Predicting Student Progression as a Markov Chain

Victor Carneiro Lima, Julia Alves Farias, Renato da Rocha Lopes, Matheus Souza, Alim Pedro de Castro Gonçalves

***Abstract*—Effective school management may benefit from accurate student progression models to optimize the allocation of instructors and teaching spaces. However, existing research predominantly focuses on Markovian models for year-based programs, leaving a gap in the literature for credit-based undergraduate courses. This paper proposes a framework for modeling the progression of a group of students in credit-based programs using Markov chains, without requiring extensive training. The research addresses three key questions: (1) Can student enrollment and progression be modeled as a Markov chain? (2) Does a multi-agent simulation of a Markov model accurately replicate student-group behavior? (3) Are the multi-agent Markov model simulation results akin to real student progression data?**

To answer these questions, we analyzed student progression data from an engineering program to extract probability distributions for course failure, class enrollment, and dropout rates. These distributions were then used to develop Markov chain models, which were simulated using a multi-agent approach and were compared the real data progression rates. Our findings indicate that a Markov chain model can effectively emulate student progression in engineering programs, and the results from our multi-agent simulation closely match the actual progression data. This framework provides a valuable tool for school management and course demand prediction in credit-based undergraduate programs.

***Index Terms*—Markov chains, School management, Multi-agent simulation.**

I. Introduction

ONE consistent difficulty in school administration is to accurately estimate the progression of students along a proposed credit-based program. The evolution of the students is often affected by personal, environmental and pedagogical variables that may lead to difficulties for allocating the scarce resources available at the university. For instance, the demand for a specific class may be higher than predicted, leading to a lack of seats and to students who cannot take that class. This, in turn, may cause student retention and, eventually, dropout. On the other hand, having an accurate predictor for course enrollment behavior and student progression provides the school administration with a valuable tool to identify problems and optimize resource (human, infrastructure and financial) allocation.

This paper proposes to employ Markovian models for student progression in a higher education program. As Markov processes describe a sequence of events for which the probability of the following event only depends on the current state [2], these models can be designed intuitively and may be able to provide a reliable estimator for student progression.

A. Brazilian Public Higher Education Model

The Brazilian public higher education system operates on a credit-based structure where students must apply directly to a specific, specialized degree program (such as Civil Engineering, Law, or Medicine) prior to admission. Once enrolled, students navigate a highly rigid curriculum tailored explicitly to their chosen profession. This curriculum is divided into mandatory courses, which form the inflexible core foundation of the degree and must often be taken in a specific prerequisite sequence, and elective courses, which allow for a minor degree of specialization within or closely related to their field. To maintain steady progress and manage the limited spots in these institutions, public universities enforce strict maximum completion times for every program. While a standard undergraduate degree typically requires four to five years of full-time study, students are granted a capped extension, usually about 50% more than the standard duration, to complete all required credits and graduate. Failing to finish the degree within this maximum timeframe, or repeatedly failing courses without demonstrating academic progress, results in a mandatory dismissal where the student's enrollment is permanently canceled. This stringent policy ensures that publicly funded resources are utilized efficiently, keeping students accountable to the structured timeline and continuously making room for new incoming classes.

It should also be noted that the number of spots available for each course in Brazil is also limited. This is in general due to limitations in the capacity of the classroom. Whatever the reason, the courses can only accommodate a limited number of students, and this value is determined by the time the students first try to register for the specific courses. For the administrator, figuring out how many students might need to

This work was partially funded by CNPq grant number 309849/2025-0 and FAEPEx grant number 2466/23.

Victor Carneiro Lima is with the Communications Department, School of Electrical and Computer Engineering, University of Campinas, Brazil. ORCID:0000-0002-2004-1948 e-mail:victorcl@unicamp.br

Julia Alves Farias is with the Communications Department, School of Electrical and Computer Engineering, University of Campinas, Brazil. ORCID:0009-0009-4697-2896 e-mail:juliaalfarias@gmail.com

Renato da Rocha Lopes is with the Communications Department, School of Electrical and Computer Engineering, University of Campinas, Brazil. ORCID:0000-0003-2618-5499 e-mail:rlopes@unicamp.br

Matheus Souza is with the Computer Engineering and Automation Department, School of Electrical and Computer Engineering, University of Campinas, Brazil. ORCID:0000-0002-1538-3361 e-mail:matsouza@unicamp.br

Alim Pedro de Castro Gonçalves is with the Computer Engineering and Automation Department, School of Electrical and Computer Engineering, University of Campinas, Brazil. ORCID:0000-0003-0843-7971 e-mail:alimped@unicamp.br

A preliminary version of this paper was presented in XLIII Simpósio Brasileiro de Telecomunicações (SbrT' 25), Natal, RN, Brazil, September 29 - October 2, 2025[1].

Digital Object Identifier: 10.14209/jcis.2026.13

Submission: 2025-11-12, First decision: 2026-01-30, Acceptance: 2026-07-13, Publication: 2026-07-27.

take a particular class becomes an important challenge: there needs to be a balance between the needs of the students, on the one hand, and, on the other hand, the limited resources available at the school, both in terms of the availability of instructors and the capacities of the classrooms.

B. Related Work

The literature features a variety of predictive and stochastic approaches designed to model student enrollment and academic progression, which can be broadly categorized into traditional year-based Markov models, granular credit-based frameworks, and data-driven machine learning alternatives.

