A Flipped Classroom Approach for Teaching a Master's Course on Artificial Intelligence

Robin T. Bye*

Software and Intelligent Control Engineering Laboratory
Department of ICT and Natural Sciences
Faculty of Information Technology and Electrical Engineering
NTNU — Norwegian University of Science and Technology
Postboks 1517, NO-6025 Ålesund, Norway
Email: robin.t.bye@ntnu.no
Website: http://www.robinbye.com

Abstract. In this paper, I present a flipped classroom approach for teaching a master's course on artificial intelligence. Traditional lectures from the classroom are outsourced to an open online course that contains high quality video lectures, step-by-step tutorials and demonstrations of intelligent algorithms, and self-tests, quizzes, and multiple-choice questions. Moreover, selected problems, or coding challenges, are cherry-picked from a suitable game-like coding development platform that rids both students and the teacher of having to implement much of the fundamental boilerplate code required to generate a suitable simulation environment in which students can implement and test their algorithms. Using the resources of the online course and the coding platform thus free up much valuable time for active learning in the classroom. These learning activities are carefully chosen to align with the intended learning outcomes, curriculum, and assessment to allow for learning to be constructed by the students themselves under guidance by the teacher. Thus, I perceive the teacher's role as a facilitator for learning, much similar to that of a personal trainer or a coach. Emphasising problem-solving as key to achieving intended learning outcomes, the aim is to select problems that strike a balance between detailed step-by-step tutorials and highly open-ended problems. This paper consists of an overview of relevant literature, the course content and teaching methods, recent evaluation reports and a student evaluation survey, results from the final oral exams, and a discussion regarding some limiting frame factors, challenges with my approach, and future directions.

Keywords: Flipped Classroom, E-Learning, Active Learning, Constructive Alignment, Problem-Solving, CodinGame, edX, C-4 Dynamite for Learning.

* Corresponding author. This paper is an extended and revised version of a paper presented at the 9th International Conference on Computer Supported Education (CSEDU '17) in Porto, Portugal, April 2017 [13].

1 Introduction

A simple definition of the flipped, or inverted, classroom teaching methodolody is given as the act of swapping learning activities that traditionally have taken place in-class with learning activities that traditionally have taken place out-of-class [24]. However, this definition is somewhat limiting. As noted in a survey of research on flipped classroom by Bishop and Verleger [8], flipped classroom constitutes much more than a mere reordering of activities performed at home or in class. They give a more useful definition as the flipped classroom being "*an educational technique that consists of two parts: interactive group learning activities inside the classroom, and direct computer-based individual instruction outside the classroom.*" Furthermore, Bishop and Verleger [8] highlights that human interaction is the key component of the in-class activities, with a foundation in student-centred learning theories such as those of Piaget [31] and Vygotsky [42], where explicit instruction methods based on teacher-centred learning theories are outsourced to automated computer technology.

In similar vein, Abeysekera and Dawson [1] emphasise that in a flipped classroom, "*the information-transmission component of a traditional face-to-face lecture is moved out of class time* [. . . and replaced with] *active, collaborative tasks.*" By having students prepare for class by outside-class activities normally covered by traditional lectures, the teacher can free up valuable in-class time to student-centred learning activities. After class, students can continue working on in-class tasks they did not finish, explore some topics in more detail, revise material, and further consolidate knowledge [1].

But flipped classroom is not necessarily only about replacing traditional lectures outside of the classroom. Teachers can also take advantage of the existence of ready-made online material such as step-by-step tutorials, interactive web demonstrations, quizzes and tests, and coding platforms, to mention a few of the resources that teachers commonly spend much time preparing themselves. Thus, the teacher's focus can be shifted from the time-consuming task of making such resources from scratch themselves to selection and quality assurance ("cherry-picking") of online resources that the students can use for learning.

1.1 Active Learning

A common claim is that higher education is dominated by the *transmission method* of teaching, which can be popularly rephrased as *teaching by telling* [e.g., 38]. Synthesising research on the effectiveness of traditional lectures, Bligh [9] shows that they are not very effective for personal development, including skills or values, all of which are natural learning goals in higher education. Flipped classroom moves this one-way passive learning activity outside the classroom and replaces it with active learning, which can be defined as any teaching method that engages the students in the learning process [32]. However, as pointed out by Bishop and Verleger [8], this definition could in principle also include lectures where students reflect, take notes, and ask questions. Thus, to maintain a contrast with teacher-centred learning, it can be useful to separate lecture-based

methods from traditional lectures and define active learning as students participating in the learning process, doing more than just passive listening [10]. In line with this, active learning is indeed an inseparable part of flipped classroom [36].

Several active learning paradigms can be identified [e.g., see 8], for example *constructivism* and *collaborative learning*, which originate from the theory of cognitive conflict by Piaget [31]; *cooperative learning*, based on the theory of Vygotsky [42] on the zone of proximal development; *peer-assisted learning*, defined by Topping and Ehly [40] as "*the acquisition of knowledge and skill through active helping and supporting among status equals or matched companions*" and thus encompassing both the theoretical positions of constructivism and cooperative learning; and *peer-tutoring* [see 41, for a review]. Yet another active learning method is *problem-based learning*, which has overlap with learning methods under the umbrella of peer-assisted learning but can also be undertaken individually [8].

There are several metastudies that show that active learning in science, technology, engineering, and mathematics (STEM) has several advantages regarding performance, ability to reproduce material, and motivation and engagement [e.g., 18, 32, 37]. In particular, cooperative learning strategies have been shown to be particularly effective for achieving deep learning [e.g., 12, 17, 23, 39].

1.2 Constructive Alignment

For the last decades, there has been a dramatic change in higher education worldwide, with many more students enrolling, from a wider diversity of background, and with a broader range of approaches to learning [5]. For example, at my own institution, about 50% of our students in the bachelor programmes of automation, computer, and power systems engineering have a background from vocational school [30]. This means that they often have strong practical skills and good self-discipline, especially if they have been in the work force for some years before entering university, but it also means that they sometimes lack academic skills.

Today's engineering students conceive learning as anything from simple memorisation of definitions, applying equations and procedures, or making sense of physical concepts and procedures, to seeing phenomena in the world in a new way or changing as a person [26]. The big variation among students in this taxonomy, ranging from *surface learning* (memorisation) to *deep learning* (learning associated with understanding and ultimately changing as a person), with an added category of *strategic learning*, in which students aim for good grades with minimal effort [15], has necessarily had an impact on how higher education is being taught [16].

It is well documented that students' *approaches to learning* has a significant effect on achieving learning outcomes [e.g., 19, 27]. Therefore, many studies have tried to identify factors that promote deep learning [e.g., 28, 33]. A popular answer for dealing with the challenges of today's diverse student mass is the theory of *constructive alignment* (CA) [6]. CA merges the constructivist view

that students learn by doing with aligning the teacher, the students, the teaching context, the learning activities, and the learning outcomes [5]. This is achieved by the process of *working backwards* when designing a course, first starting with the *intended learning outcomes* (ILOs), then defining *assessment tasks* closely related to the ILOs, and finally choosing *teaching methods* and *learning activities* aligned with the ILOs and assessment tasks [5].

A particular concern of Biggs [e.g., 5–7] is how the students' approaches to learning are strongly linked to the assessment tasks. Many students tend to be strategic in their approach to learning and are often mainly concerned with the level of their final grade, not learning per se. The result is an approach to learning where the exam (the assessment tasks) effectively defines the curriculum, since only those parts of the course that will affect their final grade will be prioritised. The answer in CA is to incorporate assessment tasks with grades well aligned with ILOs to ensure students achieve the latter.

1.3 Cognitive Load Theory

As noted in the literature review by my colleagues Schaathun and Schaathun [36], an active learning approach adopts the constructivist view that learners must construct their own knowledge. However, at the same time, in order to move new information from a working memory with very limited capacity into long-term memory, deep cognitive processing is required [4]. This process, referred to as *cognitive load theory* [14], represents a bottleneck in learning when too complex problems and information must be processed in too short a time. According to Sotto [38], a remedy suggested by several authors is to break problems into small, manageable exercises in order to reduce the cognitive load. In flipped classroom, where typically online video lectures and other resources are offered outside the classroom, students can manage this cognitive load by regulating pace and rewatching video lectures [1], as well as completing their other learning tasks in their own order of preference, with possible repetitions if necessary. They will also be better prepared when turning up in the classroom and getting help on difficulties they have experienced off campus.

