mathcal{U}(0,1),\mathcal{U}(0,1),0,0,0]$. Similar extending patterns apply to the following three sets of 100 TLAs. The TLAs are sampled without replacement.

We then compare the dynamic management approach proposed above against a reference controller constituted by an open-loop approach executing 30 TLAs with a predefined order. To choose this order meaningfully, we note that the $\mathcal{P}$ matrix in (15) defines a natural order for teaching the various skills x_1 to x_5. The course shall thus start with 60/5 hours of TLAs that involve only x_1. The first 6 TLAs shall thus be so that $w_j = [1,0,0,0,0]$ and $h_j = [j/6,0,0,0,0]$. The consecutive set of 6 TLAs shall then be $w_j = [1,1,0,0,0]$ and $h_j = [(j-6)/6,(j-6)/6,0,0,0]$, thus increasing difficulty and involving more skills. This extending pattern applies to the following three sets of 6 TLAs.

We then simulate the dynamics of the to-be-controlled system equivalently for both the open- and the closed-loop, i.e., (14). This constitutes a logical flaw: we are indeed comparing an open-loop approach vs. a receding horizon estimation & control approach (both designed explicitly starting from (14)) via simulations that are again based on (14). This means that we are unfair to the open-loop approach. However, the results in this paper are intended to be qualitative and aim at showing that, for meaningful model parameters, the open-loop approach is dramatically outperformed.

For the sake of completeness, our choice of parameters and hyperparameters for instantiating model (14) is $T = 2$ [hours], $\alpha = 0.001$, $\beta = 0.25$, $\varepsilon = 0.25$, and $\gamma = 50$. These values have been chosen to reflect anecdotal experience from how a typical class, with a certain distribution of initial knowledge, arrives at a final knowledge after a course.

We thus compare the evolution of the aggregated performance index $\|x(t)\|$ in time for the two strategies in Fig. 2. As expected, the closed-loop approach performs noticeably better. The interesting point is though *how much better* it performs: since our choice of hyperparameters reflects anecdotal evidence about how well students currently uptake knowledge, the figure indicates that if the teacher had a wide enough set of TLAs and adopted the proposed approach, the same class would have learned much better.

The reasons why this happens may be explained by inspecting Fig. 3, showing the trajectories of the individual estimated and actual knowledge levels states. More pre-

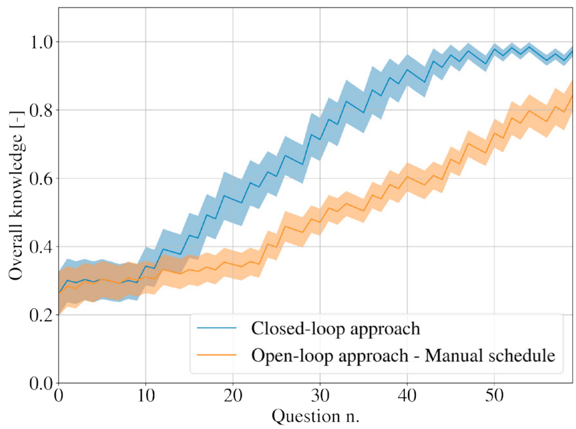

Fig. 2. Evolution of the aggregated performance index $\|x(t)\|$ in time for the two considered strategies.

cisely, the average knowledge of the class for the closed-loop approach increases faster, given the set of available TLAs. This results in more time available for refining the most advanced skills, typically those in classical courses with the most risk of suffering from misunderstandings due to weaker knowledge of their prerequisites.

Fig. 3. Trajectories of the average knowledge of the class per skill for the closed-loop approach (top) and open-loop one (bottom).

4. CONCLUSIONS

We defined a strategy for dynamically curating program contents while adapting to the current class' needs and accounting for the final learning target. The proposed closed-loop approach ladders on two main ingredients: a model of the learning dynamics and the availability of students' assessments. These ingredients enable coding an approach that is capable of outperforming classical strategies for which the list of which Teaching and Learning Activities (TLAs) shall be executed is immutable and defined at the beginning of the learning period.

The proposed strategy needs though information that is currently unavailable in real-life settings. We believe that this unavailability is the primary cause of immutable program structures. We further believe that by showing the untapped potential of such information quantitatively, this paper contributes towards implementing more systematic data collection in class.

REFERENCES

Mustafa İlhan Akbaş, Prateek Basavaraj, and Michael Georgiopoulos. Curriculum gps: an adaptive curriculum generation and planning system. In *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*, pages 1–11, 2015.

Preston R. Aldrich. The curriculum prerequisite network: Modeling the curriculum as a complex system. *Biochemistry and Molecular Biology Education*, 43(3), 2015.

P. Ashwin. Knowledge, curriculum and student understanding in higher education. *Higher Education*, 67(2): 123–126, 2014.

E. F Crawley, J. Malmqvist, S. Östlund, D. R. Brodeur, and K. Edström. The CDIO approach. In *Rethinking engineering education*, pages 11–45. Springer, 2014.

Rogério Duarte, Ângela Lacerda Nobre, Fernando Pimentel, and Marc Jacquinet. Broader terms curriculum mapping: Using natural language processing and visual-supported communication to create representative program planning experiences. *arXiv preprint arXiv:2102.04811*, 2021.

B. S. M. Gatawa. *The politics of the school curriculum: An introduction*. College press, 1990.

Jay M. Lightfoot. A graph-theoretic approach to improved curriculum structure and assessment placement. *Communications of the IIMA*, 10(2), 2010.

Roland Molontay, Noémi Horváth, Júlia Bergmann, Dóra Szekrényes, and Mihály Szabó. Characterizing curriculum prerequisite networks by a student flow approach. *IEEE Transactions on Learning Technologies*, 13(3): 491–501, 2020.

Jaime A. Pavlich-Mariscal, Mariela Curiel, and German Chavarro. Cdio curriculum design for computing: a graph-based approach. In *Proceedings of the 15th International CDIO Conference*, 2019.

J. Posey and G. W. Pitter. Supporting the provost and academic vice president. *The handbook of institutional research*, pages 145–164, 2012.

Raita Rollande. *The Research and Implementation of Personalized Study Planning as a Component of Pedagogical Module*. PhD thesis, Riga Technical University, 2015.

André MH Teixeira, Aida OPDC Guerra, Steffi Knorn, Kjell Staffas, and Damiano Varagnolo. Computer-aided curriculum analysis and design: existing challenges and open research directions. In *2020 IEEE Frontiers in Education Conference (FIE)*, pages 1–9. IEEE, 2020.

M. Tight. *Researching higher education*. McGraw-Hill Education (UK), 2012.

Damiano Varagnolo, Steffi Knorn, Kjell Staffas, Eva Fjällström, and Tobias Wrigstad. Graph-theoretic approaches and tools for quantitatively assessing curricula coherence. *European Journal of Engineering Education*, 46(3):344–363, 2021.

Sencer Yeralan and Özge Büyükdağlı. A decision support system for curricula design. *Periodicals of Engineering and Natural Sciences*, 9(4):632–642, 2021.