A significant body of research utilizes absorbing Markov chains to track macro-level student movement through discrete academic stages. Early foundational frameworks, such as the planning models proposed by Massy [3], laid the groundwork for using probability transition matrices derived from historical data to estimate student flows. Following this paradigm, Brezavšček et al. [4] developed an absorbing Markov chain with five transient states (representing each program year) and two absorbing states to monitor quality and effectiveness indicators in a Slovenian higher education institution, mapping progression patterns alongside dropout probabilities. A structurally similar framework was adopted by Eldose et al. [5] to evaluate a four-year engineering program in India; by computing transition probabilities from historical frequency matrices, they determined the expected time spent in specified transient states and demonstrated how institutional interventions, such as remedial classes and mentoring, positively shift graduation and dropout rates. Beyond institutional evaluation, these serialized level-to-level transition models serve administrative planning functions, as demonstrated by Rouhi et al. [6], who applied a three-level bachelor's degree transition matrix to forecast future enrollment numbers and ensure accurate budget allocations.

To overcome the rigidities of tracking students strictly by chronological years, several researchers have shifted the modeling focus toward more granular academic and behavioral variables. Borden and Rajecki [7] expanded the state evaluation paradigm by incorporating student-specific characteristics, such as GPA and course-credit load, though they concluded that minor grade variations had a relatively low impact on final graduation rates. Addressing the limitations of year-by-year classification, Gandy et al. [8] proposed a credit-hour flow matrix that groups students by their percentage of program completion rather than completed calendar years. This approach allows administrators to monitor the student pipeline dynamically and isolate specific academic terms where major "leaks" (withdrawals or stopouts) occur. Taking granularity further into the behavioral domain, Witteveen and Attewell [9] deployed Hidden Markov Models (HMM) on longitudinal transcript data to uncover latent states tied to course-taking patterns. Their trajectories revealed that while graduating and non-graduating undergraduates attempt difficult technical courses at equal rates, successful degree completion within six years is highly correlated with a student's ability to systematically alternate between heavy and light course loads across semesters.

As an alternative to these stochastic frameworks, data-driven machine learning algorithms have emerged to capitalize on the increasing volume of digital academic data. Hardman et al. [10] proposed a Random Forest (RF) algorithm utilizing data from Manchester Metropolitan University, integrating datasets from management information systems, student records, and virtual learning environments (VLE). Their findings indicated that micro-behavioral metrics—such as the specific time of day a student accesses the VLE, their last login timestamp, and total document interactions—served as the strongest predictors of student progression.

Despite their predictive capabilities, these diverse methodologies share common structural limitations. As highlighted by Ewell [11], Markov-based models are highly sensitive to the specific period in which historical data is collected and frequently overlook population heterogeneity by grouping all individuals into a single, uniform population. Furthermore, with the exception of non-Markovian data-driven models and specific credit-hour adjustments, the majority of existing stochastic approaches assume a strictly serialized program model. Under this view, if a student fails a single component, they are modeled as failing the entire year and must redo the block, thereby failing to encompass individual elective courses, highly granular credit accumulation, or the complex, non-compulsory dynamics that drive modern student dropout.

C. Contribution of this Paper

In this paper, we propose a Markov multi-agent model for student progression, customized for credit-based programs. Our goal is to provide a tool that can extrapolate past observations, such as course-specific failure rates and dropout rates, and use these data to predict future course demand and the effects of program changes on resource requirements. Our key contribution is to provide a tool to estimate student progression behavior based on the simulated agents. Our model is different from the other documented models as it is designed to account for:

- **Credit-based model:** as state before, most papers deal with serialized models, which are more rigid with respect to student progression and course completion. In contrast, credit-based programs offer greater flexibility, which can be a two-sided coin: while students may benefit from more accessible pathways to graduation, this flexibility complicates school management and resource planning.
- **Additional modeling elements:** our model copes with additional factors, such as elective courses, dropout rates and seat-limits in courses. These are key elements when reliable enrollment rates must be predicted.

D. Paper Outline

This paper is organized as follows. In Section II, we present the model for student progression along the program. In Section III, we detail the model we used to emulate student dropout along the program. In Section IV, we detail the model for course selection and enrollment. Finally we present a comparison between modeled and real data for student progression in three programs from the University of

Course 1	Course 2	Course 3	...	Electives
2	1	0	...	4

Fig. 1. Markov chain state vector of a single student. On this example, the student has passed course 1, is enrolled in course 2 and is not enrolled neither have passed course 3. The student also has 4 elective credits.

Campinas in Section VI and conclusions and possible future developments in Section VII.

II. PROGRAM AND PROGRESSION MODELING

To emulate a student's progression along a higher education program, we simulate the completion of the credits needed for graduation. Most of these credits are divided into m mandatory courses; the remaining credits, denoted by o, must be completed with elective courses, which may be organized in groups. Thus, to graduate, a student must have all of the mandatory courses and simultaneously have completed the needed credits for each elective group. The mandatory credits cannot be exchanged for another similar course and the elective groups may restrict the courses that count for completion.