1.4 The Teacher as a Facilitator for Learning

A key factor for success in CA is a teacher who is able to create a learning environment that facilitates learning activities that in turn make the students achieve the desired learning outcomes [7]. All teaching and learning components such as the curriculum and the ILOs, the teaching methods, and the assessment tasks must be aligned to each other. However, despite such measures being taken, it is commonly accepted that lack of *self-monitoring* and *self-regulation* among students will lead to poor academic results [e.g., 11, 25]. Consequently, the learning environment itself is not sufficient to achieve ILOs, since students' individual skill in self-monitoring and self-regulation, that is, selecting and structuring the material to be learnt, will highly affect the learning outcomes [21].

Addressing this problem, Gynnild et al. [20] suggests that the teacher must adopt a role as a *facilitator for learning*, much similar to a personal trainer at the gym, guiding the trainee to do the right exercises, adjusting the "weights," or cognitive load, whilst encouraging and supporting the trainee, and eventually making the trainee self-monitored and self-regulated. In other words, the main job of the teacher is to guide the students along a path of learning, allowing for detours, and making sure that students achieve the intended learning outcomes at the end of the path, having walked themselves, and not having been lifted on the shoulders of the teacher.

Obviously, this is a bold ambition that can be challenging to achieve with large classes for which it is difficult to act as personal trainer and give individual advice and guidance. However, a flipped classroom approach may just be the tool that makes this possible, for example through systems that can automatically mark assignments and give immediate feedback on what the student struggles with, and suggest remedies such as topics to study more or exercises to be practiced.

1.5 Outline

In the following I first give an overview of a master's course on artificial intelligence (AI) in which I have employed several aspects of a flipped classroom approach in my own teaching (Section 2). Next, I present the teaching methods I have used (Section 3) and present a course evaluation that includes feedback from students in the autumn 2016 and spring 2017 cohorts (Section 4). Additionally, I present very recent results from an end-of-semester student evaluation survey of the 2017 cohort (Section 5) and final oral exam results from both cohorts (Section 6). Finally, I discuss some limiting frame factors and challenges to my approach, and point to possible future directions (Section 7).

2 Course Overview

The master's course *IE502014 Topics in Artificial Intelligence (TAI)* is a 7.5 credits (ECTS[1]) elective course (compulsory from 2018), which corresponds to a quarter of a full semester load. The course has been run with three different cohorts (autumn 2015 and 2016 and spring 2017) since the inauguration of a new master of science programme in simulation and visualization at NTNU in Ålesund autumn 2014. The course runs for 14 weeks during a semester, with all in-class activities confined to a single weekly teaching day, referred to as *workshops*. TAI provides an introduction to the field of AI and a number of selected topics therein relevant for solving real-world problems. The following key aspects are common across the topics being taught:

– *modelling* problems in suitable state space

[1] European Credit Transfer and Accumulation System.

- *design* and *implementation* of intelligent search, optimization, and classification algorithms
- *simulation* and *testing* of models and algorithms
- *visualization* and *analysis* of the results

The ILOs of TAI are defined within the three categories of knowledge, skills, and general competence. Specifically, for each category as indication, students should upon completion of the course be able to

- *knowledge*
  - describe AI as the analysis and design of intelligent agents or systems
  - explain terms such as perception, planning, learning, and action as fundamental concepts in AI
- *skills*
  - define problems in suitable state space depending on choice of solution method
  - solve problems by means of search methods, evolutionary algorithms, swarm algorithms, neural networks, or other AI methods
- *general competence*
  - collect information from scientific publications and textbooks and reformulate the problem, choice of methods, and results in a short, concise manner
  - discuss and communicate advantages and limitations of AI as a science

The course is split in two parts provided by two teachers. Part A, taught by me, gives an introduction to AI and revolves around solving problems by means of intelligent agents and the use of various search algorithms, including both uninformed and informed (heuristic) search, local search, and adversarial search. Part B, taught by my colleague Associate Professor Ibrahim A. Hameed, has a focus on optimisation and constraint satisfaction problems, and machine learning topics such as fuzzy expert systems, artificial neural networks (ANN), and hybrid intelligent systems. In this paper, the teaching methodology I apply for Part A is the most relevant but I may refer to Part B or the course as a whole where appropriate.

The main components of the course are as follows:

- course textbooks and scientific literature
- the learning management system (LMS) Fronter
- three mandatory assignments and a final oral exam
- online code development platforms
- massively open online courses (MOOCs)
- full-day in-class workshops

In the following, I will present each of these components in turn.

2.1 Textbooks and Scientific Literature

The course relies heavily on the following textbooks:

- *Artificial Intelligence: A Modern Approach* (AIMA) [35]
- *Artificial Intelligence: A Guide to Intelligent Systems* (AIGIS) [29]
- *Learning for Data: A Short Course* (LFD) [2]

The course description gives the freedom to experiment with presenting different topics from the broad field of AI each time the course is run. However, all the three times we have run the course, the following ten topics from the textbooks have been studied (with some slight variation), where topics 1–5 constitute Part A, and topics 6–10 constitute Part B:

1. Introduction to AI (AIMA Ch. 1)
2. Intelligent Agents (AIMA Ch. 2)
3. Solving Problems by Searching (AIMA Ch. 3)
4. Beyond Classical Search (AIMA Ch. 4)
5. Adversarial Search (AIMA Ch. 5)
6. Constraint Satisfaction Problems (AIMA Ch. 6)
7. Fuzzy Expert Systems (AIGIS Ch. 2–4, case study Ch. 9)
8. Artificial Neural Networks for Pattern Recognition (AIGIS, Ch. 6, case studies Ch. 9)
9. Hybrid Intelligent Systems (AIGIS Ch. 8, case studies Ch. 9)
10. Nonlinear Transformation and Feature Extraction (LFD Ch. 3)

In addition, relevant scientific papers are provided to students on selected topics as case studies to provide insight into practical application and state-of-the-art in AI.

2.2 Fronter LMS

Until summer 2017, Fronter[2] is the official LMS in use at NTNU in Ålesund, after which all campuses of NTNU will use Blackboard.[3] Every course has its own virtual "room" in Fronter that enrolled students are required to use for accessing course material and receiving updated news messages. The course room for TAI contains a continuously updated Frontpage with a summary of all relevant information, including a welcome message; information about where to buy the course textbooks; and course material (course overview, oral exam information, assignment handouts, and workshop material). The Frontpage also contains internal hyperlinks to the course material (usually pdfs) contained in a well-structured Documents directory. Students can quickly and easily access the material from the Frontpage, which is important if using a smartphone or tablet, but can also choose to navigate the Documents directory for complete access to all material. Assignments are handed in through the Fronter submission system and detailed feedback on students' performance is also posted there. Using the Fronter news functionality, we inform students whenever new material is available or when already posted material has been updated.

The teachers have tried to encourage the Fronter *discussion forum* by posting answers to several questions that students ask in class or by email but there has been very little activity from students in the forum. According to the students, this is due to being on-campus and seeing each other steadily, thus reducing the need for an online forum. Still, the teachers find the forum functionality useful but then acting more as a knowledge bank of frequently asked questions (FAQs) with answers.

[2] http://www.itslearning.eu/fronter
[3] http://www.blackboard.com

2.3 Assignments and Oral Exam

From experience with various coursework approaches at our department, we believe that compulsory coursework is very often required in order to make the students work steadily throughout the semester and be well prepared for the exam. Indeed, we have found that students' performance improved about one grade on average after making the coursework compulsory in some courses [30].

Three mandatory assignments must be passed for permission to enter the end-of-semester final oral exam. The assignments typically consist of both theoretical short answer questions and project-like programming (coding) exercises. Employing a grading scheme consisting of the letters A to F, where A is best and F is fail, a pass grade is awarded if the work has the quality of a C or better.

Whilst a requirement of C to pass the assignments may seem somewhat harsh, the reasoning behind this rule is to provide a strong incentive to students for studying hard before the exam. Furthermore, it should be noted that teachers tend to be somewhat lenient in grading, at least in borderline cases, and students usually are allowed at least one more attempt on an assignment they have failed.

The grades on the assignments do not contribute to the final grade, which is determined solely by a final individual oral exam. Students usually get 3–4 weeks to complete and submit their assignments on Fronter.

Individual assignment reports, as well as a laptop with computer code, can be brought to the exam to provide individual entry points of discussion. The oral exam is 25 mins and students are examined by the course teachers.