The core of our model is a structure that simulates the transcript of a single student. As the aggregate progression of credits depends exclusively on the enrollment and the success rate of each course, we can picture a state model depending on the needed courses for graduation. However, students may only pass a course that they have previously enrolled in, so we need intermediate states to store enrollment information, one for each course. We also must take in consideration the period of each student as the list of courses they must enroll depends on this parameter. On this paper we will develop a model for programs divided in six-month periods and credit-based. The final model correspond to a Markov chain of $3^{K(m+o)}+3$ states where m is the number of mandatory courses, K is the maximum number of periods a student may stay in the program and o is the number of elective credits the student must acquire to graduate. The current position on the markov chain is stored as a vector of length $m+1$ codifying the student transcript, as shown on Fig. 1. The base of number of states is proportional to a power of base three because we consider three possibilities for each class on the program: not enrolled, enrolled and approved. A not enrolled student is a student who never enrolled or enrolled and failed the course. An enrolled student is taking the course on the observed semester and a passed student has completed the course on a previos semester. The relation of these states for a single course is presented on Fig. 2

There are three more absorbing states, one to model dropout, as presented in section III, one to model students who exceed their maximum time limit to graduate and the absorbing state that represents students who completed the program. Further explanation on why we are using two separate states for dropout and termination are on section III. A progression schematic is presented on Fig. 3. The size of this problem indicates a non-scalable need for computer resources. However many of these states are unreachable (many courses have pre-requisites, locking out intermediate states corresponding to a

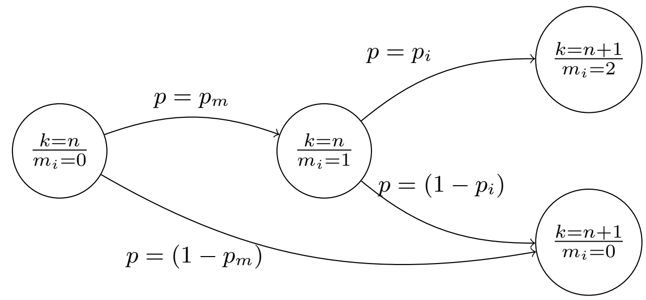


Fig. 2. One student enrolled on the i^{th} course at the n^{th} semester may transition to the passed course state with probability p_i or the not passed state, with probability $1-p_i$ at the $n+1^{th}$ semester. A student not enrolled must transition to the not passed state.

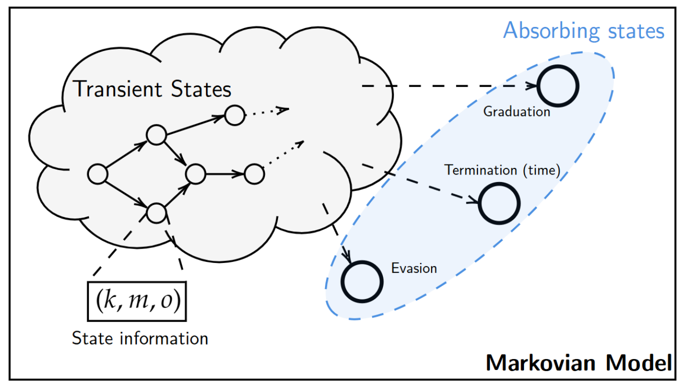


Fig. 3. Markov chain model of student progression. The chain simulates enrollment and completion of courses with three absorbing states: graduation, dropout, and exceeding the time limit. Intermediate states represent courses with enrollment and completion status, while prerequisites create a sparse transition matrix.

course the student cannot enroll in) and students cannot reach a state corresponding to passing in a course they have not previously enrolled. These unreachable states are encoded by making the transition probabilities to the enrolled state of a blocked course and to the passed state of a not-enrolled course both 0. This feature makes our Markov transition matrix highly sparse.

In fact, the only non-zero entries of the transition matrix will be:

- If the current state is at period k before enrollment in courses:
  - The entry that leads to the maximum timeframe absorbing state if $k > K$ — in this case all other transitions have probability 0;
  - The entry that leads to the dropout absorbing state with the corresponding probability detailed in section III;
  - The entries leading to states that model enrollment in the available courses at the period k, given the history of completion with probability distribution detailed in section IV.
- If the current state is at the k-th period after enrollment:
  - The entries that lead to all possibilities of passingin enrolled courses at period $k+1$, including the final graduating state. The probabilities are given by

historical rates for passing or failing in the enrolled courses.

For our simulation we chose to simulate multiple agents through the Markov chain. This choice allows us to extract the average progression along the simulated student path and compare it to the progression of actual students along the program. In our implementation we considered only the non-zero probabilities and used the resulting Markov chain to simulate the possible student progressions in the course, which correspond to possible paths in the Markov chain. The computational complexity of this simulation will be discussed in section V. To implement our model, each admission season, N students are created with empty transcripts and a set of graduation requirements. This flexibility of requirements allows us to model requirement changes and to observe the effects of the transition between programs. For each of the N students the current state is stored as an encoding vector as presented on Fig. 1, where each of the first m entries correspond to the mandatory courses. The value of each entry encodes if the student has not passed nor is enrolled, is currently enrolled or has passed the corresponding course. For each course in the mandatory group, we estimate a success rate from the academic records.

At each semester, a transition model is applied to evaluate if a given student will pass in an enrolled course. The enrollment process will be presented with more detail in section IV. The transition model will update each of the requirements components and the student will graduate when all graduation requirements have been met. This final state is absorbing and there is no transition from it.

A. *Model extension*

Some questions may be raised concerning the limitations of our model and the estimated transition probabilities. The first one is that once a student has failed one course, the probability of passing on the second try may differ from the probability of passing on the first try, and this difference was not taken into consideration on our Markov chain. Indeed we observed a correlation between the number of failed attempts and the passing rate on our empirical estimations. This behavior may be modeled as new intermediate states in which we store the number of failed attempts for each course, limited to a maximum number of attempts, t, for which the statistic relevance of the historical data used to estimate the probability starts to decline. In this case, the number of states will be $3^{K(tm+o)}+3$. For simplicity, we have not modeled this effect in our simulations.

The other consideration is that often dropout probabilities are correlated to failing students. Thus, considering that the probability of dropping out is independent from the performance and all students may dropout uniformly is not representative of the real situation. Again, this is correct; however our model is designed to be accurate on a large scale simulation, with kn students. For an individual simulated student, this assumption may lead to a scenario where a straight-A student drops out, which is a rare occurrence in our real data. However in a large scale, the model tends to converge to an accurate representation of reality.

III. Dropout modeling

One key element of mass enrollment prediction is dropout tendencies. Assuming that graduation rates are not 100%, if no student changes programs or stops attending, the failure rates would create an inflating demand for courses at the beginning and the middle of the program and lead to an unstable model with a constantly increasing number of students. On the real programs, this effect is addressed with a limit on the number of periods a student can be enrolled at the program. We consider two pathways to the dropout absorbing states: compulsory termination and non-compulsory termination.

Compulsory termination occurs when a student exceeds the number of periods by the program's regulations. These transitions are certain, happening with probability one. This behavior places an upper bound on the total number of enrolled students and sets the maximum simulation size to be $k \times N$ students, in which k is the maximum period allowed and N is the number of new students per period. Although the theoretical limit is $k \times N$, in practice the number of students present at any given time is much smaller, as most students graduate before reaching the k-period limit.

Non-compulsory termination refers to the dropout rate of students, a behavior that is more challenging to model due to the many factors influencing this decision. Although we are trying to impose a model for this effect, non-compulsory termination is often influenced by socioeconomic factors that cannot be accurately captured by deterministic models, such as financial hardship, physical and mental health issues, employment and immigration status, among others. In this paper we model dropouts using a Poisson distribution, where the probability of a student dropping out depends on the period in which they are enrolled. For each program, the conditional probability of dropping out in a given semester must be estimated for this model. This estimation is based on statistical data recording the number of students who dropped out in each period. Using this data, we applied the Parzen window method [12] for kernel density estimation and derived the beta distribution of dropout probabilities. We chose to model p_e as beta distributed random variable because of the small sample size of dropouts. This value gives the probability of a transition from a pre-enrollment state at semester $n-1$ to the dropout absorbing state. Given the many nuances of dropout and termination behavior, we chose to separate them in two different absorbing states, providing school administration with a better tool to address each problem.

IV. Enrollment modeling

The enrollment process is based on the suggested completion track for each program. This track is divided into periods with selected courses for each period. The courses have prerequisites that limit which course may be taken on each period and there is a hard limit on the number of credits that a student may take on each period. The goal of the enrollment process in our model is to determine which courses a student will take on each period. This decision will be based on past statistical data and the rules of the program.

First, we list all the mandatory courses for which the student has the needed pre-requisites. All the transition probabilities to intermediate states that represent enrollment in courses in which the students do not have the prerequisites are zero. Then we divide these courses into 3 groups: Courses that are delayed (that is, a student at the current period was expected to have already passed this course), courses on-schedule and future courses. The student will enroll on delayed and on-schedule courses until a maximum of C credits is reached. This limit is usually specified by the program rules. If C is not reached within these courses, the transition from the current state to all the intermediate states that do not represent enrollment in all these courses will be zero. If the credit limit is reached at this stage, the transition probabilities to all combinations between these courses are drawn from a uniform distribution.

After the mandatory courses, the remaining credits must be distributed between the future mandatory courses and electives. In both cases, only courses for which the student has the necessary prerequisites are considered. The enrollment in a future mandatory course is modeled as a Bernoulli distribution with probabilities determined by past statistical data depending on which period is being simulated. The enrolled course is drawn uniformly from all available courses. The elective selection is modeled as a Poisson random variable with λ equal to the number of elective courses predicted by the completion suggestion until the n^{th} period minus the already obtained elective credits by the simulated student. Given the number of elective courses available and the irregular availability, all electives are modeled as a single course, depending only on the number of credits associated.

In summary, for each course, the associated enrollment probability will be:

- The probability of enrollment on all m past and present mandatory courses:
  - 1 if credit limit is not reached;
  - $\frac{1}{m}$ if credit limit is reached.
- The probability of enrollment in a future mandatory course:
  - 0 if the credit limit is reached on mandatory courses;
  - $\frac{p}{o}$ with the probability p of early enrollment and o the number of available future courses otherwise;
- The probability of enrolling in elective courses:
  - $\frac{\lambda^k e^{-\lambda}}{k!}$ to each elective group corresponding to k credits until the credit limit is reached;
  - 0 for k greater than the credit limit.
- The probability of enrollment on any course for which the student does not have the necessary pre-requisites is 0.
- The probability of enrollment on any completed course is 0.