2.4 Code Development Platforms

With a focus on *learning by doing* and practical application, TAI requires coded simulation environments in which students can implement, test, and analyze intelligent algorithms. However, developing such environments can be a time-consuming task and divert attention from the learning outcomes that we want the students to achieve. For the programming exercises for Part A, I have relied on two online websites that provide such simulation environments, namely *HackerRank*[4] and *CodinGame*.[5] Students have to register as users (free of charge) on both websites.

Using HackerRank and CodinGame removes the need for writing a lot of boilerplate code and simulation logic, leaving the students more time to focus on the design and implementation of the algorithms, and the teacher more time to do other more meaningful work. Both websites support more than 20 of the most common programming languages, provide discussion forums, blogs, and leaderboards (rankings), with CodinGame also providing entertaining game-like visualizations of the code execution, which is quite useful when testing an algorithm and for debugging purposes. They both provide fun and playful environments that act as a catalyst for learning where students can compete both against themselves and others in the quest of obtaining better scores.

[4] https://www.hackerrank.com
[5] https://www.codingame.com

2.5 Open Online Courses

With the advent of MOOCs, numerous online e-learning resources exist, often free of charge. Two of the leaders and pioneers in the field of MOOCs are Udacity[6] and edX.[7] Udacity was perhaps the very first MOOC and was born out of a experiment at Stanford University by famous AI researchers Sebastian Thrun and Peter Norvig, the latter of whom is also the director of research at Google and co-author of the AIMA course textbook [34]). They offered an online course called *Introduction to Artificial Intelligence* online for free and managed to achieve a simultaneous enrolment of more than 160,000 students in more than 190 countries. EdX was founded by the Massachusetts Institute of Technology (MIT) and Harvard University in May 2012 and provides courses from more than 60 universities, corporations and institutions.

We *encourage* students to enroll for free in one or more of the AI courses provided by Udacity, and *demand* that they enroll, also for free, in the edX course *CS188.1x Artificial Intelligence* prepared by staff at the University of California Berkeley. The courses that we recommend build heavily on the course textbook (AIMA) by Russell and Norvig [34] and other textbooks on AI and contain numerous interactive video lessons, self-tests such as quizzes and multiple-choice questions (MCQs), programming exercises, and other material that encourage active learning and are useful for TAI.

2.6 Workshops

All in-class learning activities in TAI take place in an ordinary flat classroom on a single weekday that typically begins at 8:15 and ends at 16:00. Calling this full day of teaching and learning activities a *workshop* is not unintentional but emphasises to students that they will be met by an active, collaborative, student-centred learning environment for constructing their own learning, in contrast with more teacher-centred learning environments.

During the workshop, the teacher is neither required nor expected to be present at all times during the entire day but will have a detailed plan for the students to engage in learning activities (which will be discussed in the following chapter).

3 Teaching Methods

3.1 Information and Structure

Three important bits of information must be distributed to students before the very first workshop:

1. name of course textbooks and where to buy them
2. 10-page detailed course overview

[6] http://www.udacity.com
[7] http://www.edx.org

3. description of first workshop and preparatory homework

Naturally, many students fail to do this first homework and the workload of the workshop is therefore intentionally smaller than the remaining workshops to allow for this, with no homework reading, exercises, or video lectures.

Additionally, both teachers provided a detailed *list of potential oral exam questions* as early as possible in the semester. Posting such a list serves several purposes [30]: it reduces uncertainty and stress related to the exam but just as important, it provides students with *mental hooks* for constructing knowledge as they progress, since having read the questions will reduce the cognitive load when new material is introduced, and they will be able to more quickly putting new knowledge into context and storing it in long-term memory.

All of the above information is posted on Fronter in a highly structured way for easy and quick access, even when using smartphones and tablets. The course overview is rich in detail but is a document that we encourage students to refer back to regularly. It is useful for the students to have a schedule of when all the different topics will be introduced as well as assignment deadlines as early as possible so that students can plan their studies and reserve time for assignments. Introducing the teaching methodology with the emphasis on flipped classroom and active learning, with overarching ILOs and a well-defined main objective of the course, also provides a useful top-down view that guides the students during the course.

3.2 Workshops

Some time before each workshop (typically at least 4–5 days), a new workshop description is released on Fronter. Each workshop description contains details such as

- the date, classroom, start time, name of teacher;
- references to textbook chapters and online videos;
- the ILOs of that particular workshop;
- a rough schedule of tasks to be completed;
- details on each task; and
- and homework for the next workshop.

This homework consists of reading selected chapters and solving selected exercises in the textbook, as well as completing various tasks in the online edX course (see Section 3.3) or on the code development platforms (see Section 3.4).

Current Status Every workshop begins with a short evaluation of the current status regarding problems students have faced in their homework, questions they may have, what they think about the teaching methods, and other things that come to mind. Typically, there will be some parts from the homework that students have struggled with and have questions about. Minor questions are answered immediately, however, if it is clear that a more thorough explanation

is needed, the teacher writes down the questions or topics on a list and typically gives a *micro-lecture* on the various topics afterwards.

Next, students are given a number of active learning tasks, varying in size. For example, a two-minute task could be to discuss with another person in class the possible answers of a question I give them, whilst students can be working for up to several hours on larger exercises or assignments.

The teacher is not required to be available at all times. Indeed, it may be beneficial to leave the classroom at times, as internal discussion and group work seem to flourish with the teacher absent.

Self-Tests For *self-tests* (quizzes or MCQs), I usually ask everyone to do them individually, but encourage discussions with other students and myself. This has the advantage that those who prefer some time on their own to reflect on the answer get that, whereas those who prefer to interact with others in search of answers can do so. Afterwards, I ask students in turn for answers, and most importantly, the reasoning behind the answers. If the student's reasoning is weak, I ask for clarifications from the rest of the class, and finally, I often provide my own explanation, in different wording.

Digital self-tests has the advantage that *immediate feedback* is returned to the students. However, they also require questions to be unambiguously phrased. During a semester, there will very often be students who are dissatisfied with a question, claiming that another answer is more correct, or that there is no right answer. This should not be viewed as something unfortunate. Rather, it is a golden opportunity for discussion in class and a driver for deep learning. Students must have a thorough understanding of the topic of the question and be able to formulate to the teacher why they disagree with the question, and other students have to opportunity to delve into the discussion.

Oral Presentations For *oral presentations*, students will generally be working in groups, which tend to lower the fear of presenting and generally improves the quality of both quiz answers and presentations. Each group will randomly be assigned a topic for the presentation, for example, a particular search algorithm. Knowing that they will have to make a presentation in class but not knowing the topic beforehand likely acts as a catalyst for completing the homework in a serious manner. Also, making the presentation ad hoc in class and explaining the topic to others force the students to organise their knowledge and insight on the topics before turning up to class.

Naturally, the students' presentations will suffer from the short preparation time in the classroom. I therefore interrupt the presentations when needed and first ask the presenting students if they are able to clarify the issue at hand, and if not, I ask the class as a whole if someone can help out. Even if I get a good answer, I usually rephrase the answer with my own words and try to be as concise and precise as possible, and hopefully, both the presenters and the rest of the class get a much clearer picture of the particular issue.

Textbook Exercises Another activity during workshops include discussing answers to *textbook exercises*. These exercises differ from typical online exercises and self-tests, as they often require more work and are more time-consuming; one often have to make an analysis of the problem first, and then provide a long and elaborate answer. It would therefore be a waste of time to do such exercises in class, and instead, students are given a short list of recommended textbook exercises to complete at home before each workshop. In class, answers to the exercises are walked through and discussed as needed.

Programming Exercises Finally, being a course on topics in AI, writing computer code and implementing intelligent algorithms is an inherent and important part of the course. During workshops, on average, a large portion of the time is devoted to larger *programming exercises*, or *coding challenges*, contained in the assignments, or minor programming exercises that are solvable in a short period of time in class. More details on coding are provided in Section 3.4.

Competitions To add some spice to the learning environment, I sometimes arrange for small informal *competitions* in the class, where the winner, or winning team, gets nothing more than pride and glory. Sometimes I split the class in groups and let each group work on the same problem for 10–15 mins, say. The first team who submits a working solution gets a point. Points are then accumulated for various learning tasks during the day. The students tend to like this concept but it should not be overdone and detract from the learning process.

Discussions *Discussions* are not a scheduled task in the workshop but rather something that happens ad hoc during the tasks mentioned above. Forcing students to discuss a topic is rarely very awarding. Instead, having outsourced the traditional lectures from the classroom provides more time to allow discussions to emerge as needed. This is extremely useful for deep learning, as it helps students learning by reducing the cognitive load, delving deeper into topics, and allowing time to organise and store new material in long-term memory. In addition, discussions will often sidetrack into related topics, evident of students looking at the world from a different perspective, discovering relationships across topics, and even changing as a person.