The transition to each possible state is given by the association of these probabilities for each course represented on the state vector for all combinations of courses and states. Similarly to the transition probabilities after enrollment, this task requires a monumental amount of computational resources, however the vast majority of states will have 0 transition probability, and the transition matrix is sparse, reducing the required computational needs. This sparsity of the transition probabilities allows the efficient implementation of the multi-agent simulation instead of the calculation of the full transition matrix.

Model extension

Several adaptations to the proposed model could be implemented, and some are outlined in this section. Elective courses could be grouped into different probability categories based on students' preferences for enrollment. The behavior of students' elective choices could be further analyzed to refine the weighting of these groups.

For mandatory courses, there is a probability that a student may intentionally delay taking a course, often due to its difficulty or the preferences related to the instructor who will teach that course in a particular semester. This factor has not been modeled here. However, in the results section, we will address this behavior by comparing the real data to the model's output.

V. Implementation and computational complexity

Our model implementation takes in consideration the sparsity of the transition matrix. Although it is possible to determine all transition probabilities to simulate progression on our Markov chain, we chose instead to simulate one agent at a time for each step, calculating only the non-zero probabilities described on sections II, III and IV. This implementation is based on creating a class *student* with features regarding the progress of the student along the program. Each admission season, N objects of the student class are created and added to the simulation pool.

These objects are initialized with attributes $k = 1$ regarding the first simulated period, H is an empty vector that represents the student transcript, Pr is the simulated program and contains all the mandatory courses for each period, e_t is a variable to record all the elective credits achieved, initialized as 0.

Algorithm 1 Student period step

INPUTS: p_e,p_f,λ,p_D,c_{MAX}
ATTRIBUTES: k,H,Pr,e_t.
  $e \leftarrow$dropout(k)
  IF e
    return inactive,evaded
  $M, e_l \leftarrow$ enrollment(k, $c_{MAX}, H, Pr, p_f[k], \lambda[k]$)
  $H, e_t \leftarrow$ progression(k, M,p_D,e_l)
  IF H is equal to Pr and $e_t >$ elective minimum
    return inactive,graduated
  $k \leftarrow k + 1$
  IF $k > K$
    return inactive, max periods exceeded-
  return active

The simulation steps implemented in our model are detailed in Algorithm 1. This algorithm takes the aforementioned attributes and the vector p_e of probabilities of dropout per period, the vector p_f of probabilities of enrollment in a future class per period, the vector λ of expected elective courses per period and the vector p_D of approval rate of each course. Each sub-step is detailed on Algorithms 2,3 and 4. Algorithm 2 simulates the dropout transitions. It takes as a parameter the probability p_e of dropping out in period k drawn from a beta distribution [13] modeled as presented in section III. To simulate dropout we sample on a realization of a Bernoulli variable with parameter p_e. In case of success of the Bernoulli trial we consider that the student dropped out and is set as inactive, being removed of the simulation pool.

Algorithm 2 dropout model

```
INPUTS: k, p_e
    x ← U(0,1)
    IF x < p_e(k)
        e ← true
        return e
    ELSE
        e ← false
        return e
```

Algorithm 3 simulates the enrollment process described in section IV. After enrollment, it returns a vector M with the enrolled disciplines for this student. This process emulates the transition from pre-enrolled states to enrolled states described on the previous section.

Algorithm 3 Enrollment model

```
INPUTS: k, c_MAX, H, Pr, p_f, λ_k
    c ← 0
    FOR i = 1 : k − 1
        FOR D in Pr[i]
            IF D is not in H
                M ← [M, D]
                c ← c + c_D
    FOR D in Pr[k]
        IF D is not in H
            M ← [M, D]
            c ← c + c_D
    x ← U(0,1)
    IF x < p_f(k)
        FOR i = k + 1 : l
            FOR D in Pr[i]
                IF D is not in H
                    M ← [M, D]
                    c ← c + c_D
    e_l ← poisson(λ_k)
    e_l ← min(c + el, c_MAX)
    return M, e_l
```

Algorithm 4 simulates the passing or failing of courses on enrolled vector M. It takes in consideration passing rates of each course. After this simulation we evaluate the possible conclusion of the program and the maximum periods rule. If any of these conditions are met, the agent is removed to the corresponding absorbing inactive state. Otherwise, the student remains active and k is increased. This implementation codifies the progression of a student along a higher education program as a Markov chain, taking into account each individual course instead of the enrollment year.

Algorithm 4 Progression model

```
INPUTS: k,M,p_D,e_l
    FOR D in M
        x ← U(0,1)
        IF x < p_D
            H ← [H, D]
    e_t ← e_t + e_l
    return H, e_t
```

A. Computational complexity

The computational complexity is determined by how our model scales with respect to two factors:

- **program size**, defined by the number of mandatory courses m and by the number of elective courses o; and
- **the number of students**, N.

Adding a new course increases the model's state space, thus raising memory requirements. As discussed in Section II, if all possible states were included, the total would reach approximately $3^{K(m+o)}$. However, due to sparsity in the transition matrix, many states are unreachable, resulting in a much smaller set of meaningful states. Consequently, the simulation approach adopted here is both more efficient and better suited to cope with this structure.

In terms of space, $\mathcal{O}(Nm)$ variables are required to store the information for all agents and $\mathcal{O}(m)$ parameters are necessary to store the probability transition matrix, given that $o \in \mathcal{O}(m)$. Hence, the proposed approach demands a polynomial space in terms of both the number of students and program size.