3.3 The Online AI Course

As mentioned earlier, all students in TAI are required to enrol in an online AI course offered by edX called *CS188.1x Artificial Intelligence*. This course has been archived since 2014, which means that no teaching staff maintain or are active in the course. Nevertheless, all the resources provided are available, free of charge, to edX users. In Part A of TAI, four of the five topics are covered by the edX course, with excellent video lectures, step-by-step tutorials, and quizzes. The video lectures come both in long and unedited versions and in short and condensed edited versions that are also interspersed with self-tests with immediate

feedback, all easily accessible on portable devices such as mobiles and tablets. The voices in the videos are transcribed to text, which makes it easy for students to quickly browse through a video by scrolling through the text until a given point of interest.

Whereas some students favour watching the long videos in one go and then use the the short, edited videos for repetition, others jump straight to the edited videos, especially if they have studied the textbook first. All the edX material is readily avaiable via an edX app, which, when combined with the video format and interactive material, also make it easy for students to squeeze in short micro-sessions of homework during their day, e.g., when riding the bus to campus.

Fig. 1 shows an example of the video user interface of the edX course, where users watch very short transcribed videos in sequence, interspersed with interactive quizzes and demos.

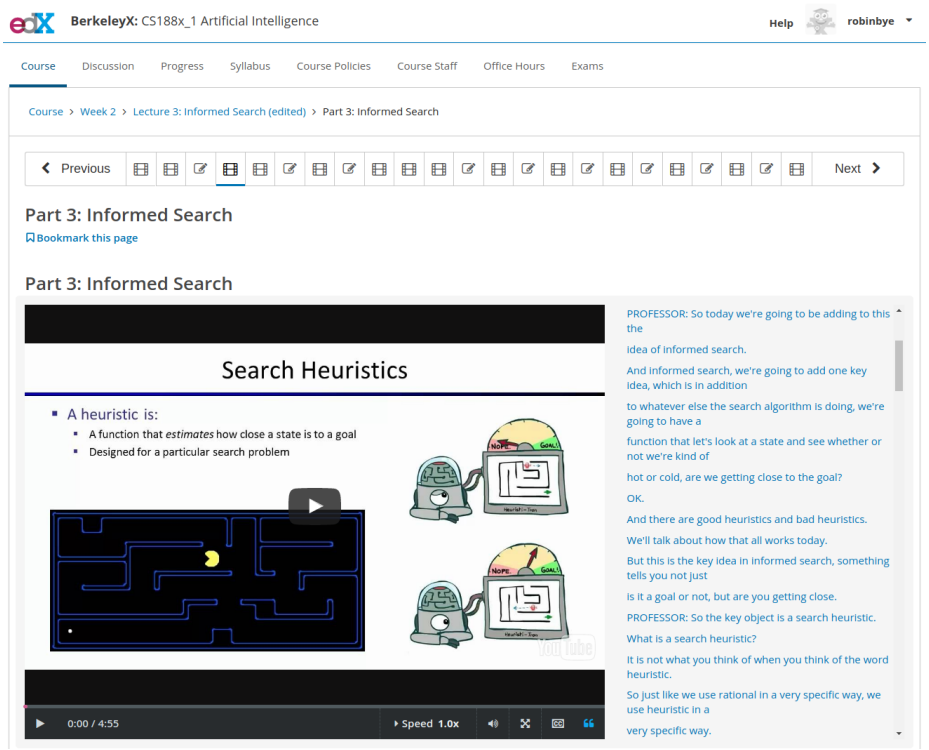

Fig. 1. Video interface for the edX course CS188.1x Artificial Intelligence.

3.4 Coding Challenges

To construct knowledge, skills, and competence within the topics of TAI, students need to practice on modelling problems with relevance to the real-world, and be able to select or modify existing intelligent algorithms from the AI literature or devise new algorithms suitable to the level of abstraction in their models. The next step is to implement the algorithms and test them in a simulated environment, and perform an analysis of their behaviour, often requiring visualisations to understand what is going on. Doing this from scratch requires both skills and a lot of time, and henceforth we have outsourced this to two code development platforms, HackerRank and CodinGame, both of which provide excellent frameworks for students to focus on coding algorithms.

HackerRank HackerRank provides numerous programming challenges in several domains and programming languages, including the domain most relevant for this course, namely AI. All challenges consist of a problem description and a code window in a web browser (see Fig. 2, top) where students can enter their programs, or they can use a plugin to enable writing their code locally in their own editor on their own computers. Programs are then tested on text input and must produce the correct text output. Sample stub code to get students started is provided in many cases.

For Part A of TAI, I cherry-pick exercises from relevant AI subdomains on HackerRank, including Graph Theory, A* Search, Bot Building, Alpha Beta Pruning, Combinatorial Search, and Games. An example exercise is BotClean Large,[8] in which students must program an *intelligent agent* (bot) to clean all the dirty cells in a 2D grid. After submitting their code, a simulation tests the bot of each student on a number of different scenarios and returns a score.

An example of input and output text formats for BotClean Large is shown in Fig. 2 (middle). For the input, the first line shows the the coordinates ($x = 0$, $y = 0$) of the cleaning bot; the second line gives the dimensions of the 2D grid (5 rows $\times$ 5 columns); and the next 5 rows shows whether a cell in the maze is dirty ('`d`') or clean ('`-`'), and whether the bot is in a particular (clean) cell ('`b`'). For the output, the AI program must iteratively in an infinite loop return one of the actions RIGHT, LEFT, UP, DOWN, or CLEAN, with the action RIGHT chosen in this particular case.

CodinGame CodinGame works in a similar manner as HackerRank but also provides game-like 2D graphical feedback of the behaviour of the code. The stub code to get students started on a particular programming challenge is usually better than the one on HackerRank. Programming challenges are divided into single-player challenges (ranked Easy, Medium, or Hard) that can be found in the Training category, and larger, multiplayer games called Bot Programming, that can be found in the Multiplayer category (past contents) or Contests (currently running contests). One example of a multiplayer game is Game of Drones, in