In terms of time, for each period step and each student, $\mathcal{O}(m^2K)$ operations (searches and updates) are carried out due to enrollment and progression. As these operations are carried out for $\mathcal{O}(K)$ academic periods, the approach demands $\mathcal{O}(NK^2m^2)$ operations to simulate N students during the whole program. This number may be much smaller due to graduation or dropout. Thus, the proposed approach can be computed in polynomial time.

VI. Results

To assess our proposal, we compared the simulation to anonymized, compiled data from the University of Campinas registrar's office, specifically for Electrical and Computer Engineering students from 2013 to 2023. Data from 2020 and 2021 were excluded due to pandemic-related measures that introduced outliers. Input probabilities and pass rates were estimated using data from 2009 to 2019.

The programs at the School of Electrical and Computer Engineering admit 70 students for the daytime Electrical Engineering program, 30 for the nighttime Electrical Engineering program, and 90 for Computer Engineering. The simulation uses the same curriculum as in the real data, starting with an initial group of 190 students as if the program were beginning from scratch and adding new 190 students every other period.

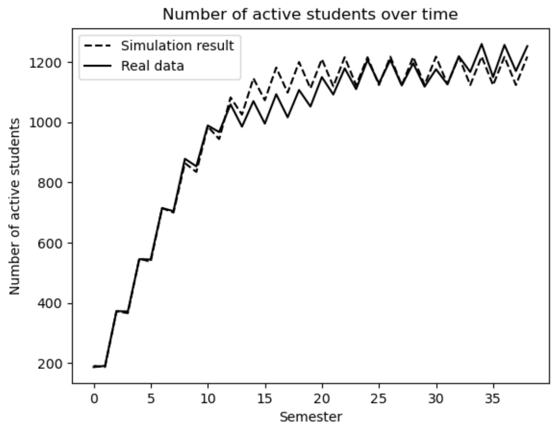


Fig. 4. Average active students per semester

The first characteristic evaluated was model stability. As the simulated students are transferred to the terminal states of graduation and dropout (compulsory or non-compulsory), the rate of new active agents should, at some point, reach an equilibrium with the rate of absorption by the terminal states. If our estimated transition probabilities are correct, this equilibrium point should be near the equilibrium of active students on real data. In Fig. 4, we observe that our model, after an initial transient response corresponding to the first simulated periods where none of the agents have yet completed the program, reaches an oscillation pattern around a constant number of active students. The oscillation occurs because new agents are admitted yearly, while transitions to absorbing states happen each period. This equilibrium regime represents the student body sample of the simulated program at any given period, containing students in different stages of program completion. Besides the model stability, we observe that the number of active students in steady state accurately follows the actual number of active students of the modeled programs.

After assessing model stability, we compared its outputs with real data. Our model reached a steady state after 14 semesters. Given that the three programs used for comparison have been running long enough to yield stable statistics, we ran our model for 24 semesters over 500 simulations, averaging the numbers of active, graduated, and dropout students at the end of this window. Tab. I compares the model's steady-state statistics with real data of the 24-th semester from a window of real data obtained with the registrar's office at UNICAMP. A slight disparity in dropout numbers is likely due to the small sample used to estimate transition probabilities and inherent statistical noise due to the multiple factors that influence the dropout pattern, as discussed on section III. To improve the individual detection of dropout risk, a more comprehensive analysis of each student is required, considering more factors than academic performance. A preliminary attempt to address this multifactorial analysis is presented on [14]. Course completion rates, however, are predicted with higher accuracy, as the number of graduated students is larger.

Tab. II compares the average time in semesters each student took to graduate. Agents simulated by the Markov model consistently took more time to graduate than real students. However, progression curves shown in Figs. 5 through 7 indicate that, while agents take longer to complete all credits, their progression through the program closely resembles that of real students. These results indicate that the difference in average completion time is due to agents not enrolling in elective credits, a direct consequence of the stochastic model for enrolled credits per semester. Fine-tuning the distribution parameters may help address this discrepancy.

TABLE I
NUMBER OF STUDENTS AT STEADY-STATE

Student Class	Simulation	Mean
Electrical Engineering (day)		
Active	409	358
Graduated	328	320
Dropout	103	163
Electrical Engineering (night)		
Active	200	178
Graduated	109	115
Dropout	51	80
Computer Engineering		
Active	171	153
Graduated	98	83
Dropout	32	18

TABLE II
AVERAGE COURSE COMPLETION TIME

Program	Simulated	Real data	
		Mean	σ
EE (day)	13.13	11.34	2.59
EE (night)	15.06	13.61	3.74
CE	14.55	11.85	2.67

Course completion curves indicate that our model reasonably simulates students' credit load preferences each semester. As discussed in Section IV, our model does not take into account intentional delays in course enrollments. This behavior is rather common among nighttime Electrical Engineering students. Fig. 6 illustrates this difference, with model agents taking more credits per semester than the students in this program, resulting in a steeper progression curve. Although this does not impact graduation rates, it slightly affects steady-state enrollment estimates, which could hurt management