[8] https://www.hackerrank.com/challenges/botcleanlarge

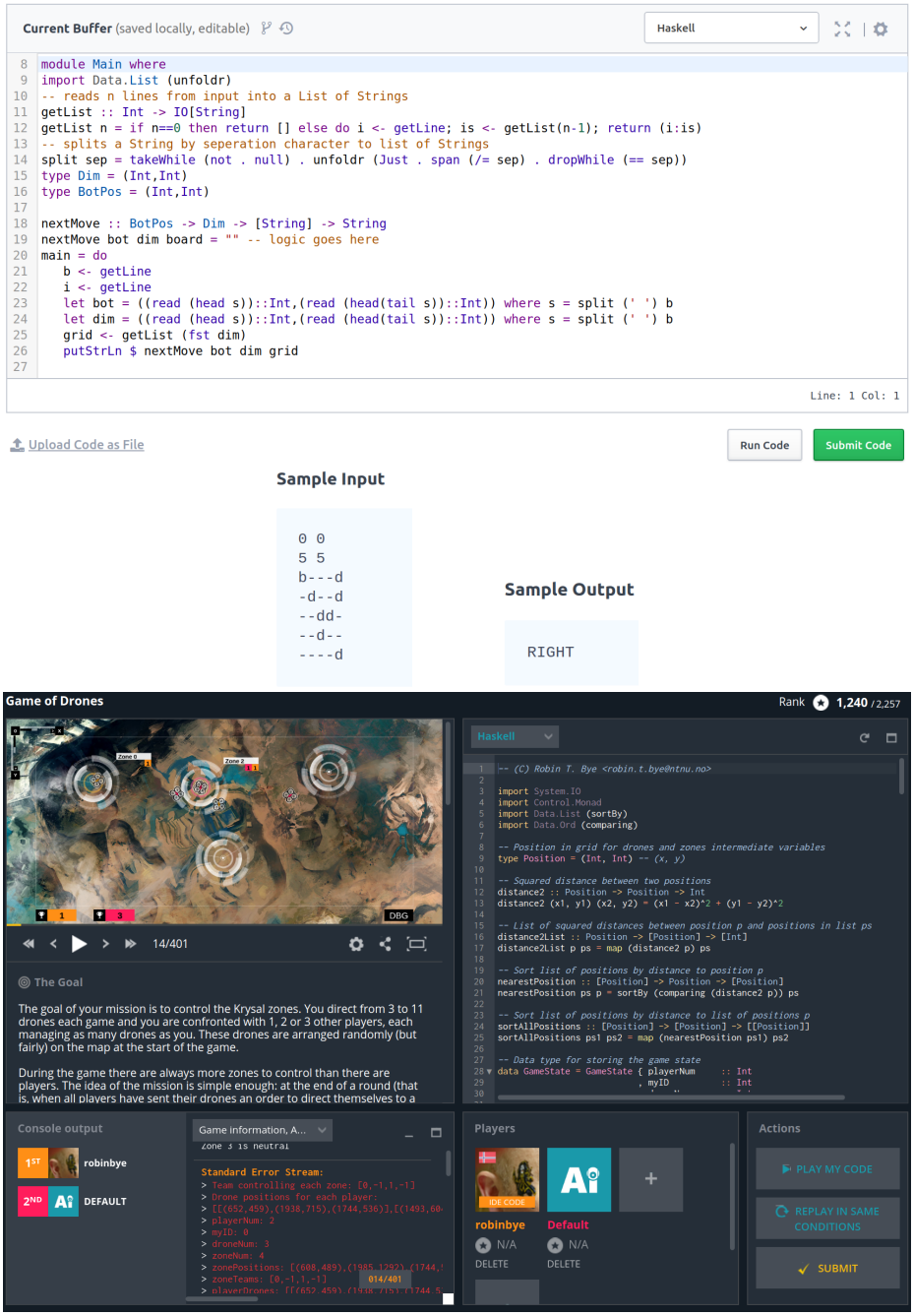

Fig. 2. HackerRank coding window (top) and sample input and output text for Bot-Clean Large challenge (middle), and example screenshot of CodinGame environment for Game of Drones challenge (bottom).

which students must control a fleet of drones and compete against other players (classmates, online players, or the default AI bot provided by CodinGame) with their own fleet of drones in a battle of controlling as many zones as possible in a large 2D grid.

The screenshot in Fig. 2 (bottom) shows an example of the CodinGame environment run in a web browser for Game of Drones. The top left window shows a graphical simulation of the game; the top right is the coding window; bottom left shows the console output; and bottom right shows which players are competing and has some buttons to for testing and submitting code.

Intelligent Agents Both platforms have been designed in a similar manner where the user must input a computer program, commonly referred to as the AI, or *bot*, that reads data from standard input, does some processing, and then outputs a desired action to standard output. This loop repeats for a number of time steps until the simulation, or *game*, ends. Thus, the internals of the simulated environment behaves much like a black box to the bot, however, certain parameters in the environment are "sensed" and provided to the bot at each time step, and in addition, each coding challenge provides details of the model the simulated environment is based upon sufficient that a useful bot can be constructed. Hence, the coding challenges adopts the same paradigm as the course textbook, in which AI is centred around *intelligent agents* that *sense*, *plan*, and *act* [34].

Coding challenges typically revolve around certain themes in computer science and AI such as queues, lists, search algorithms, path planning, control, etc. Very often, each challenge comes in several versions, Easy, Medium, and Hard, in which the challenge becomes gradually more complex and difficult to solve. An example is the Mars Lander challenge on CodinGame, where students doing the Easy level must design an intelligent agent, first for controlling the landing of a spacecraft in the vertical dimension only, then, for the Medium and Hard levels, in two dimensions, which is exceedingly harder.

Design Process When working on coding challenges, I emphasise the importance of thinking about the problem before jumping straight into coding. Together with the students, we brainstorm various potential models of the problem to be solved, and discuss aspects such as levels of abstraction, and their advantages and limitations. The solution method to be used is dependent on the model, as an intelligent algorithm requires the problem to be represented by exactly those components that the algorithm was designed for. The students then run a number of tests on the coding platform to verify their design and receive automatic feedback on which tests were passed or failed.

In contrast with HackerRank, the CodinGame platform also provides *visualizations* of the simulated environment. This is very valuable to students in the iterative process of testing, debugging, refining, and analysing their algo-

rithms, because visual feedback quickly can reveal problems in the execution of the algorithm.[9]

4 Course Evaluation

To conform with the requirement in NTNU's Quality System for Education that a *course evaluation* must submitted after every semester that a course is run, I have adopted a scheme where all students present during workshops (typically 7–8 students) belong to a so-called *reference group*. At the beginning of a workshop, we begin with a class discussion, or *status meeting*, about course-specific issues such as status, progression, problems, etc., as well as general issues about the master programme in general. These meetings, which are typically quite short (5–20 mins) provide an important arena and opportunity for students to help the teacher adjust the course early before issues become too big to fix. Once or twice a semester, I ask the students to answer a detailed list of evaluation questions before coming to class, in order to get a more in-depth discussion. During the status meeting, the teacher takes notes, which form the basis of an end-of-semester course evaluation report by the teacher. Likewise, members of the reference group are required to write a short evaluation report on behalf of the students. Both reports are submitted to the university and are available to all staff and students.

In the sections below, I highlight some of the findings from my evaluation reports for the TAI courses that was run autumn 2016[10] and spring 2017.[11] I also present results from an end-of-semester student survey for the spring 2017 cohort.

4.1 Relevance

Students in both the 2016 and 2017 cohorts praised both this specific course on AI, as well as the master programme in simulation and visualization as a whole, for being highly relevant in today's society and job market. Specifically, they mentioned how many newly started technology companies of today both in Norway and internationally seem very focussed on incorporating AI and other themes of the master programme into their business models across a variety of domains. Students also gave the names of several companies that bear resemblances with the names of courses in the master programme to illustrate their point.

4.2 Workload

The 2016 cohort thought the overall workload was appropriate, although perhaps slightly bigger than in some other courses in the master programme. Some

[9] For this reason, and other minor reasons, we have chosen to use the CodinGame platform exclusively starting from 2017.

[10] http://robinbye.com/files/reports/TAIevaluation16.pdf

[11] http://robinbye.com/files/reports/TAIevaluation17.pdf

students thought the workload for the assignments was high but thought this was justified by the assignments being highly valuable. For example, one student stated that "*(. . .) the assignments were big but good. Even though this course had a high workload, in retrospect I would say that this was necessary, and the main reason I learnt so much.*"

Among the 2017 cohort, some students thought the overall workload was high and close to the limit of cognitive overload. Others agreed but underlined that this was counterbalanced by the assignments being highly valuable. In the beginning of Part A on search algorithms, some students said this part was a little overwhelming, and that it was difficult to know what was important in the AIMA textbook and the video lectures.

The teachers think both cohorts were faced with about the same workload in both Part A and Part B, thus the differences in perceived workload seem rooted in different expectations among the cohorts and not actual differences in workload.

4.3 Teaching Methods and Learning Environment

Both the 2016 and 2017 cohorts expressed enthusiasm for the workshop format and the elements of flipped classroom but for the 2017 cohort, especially in the beginning, this style of teaching was something they were not used to. The 2016 cohort suggested that when revising theory from homework, e.g., in a micro-lecture, it would be slightly better for the teacher to have slides than having to move back and forth in the online video lecture, whereas the 2017 cohort suggested that short revision/micro lectures on selected topics should still be prepared and provided in class and not only provided *ad hoc* on requests by students. This latter approach will of course require more work from the teachers, especially since it is difficult for the teacher beforehand to know which topics from the homework that will need a thorough treatment in class. However, it was pointed out by students that there exist a website[12] with the same online edX course material and maintained by the same edX teachers that also contains slides freely available to use. The teachers should therefore obtain copies of these slides and have them ready when questions regarding the online video lectures arise in class.

Both cohorts appreciated the fact that videos were available in both short and long versions, and different students sometimes favoured one over the other. The 2017 cohort also meant that the online video lectures provided a good "compression" and focus on key topics in the AIMA textbook.

In addition, students in both cohorts expressed that the interactive material was appropriate and useful, however, some of the interactive MCQs were not always well designed and could contain questions that were ambiguously phrased and therefore one could argue which answer was really the correct one.

Students critically challenging the course material such as online quizzes is very useful for both the teacher and for fellow students, as it triggers class

[12] http://ai.berkeley.edu

discussion and deeper insight into the problem that is being studied. This is also an example of deep learning, commonly observed with skilled students, as more shallow surface learners will fail to note such ambiguities in the questions.

4.4 Assignments

Both the 2016 and 2017 cohorts thought the topics covered in assignments and the design of the problems to be solved were appropriate, although the workload was quite high. They also appreciated that assignments are first released as draft versions and then modified by the teacher and the students together through constructive class discussions.