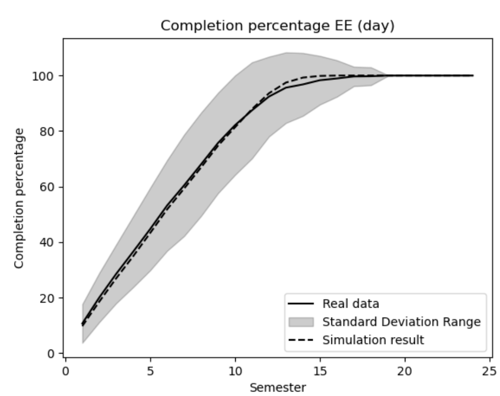


Fig. 5. Average course progression - EE (day)

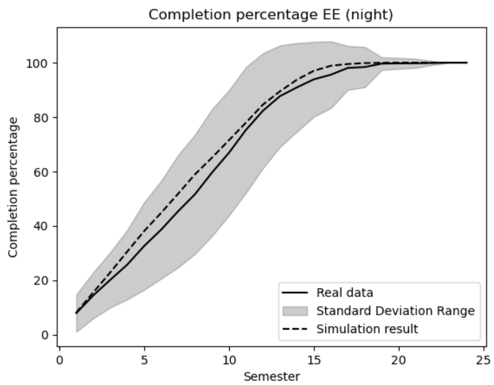


Fig. 6. Average course progression - EE (night)

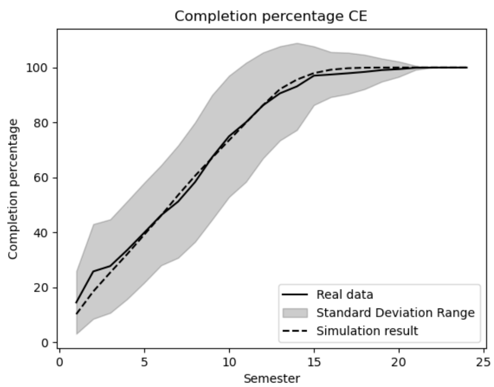


Fig. 7. Average course progression - CE

decisions based on the simulation. The progression curves indicate that our model has a progression pattern well within the standard deviation for the real progression for all three programs evaluated, providing an answer to our second and third research questions.

As a final test, we evaluated the model ability to accurately predict the individual student behavior for a future window. For this test, we randomly selected a semester from real data and loaded all active students at this semester as agents on the model. The total number of agents was 1220. Each agent was loaded with the respective transcript and current semester. We then simulated 5 semesters for this agents. At each step, the simulated transcript and course evolution was compared to real data for that student. As figures of merit, we evaluated the mean absolute deviation of course progression, defined as the percentage of the completed credit requirements, across all agents and the mean Hamming Distance [15] between the vectors encoding each agent transcript. These vectors have a value of one for completed courses and zero for courses that the student is yet to pass. The results are presented on Tabs. III and IV, for each semester of simulation after the initial setting.

Both results indicate that our model is capable of predicting the evolution of a single student through the program for a small window of semesters. As the window increases, the prediction quality is degraded. For a single semester, the

TABLE III
STUDENT PREDICTED COMPLETION PERCENTAGE ABSOLUTE DEVIATION

L	Mean AD	σ_{AD}	Median AD
1	0.05	0.07	0.03
2	0.08	0.09	0.06
3	0.12	0.11	0.09
4	0.13	0.12	0.1

TABLE IV
STUDENT PREDICTED TRANSCRIPT HAMMING DISTANCE

L	Mean HD	σ_{HD}	Median HD
1	4.27	3.43	4
2	6.87	4.26	7
3	9.2	5.12	9
4	10.95	5.74	11

median absolute deviation of course completion percentage is 3%, with a difference of 4 courses in a universe of 63 mandatory courses.

VII. CONCLUSIONS AND FUTURE WORK

This paper presented a Markov multi-agent model to simulate student progression within a higher education credit-based program. We proposed probability distributions for key events, such as dropout, course failures, and course enrollments, and described the states and transition probabilities necessary for constructing a Markov chain transition matrix. Following the model development, we assessed its computational complexity and found that it can be effectively scaled with reasonable resource requirements. This conclusion answers the first research question on our paper, proving that a higher education program can be modeled as a Markov chain.

To validate the model, we compared its outputs with real data from students enrolled in Electrical and Computer Engineering at the University of Campinas. Our findings indicate that the model satisfactorily emulates student progress and predicts group behavior, providing useful insights for school management and resource allocation, answering the other two research questions presented on our paper. Further development could enhance the model by incorporating program-specific characteristics and refining dropout modeling, which would increase its accuracy.

Finally, it is critical to acknowledge that student academic progression is influenced by a much larger universe of factors beyond the explicit, observable variables modeled in this study, such as accumulated credits, elective choices, and classroom seat capacities. In reality, a multitude of latent variables—including socioeconomic background, psychological well-being, and personal motivation—continuously shape student behavior and institutional performance indicators. While accounting for these unobserved dynamics falls outside the primary scope of this work, future research will focus on expanding this multi-agent architecture to incorporate hidden states capable of capturing these latent factors. Analyzing and classifying the effects of these hidden states according to their systemic importance represents a promising avenue to enhance the model's predictive power. Ultimately, this development will deepen the framework's utility for strategic school planning and resource management, allowing institutions to better anticipate course demands and optimize the overall cost-benefit ratio of the graduation process.

REFERENCES

[1] V. Lima, J. Farias, R. Lopes, M. Souza, and A. Gonçalves, "Abordagem multiagente para modelagem da progressão de estudantes," *XLIII Brazilian Symposium on Telecommunications and Signal Processing*, 2025. DOI: 10.14209/sbrt.2025.1571137448.