Moreover, students were dissatisfied with the timing of deadlines of assignments across different courses run simultaneously and strongly suggested that all teachers across all courses speak together at the beginning of a semester and coordinate and spread out the deadlines for assignments more evenly.

In their student evaluation report, the 2017 cohort highlighted that the assignment feedback from the professors should be more oriented on code structure and implementation. Furthermore, after personal feedback of the assignments has been released on Fronter, they suggested that the teacher spend some time of the workshop on particular problems faced in the assignments, e.g., common errors or methodologies among the students.

The average quality of all assignments was generally very high. For Assignment 3 in the 2017 cohort, two students used Haskell and a genetic algorithm that they had learnt about in the course Functional Programming and Intelligent Algorithms run in parallel with Topics in AI. They expressed that learning functional programming and also to be able to use skills and knowledge across different courses was highly valuable.

4.5 Structure and Information

Both cohorts expressed satisfaction with the course structure and the continuous flow of information by means of Fronter but the 2017 cohort complained about the lack of compatibility and usefulness of the Fronter app for smartphones. In their student evaluation report, the 2017 cohort wrote that "[the] *course material were* [sic] *perfectly organized in the combination of the textbooks and online recourses with detailed guidance by the professor*."

Furthermore, students found little use of the Fronter discussion forum, although both cohorts acknowledged that using the forum more as a *knowledge bank* of FAQs was useful.

4.6 Coding Platforms

There was mostly agreement in both cohorts that HackerRank should not be used, mainly because it is more difficult to use and lacks visualisations, in contrast with CodinGame. Students also thoroughly enjoyed the *gamification* part

of CodinGame, where one achieves rating points and badges and therefore are continuously eager to improve.

In contrast with previous years, the 2017 cohort used the CodinGame platform exclusively. Some students said it was difficult to apply concepts from theory directly in the coding challenges. One student expressed a wish for the alternative platform HackerRank to be used because it contains many problems that are more directly related to the curriculum. Specifically, many HackerRank challenges forces the students to use a particular search algorithm to solve a problem, whereas CodinGame challenges are much more open-ended and can typically be solved by a black box, where it is entirely up to the students which algorithms or methods to put inside the black box. One student expressed concern with the heavy focus on programming in this course and in the master programme as a whole for students with little programming background.

Overall, students were generally pleased with using CodinGame and doing very practical problem-solving while having some fun. In their own evaluation report of the course, the 2017 cohort suggested that it could be advantageous to limit the freedom in solution methods to the search algorithms presented in class. Currently, students are instructed to solve selected CodinGame challenges any way they like, be it using search algorithms taught in the course or any other method. In particular, students suggested the possibility of solving much fewer coding challenges (e.g., just one) but solve it many different ways, using the search algorithms of Part A of the course. Students also emphasised that this would limit coding overhead related to implementing the necessary data structures and interfaces for several different coding challenges and "provide clearer opportunity to work with search algorithms intensively."

Contrasting the 2016 and 2017 cohorts, it seems clear that the 2017 cohort was slightly dissatisfied with the open-ended nature of the coding challenges and would like problems to be more tailored to the curriculum, possibly with more examples and step-by-step tutorials prepared by the teacher.

4.7 Alignment of Part A and Part B

As described previously, Part A and Part B of the course is taught by two different teachers. Especially the 2017 cohort expressed that the linkage between the two parts is too weak, with some students suggesting that the two parts co-exist almost as two smaller and separate courses that should be taught independently.

4.8 State-of-the-art in AI

In their student evaluation report, the 2017 cohort suggested that it "*could be reasonable to have* [an] *optional lecture describing the state-of-the art solution in the AI.*" Moreover, they suggested that the proposed lecture could be student-organized, possibly with research results presentations.

4.9 Summary of Feedback

Throughout the course and in the weekly status meetings involving the entire class, students of both the 2016 and 2017 cohorts had very little negative concern about the course itself, be it teaching methods, workload, or curriculum. When they had concerns, this was mainly related to clashes of assignment deadlines across courses and suggestions to reduce workload or modify content of assignments. Interestingly, students were very eager to discuss issues of other courses and the master programme as a whole. We see this as a sign that this course provides a safe and relaxing learning environment, since students felt comfortable and very keen to share these details with the teachers.

Finally, almost all students told us that they were highly satisfied with the course, with two students expressing in writing that "*I also wish to express that this has been one of the best courses I have taken in higher education*" and "[t]*hank you for teaching one of the better courses if not the best course I've had in this master*[']*s program*[me]."

5 Student Evaluation Survey

The 11 students enrolled in the spring 2017 cohort were asked to complete an anonymous student evaluation survey about Part A of the coures online, of which 8 responded (6 full-time students, 2 part-time students). The students were first asked to specify whether they had attended most classes or not, choosing one of the following three alternatives (answers as indicated):

1. Attended most classes: 7 students
2. Did not attend most classes due to work, children, or other reasons unrelated to the course: 0 students
3. Did not attend most classes due to own priorities/lack of usefulness: 1 student

Then, for following set of 25 statements given by Table 1, the students were asked to indicate to which degree they agreed with the statements, categorising whether they strongly or partly agreed or disagreed, or were indifferent.

Table 1: Statements for student evaluation survey of Part A.

#	statement
1	*I want more traditional lectures (typically slideshows)*
2	*I want more lectures where the teacher uses the blackboard*
3	*I want more active learning activities (e.g., exercises, discussion, quizzes, competitions, student presentations, practical coding assignments, etc.)*
4	*I want more flipped classroom and e-learning/web-based learning activities*
5	*I want more emphasis on practical application than theory*
6	*I want more problem-solving learning activities*
7	*I want more computer lab activities (coding exercises in the classroom)*

8 *The teacher calling the roll makes it more likely that I turn up to class*
9 *I want more compulsory assignments*
10 *I want more/better personal feedback on my progress*
11 *I want my final grade to be decided by an single final oral exam*
12 *I want my final grade to be decided by an single final written*
13 *I want my final grade to be composed of several grade components (e.g., computer labs, assignments, projects, mid-semester test, final exam)*
14 *I want digital exams to be employed*
15 *I want home exams to be employed*
16 *The workload was appropriate*
17 *The learning activities were appropriately aligned with the intended learning outcomes, course curriculum, and assessment*
18 *The teacher's role in this course as a facilitator/guide (e.g., cherry-picking suitable online material/resources) where I must do the hard work myself is a good model for learning*
19 *The grade I received on the final oral exam reflects my knowledge/skills/general competence in the course*
20 *This course is relevant in today's society and for this master programme*
21 *The structure, information, and quality of course material/resources were appropriate*
22 *Coding assignments should be less open-ended and more tailored to applying the search algorithms taught in the course*
23 *The course should have at least one overview lecture of current state-of-the-art*
24 *The course is suitable for off-campus study*
25 *Attending workshops was valuable for my learning*

In addition, they were given the opportunity to elaborate on the statements and any other issues they wished to raise.

The results are summarised in Table 2, where the number n of student responses and the corresponding percentage is given for each statement and response category.

5.1 Analysis of Survey

A thorough analysis of this survey is outside the scope of this paper. The following sections discusses briefly the most relevant findings.

Flipped Classroom Approach Statements 1–7 shows that the 2017 cohort appears somewhat split regarding the flipped classroom approach. Half the students (4) would like more traditional lectures (statement 1), whilst the other half is indifferent (1) or partly disagrees (3). On the other hand, regarding the following remedies in statements 2–7, which support (statements 3–7), or are not in conflict with (statement 2, using the blackboard), a flipped classroom approach, half or more wants more use of the respective approaches. Furthermore,

Table 2. Results from student evaluation survey of Part A.