[2] S. Kay, *Intuitive probability and random processes using MATLAB®*. Springer Science & Business Media, 2006. DOI: 10.1007/b104645.

[3] R. H. Stein, "Planning models in higher education," *Higher Education*, vol. 11, no. 6, pp. 725–729, 1982. DOI: 10.1007/BF00129129.

[4] A. Brezavšček, M. P. Bach, and A. Baggia, "Markov Analysis of Students' Performance and Academic Progress in Higher Education," en, *Organizacija*, vol. 50, no. 2, pp. 83–95, May 2017. DOI: 10.1515/orga-2017-0006.

[5] K. Eldose, B. Mayureshwar, K. R. Kumar, and R. Sridharan, "Markov analysis of academic performance of students in higher education: A case study of an engineering institution," en, *International Journal of Services and Operations Management*, vol. 41, no. 1/2, p. 59, 2022. DOI: 10.1504/IJSOM.2022.121696.

[6] A. Rouhi, "Applying Markov-based forecasting in enrolment planning," *Journal of Institutional Research South East Asia*, vol. 2, p. 22, 2019.

[7] V. M. Borden and J. F. Dalphin, "Simulating the effect of student profile changes on retention and graduation rates: A markov chain analysis.," *Annual Forum of the Association for Institutional Research*, 1998.

[8] R. Gandy, L. Crosby, A. Luna, D. Kasper, and S. Kendrick, "Enrollment Projection Using Markov Chains: Detecting Leaky Pipes and the Bulge in the Boa," *AIR Professional File*, no. Fall 2019, p. 147, 2019. DOI: 10.34315/apf1472019.

[9] D. Witteveen and P. Attewell, "The College Completion Puzzle: A Hidden Markov Model Approach," en, *Research in Higher Education*, vol. 58, no. 4, pp. 449–467, Jun. 2017. DOI: 10.1007/s11162-016-9430-2.

[10] J. Hardman, A. Paucar-Caceres, and A. Fielding, "Predicting Students' Progression in Higher Education by Using the Random Forest Algorithm," en, *Systems Research and Behavioral Science*, vol. 30, no. 2, pp. 194–203, Mar. 2013. DOI: 10.1002/sres.2130.

[11] P. T. Ewell, "Recruitment, retention and student flow: A comprehensive approach to enrollment management research.," *National Center for Higher Education Management Systems*, 1985.

[12] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*, 2017. DOI: 10.1007/BF02985802.

[13] Y. Dodge, *The concise encyclopedia of statistics*. Springer New York, 2008. DOI: 10.1007/978-0-387-32833-1.

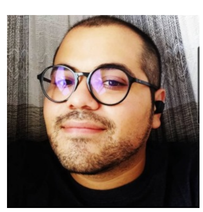
[14] J. Farias, V. Lima, M. Souza, and R. Lopes, "Modelagem preditiva de evasão escolar no ensino superior," in *Anais do LIII Congresso Brasileiro de Educação em Engenharia*, Associação Brasileira de Educação em Engenharia, Campinas, 2025. DOI: 10.37702/2175-957X.COBENGE.2025.6399.

[15] R. Roth, *Introduction to coding theory*. Cambridge University Press, 2006. DOI: 10.1017/CBO9780511808968.

Victor Carneiro Lima is a Post-doctoral researcher at the School of Electrical and Computer Engineering (FEEC) of the University of Campinas (UNICAMP). He obtained his B.Sc., MSc, and Ph.D. degrees from the same institution. He has experience as professor of electrical engineering topics for associates and undergraduate levels. His research interests includes topics on image processing, signal processing and higher education management.

Julia Alves Farias received her B.Sc. degree in electrical engineering from the University of Campinas (UNICAMP) in 2025. She is currently employed at Quinto Andar as a software engineer. During her graduation she researched student dropout modeling and prediction with machine learning techniques.

Renato da Rocha Lopes received his B.S. and M.Sc. degrees in electrical engineering from the University of Campinas (UNICAMP), Brazil, in 1995 and 1997, respectively. He received a M.Sc. degree in mathematics and a Ph.D. degree in electrical engineering from the Georgia Institute of Technology in 2001 and 2003, respectively. He is currently an associate professor at the School of Electrical and Computer Engineering at UNICAMP. His main research interest is in digital signal processing, especially in inverse problems.



Matheus Souza is an Assistant Professor at the School of Electrical and Computer Engineering (FEEC) of the University of Campinas (UNICAMP). He obtained his BEng, MSc, and Ph.D. degrees from the same institution. During his academic journey, he also spent time as a PhD visiting researcher at Maynooth University and worked as a Postdoctoral Research Fellow at University College Dublin. His main research interests focus on hybrid dynamical systems analysis and design and on convex optimisation.

Alim Pedro de Castro Gonçalves received the B.Sc. and M.Sc. degrees in Electrical Engineering from the School of Electrical and Computer Engineering, University of Campinas (UNICAMP), Campinas, Brazil, in 2000 and 2006, respectively. He received the Ph.D. degree from UNICAMP, in 2009. Since 2011, he has been an Assistant Professor at the School of Electrical and Computer Engineering, University of Campinas (UNICAMP). His research interests include nonlinear control, control and filtering of Markov jump systems.