	strongly agree		partly agree		indifferent		partly disagree		strongly agree	
Statement	n	%	n	%	n	%	n	%	n	%
1	1	12.5%	3	37.5%	1	12.5%	3	37.5%	0	0.0%
2	2	25.0%	3	37.5%	0	0.0%	3	37.5%	0	0.0%
3	2	25.0%	2	25.0%	1	12.5%	3	37.5%	0	0.0%
4	1	12.5%	1	12.5%	3	37.5%	2	25.0%	1	12.5%
5	4	50.0%	1	12.5%	3	37.5%	0	0.0%	0	0.0%
6	2	25.0%	6	75.0%	0	0.0%	0	0.0%	0	0.0%
7	2	25.0%	3	37.5%	0	0.0%	0	0.0%	3	37.5%
8	2	25.0%	1	12.5%	5	62.5%	0	0.0%	0	0.0%
9	0	0.0%	2	25.0%	1	12.5%	4	50.0%	1	12.5%
10	1	12.5%	2	25.0%	1	12.5%	2	25.0%	2	25.0%
11	1	12.5%	1	12.5%	2	25.0%	1	12.5%	3	37.5%
12	0	0.0%	1	12.5%	2	25.0%	3	37.5%	2	25.0%
13	4	50.0%	2	25.0%	1	12.5%	0	0.0%	1	12.5%
14	2	25.0%	1	12.5%	4	50.0%	0	0.0%	1	12.5%
15	0	0.0%	1	12.5%	3	37.5%	1	12.5%	3	37.5%
16	3	37.5%	2	25.0%	0	0.0%	3	37.5%	0	0.0%
17	1	12.5%	4	50.0%	2	25.0%	1	12.5%	0	0.0%
18	2	25.0%	2	25.0%	3	37.5%	1	12.5%	0	0.0%
19	4	50.0%	1	12.5%	0	0.0%	3	37.5%	0	0.0%
20	5	62.5%	0	0.0%	3	37.5%	0	0.0%	0	0.0%
21	4	50.0%	3	37.5%	1	12.5%	0	0.0%	0	0.0%
22	3	37.5%	3	37.5%	0	0.0%	1	12.5%	1	12.5%
23	2	25.0%	5	62.5%	1	12.5%	0	0.0%	0	0.0%
24	1	12.5%	3	37.5%	4	50.0%	0	0.0%	0	0.0%
25	1	12.5%	6	75.0%	0	0.0%	1	12.5%	0	0.0%

4 students agree and 2 students are indifferent to the teacher adopting a role as a facilitator for learning (statement 18), whilst the entire cohort agree (4) or are indifferent (4) to the course being suitable for off-campus study (statement 24).

Assignments, Feedback, Workload Statement 9 shows that 5 of the students disagrees with having more compulsory assignments, whilst 1 is indifferent and 2 partly agree. An encouraging finding is that only 3 students want more/better personal feedback, whilst 4 disagrees. Finally, 6 students agree (2 disagree) that coding assignments should be less open-ended (statement 22), whilst the majority (5 agree, 3 disagree) that the course workload was appropriate (statement 16).

Exam Format From statements 11–15, it appears the majority of students are not in favour of neither a single final oral or written exam (statements 11–12) deciding their final course grade, with 6 students being in favour of a composite

grade consisting of several components (statement 13, e.g., lab work, assignments, projects, mid-semester test, final exam). This finding is supportive of the hypothesis of Biggs [e.g., 5–7] and proponents of CA, who claim that students are strategic and mainly concernced about grades, therefore one should incorporate graded assessment tasks that are well aligned with ILOs. Moreover, the majority of students are in favour or indifferent to using more digital exams (statement 14), whilst they are against or indifferent to home exams (statement 15).

General Satisfaction In general, the students appear satisfied with the course. All but one student agree or are indifferent to the learning activities being appropriately aligned with ILOs, curriculum, and assessment (statement 17). 5 students agree and 3 are indifferent to the course being relevant in today's society (statement 20). 7 students agree (1 partly disagrees) that attending workshops was valuable for their learning (statement 25).

6 Final Oral Exam Results

The grade distributions for the 2016 and 2017 cohorts are shown in Table 6 below: These are quite exceptional results, but not surprising based on the effort,

Table 3. Results from final oral exam.

Grade	A	B	C	D	E	F	Absent
2016	5	2	-	-	-	-	-
2017	7	1	3	-	-	-	-

interest, and enthusiasm experienced in class. The quality of the assignments handed in by students in both cohorts were very good or excellent and being closely aligned with the course curriculum have likely served as highly appropriate preparation for the exam.

Notably, from Table 2, 5 students strongly agree with the grade they got, whereas 3 students partly disagree.

7 Discussion

In this paper I have presented a flipped classroom approach for teaching a master's course on AI. Whilst it is my clear impression that the course has been a great success, the reader should by no means believe that we have found the holy grail for teaching higher education courses, as reflected from the student evaluation survey and the evaluation reports by students and teacher presented above. Indeed, there are several limiting frame factors and challenges that must be overcome for flipped classroom to work as intended. Below I discuss some of these.

7.1 Teaching Methods and Learning Environment

This course, especially Part B on search algorithms, strongly emphasises active learning activities by outsourcing most of the theoretical foundation to online video lectures, textbooks, and scientific papers. The theory along with practical online interactive exercises such as MCQs, quizzes, and other forms of self-assessment are effectively outsourced as homework to be done before students arrive in class.

Whilst enforcing students to study the textbook and show up well prepared to class is not a new idea, my impression is that a flipped classroom strategy employing online video lectures and the interactive material increases the will of students to actually do their homework. For example, the video lectures in this course come both in long unedited versions and in short and condensed edited versions, all easily accessible on portable devices such as mobiles and tablets. The videos are also speech-to-text-annotated, which makes it easy for students to quickly browse through a video to a point of interest. The video format and interactive material also makes it easy for students to squeeze in short micro-sessions of homework, e.g., during a 15-min bus drive. When students face problems in the theory from their homework, the teacher can give a short micro-lecture on the topic.

By outsourcing traditional passive learning activities such as lectures, the time in the classroom during workshops (full days of teaching/learning activities) can be much more efficiently used for active learning activities, such as problem-solving, quizzes, informal competitions, exercises, discussion, and assignments and projects. Being a relatively small class, it is easy for the teacher to provide individual help to each student if necessary.

Importantly, the teacher is not necessarily present in the classroom the whole day, which have several advantages. For example, some students tend to be better at socialising and discussing topics when the teacher is not present, e.g., when working on a practical problem. In addition, the teacher can prepare topics or answers to questions raised in class whilst the students are working independently without supervision.

During workshops, students were active and eager to discuss topics but sometimes needed a bit of a push to get started on being active. Having outsourced valuable time for lectures also meant that the teacher has less pressure to avoid discussions or delving into sidetrack topics just because there was a pressing schedule to adhere to.

7.2 Frame Factors

The *number of students* in a class can be a limiting *frame factor* for achieving ILOs. In the course presented here, the 2015 and 2016 cohorts consisted of about 10 students, of which 2–3 quit early or never turned up, whereas 15 students were enrolled in 2017, of which 11 ended up doing the course. If the class had exceeded about 25 students, say (which is also the number of seats funded by the Ministry of Education), it may have been more difficult to achieve the same fruitful

learning environment in the classroom. For larger classes, one would likely have to abandon flat classrooms and use large auditoriums. University management would typically see no problem in this and tell teachers to go ahead and do their traditional one-way lectures as usual, which is not desirable. Optionally, one could split such large cohorts into smaller groups, which would multiply the number of teachers required and associated costs. This begs the question why there is a transition in teaching methodology from classroom-style teaching (one teacher per class of 30 students, say) in lower education to auditorium-style lecture-based teaching in higher education, thus turning students into passive learners. Flipped classroom can still be useful for large classes, however, but may require much more effort in facilitating the in-class learning activities.

Another important frame factor is the *available online resources*. In TAI, we have been lucky to find both an edX course that is closely aligned with the topics in the course textbook for Part A, as well as the CodinGame platform that provides a high quality framework with a vast variety of AI coding challenges suitable for learning key aspects of the course, such a modelling problems, implementing intelligent algorithms, and perform testing and analysis.

Producing these online resources oneself with the same quality, as well as the material contained in textbooks, is an impossible task for the common teacher. Still, with *e-learning* as a buzz word nowadays, university management are very eager for teaching staff to make videos to put online. I think this approach is flawed, as it is extremely time-consuming to produce video lectures oneself of the desired quality, not to mention developing a suitable simulation environment for testing algorithms. Instead, teachers should act as *facilitators* and devote their time to finding such resources and tying them together in a didactically sound manner. For example, the teacher must take care in selecting the right coding challenges and guiding the students during problem-solving (see the next section).

7.3 Problem-Based versus Problem-Solving Learning

Problem-based learning is a hot topic of higher education and the interested reader may refer to our accompanying paper for reflections on many of its aspects [30]. Here, I refrain myself to a very important finding by Hattie and Goveia [22], who upon examining 800 meta-analyses note that problem-based learning cannot be shown to have a positive effect on achieving ILOs! The reason for this, according to Sotto [38], is the dinstinction between *problem-based* and *problem-solving* learning. He argues that the problems handed to students must be carefully designed by the teacher so that students easily can progress and be able to practice and achieve the ILOs. Specifically, one should use a teaching approach that employs well-designed case studies and avoid problem-based and too student-centred learning, especially for larger problems and when there is no clear guidance towards how to solve them. Otherwise, students will get stuck or spend too much time on finding the necessary pieces to solve the puzzle by searching the Internet or studying textbooks.

Adopting this distinction of Sotto [38], I believe the focus on *problem-solving learning* is a major key to the success of this course. For example, the coding challenges on CodinGame are carefully selected and much attention is given during workshops on how to model the problems and how to select appropriate solution methods. Yet, there is still room for much improvement, since some students are dissatisfied with the selection of problems and would like problems to be less open-ended.

However, care must be taken to strike the *right balance*. If problems are too rigidly defined, with perhaps only one solution approach possible, as when instructing students to follow detailed step-by-step instructions, there is a danger that creativity is neglected and students only achieve surface learning [3]. Comparing with bachelor's courses, where such rigid schemes to some extent can even be desirable, master's courses should have a higher emphasis on being research-based and investigative in nature.

Hence, in TAI, we are still striving to find the right tradeoff between coding challenges being too rigid versus too open-ended. There are nearly always several paths for solving a given problem, and the teacher should often provide a simple outline of a solution as a starting point and then guide students towards improving the solution.

7.4 C-4 Dynamite for Learning

Central to my teaching approach is an emphasis on four axes of learning, **C-4**, that together are dynamite[13] for learning: *creativity*, *cooperation*, *competetition*, and *challenge*.

First, the learning activities offered in the course should facilitate *creativity*. In a creative process, students model and design solutions to a wide variety of problems but are usually not restricted (much) on how to model the problem and which method to use.

Second, many *cooperative learning* activities are employed in class and students are encouraged to cooperate with each other also for individual work.

Third, as mentioned previously in Section 3.2, I sometimes run small, informal *competitions* in class as a driver for motivation. In addition, the CodinGame platform is inherently competitive, as users get rating points as they improve their solutions or solve more problems. Students therefore compete not only against others but also against themselves in a quest for better ratings.

Fourth, students should be encouraged to critically *challenge* all the concepts they are introduced to in the course and I have indeed found that students tend to do this more as the course progresses. Triggers for the students can be as minor as mistakes in the written formulations in quizzes, as discussed in Section 3.2, or discovering that a solution method has certain limitations that they first found out about when implementing it on CodinGame. For example, a given intelligent algorithm may not be able to provide a good answer in reasonable time and a particular CodinGame test might therefore fail.

[13] C-4 is also a common plastic explosive.

Achieving all the C-4 components in a course is an ambitious task. Still, with the increasing availability of high-quality online resources in many fields of research and education, a flipped classroom approach to teaching clearly eases the burden of making this possible.

7.5 Future Work

Based partly on the work presented in this paper, a decision has been made to separate the topics of Part A and Part B of this course. Beginning 2018, Part A and Part B will form the basis of two new courses called *Artificial Intelligence* and *Machine Learning*, respectively. We will also address the issue of colliding assignment deadlines across courses run simultaneously during the same semester. Moreover, in line with theory of CA, it would be interesting to follow the students' advice and change the grading format from a single final oral exam to a composite grade, with assessment task distributed across the semester. Finally, at our department, we have begun to run educational seminars and roundtable discussions where teachers present their approaches to teaching and this course has been presented to both internal and external colleagues as a motivator for implementing flipped classroom themselves.

7.6 Conclusions

I have shown in this paper one way out of many that flipped classroom can be implemented in a master's course on AI. The key to success is to be able to facilitate learning by cherry-picking textbook chapters and relevant literature as well as suitable online resources such as video lectures, self-tests, and coding challenges. When such resources do not exist or lack in quality or theory, one can try to make one's own material, e.g., a compendium supplementary to the textbook, or one's own video lectures. However, this is a very time-consuming process if one wants to achieve the desired quality, and should be a last resort.

Whereas I favour a teaching approach that has much in common with CA, I emphasise that the right balance between too rigidly defined learning activities where students are being "spoonfed" and too vaguely defined problem-based learning activities must be found, so that the problems are investigative in nature with different possible paths towards solutions, yet at least one path must be reasonably easy for the students to find and follow. Compared to courses at the bachelor level, students of a master course like TAI must be trained in investigative and research-based methods, with higher demands on self-exploration. Offering a range of problems, where introductory step-by-step tutorials are used to introduce new material, and harder, more open-ended problems are presented subsequently, is likely a good approach.

Acknowledgements

The Software and Intelligent Control (SoftICE) Laboratory[14] is grateful for the financial support given by the Study Committee at NTNU in Ålesund through the educational research project *Research-Based and Innovation-Driven Learning (FILA)*, grant no. 70440500.

[14] http://softicelab.wordpress.com

Bibliography

1. Abeysekera, L., Dawson, P.: Motivation and cognitive load in the flipped classroom: definition, rationale and a call for research. Higher Education Research & Development 34(1), 1–14 (2015)
2. Abu-Mostafa, Y.S., Magdon-Ismail, M., Lin, H.T.: Learning From Data: A Short Course. AMLBook (Mar 2012)
3. Andersen, H.L.: "Constructive alignment" og risikoen for en forsimplende universitetspædagogik. Dansk Universitetspædagogisk Tidsskrift 5(9) (2010)
4. Anderson, J.R.: Cognitive Psychology and Its Implications. Worth Publishers, 8th edn. (2015)
5. Biggs, J., Tang, C.: Teaching for Quality Learning at University. McGraw Hill/Open University Press, 4th edn. (2011)
6. Biggs, J.: Enhancing teaching through constructive alignment. Higher Education 32, 347–364 (1996)
7. Biggs, J.: Aligning teaching for constructing learning. The Higher Education Academy, York, United Kingdom (2011), accessed online on 27.09.2011 at http://www.heacademy.ac.uk/assets/documents/resources/resourcedatabase/id477_aligning_teaching_for_constructing_learning.pdf
8. Bishop, J.L., Verleger, M.A.: The flipped classroom: A survey of the research. In: ASEE National Conference Proceedings, Atlanta, GA. vol. 30 (2013)
9. Bligh, D.A.: What's the Use of Lectures? Intellect Books (1998)
10. Bonwell, C.C., Eison, J.A.: Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports. ERIC (1991)
11. Borkowski, J., Thorpe, P.: Self-regulation and motivation: A life-span perspective on under- achievement. In: Schunk, D., Zimmermann, B. (eds.) Self-regulation of learning and performance: Issues of educational applications, pp. 44–73. Hillsdale, NJ: Erlbaum (1994)
12. Bowen, C.W.: A Quantitative Literature Review of Cooperative Learning Effects on High School and College Chemistry Achievement. Journal of Chemical Education 77(1), 116 (2000)
13. Bye, R.T.: The teacher as a facilitator for learning: Flipped classroom in a master's course on artificial intelligence. In: Proceedings of the 9th International Conference on Computer Supported Education — Volume 1: CSEDU (CSEDU '17). pp. 184–195. INSTICC, SCITEPRESS (Apr 2017), selected for extended publication in Springer book series Communications in Computer and Information Science (CCIS)
14. Clark, R.C., Nguyen, F., Sweller, J.: Efficiency in learning: Evidence-based guidelines to manage cognitive load. John Wiley & Sons (2011)
15. Entwistle, N., Ramsden, P.: Understanding student learning. Beckenham: Croom Helm (1983)
16. Felder, R.M., Brent, R.: Understanding student differences. Journal of Engineering Education 94(1), 57–72 (2005)
17. Foldnes, N.: The flipped classroom and cooperative learning: Evidence from a randomised experiment. Active Learning in Higher Education 17(1), 39–49 (2016)
18. Freeman, S., Eddy, S.L., McDonough, M., Smith, M.K., Okoroafor, N., Jordt, H., Wenderoth, M.P.: Active learning increases student performance in science, en-

gineering, and mathematics. Proceedings of the National Academy of Sciences 111(23), 8410–8415 (2014)
19. Gynnild, V.: Læringsorientert eller eksamensfokusert? Nærstudier av pedagogisk utviklingsarbeid i sivilingeniørstudiet. Ph.D. thesis, NTNU (2001)
20. Gynnild, V., Holstad, A., Myrhaug, D.: Teaching as coaching: A case study of awareness and learning in engineering education. International Journal of Science Education 29(1), 1–17 (2007)
21. Gynnild, V., Holstad, A., Myrhaug, D.: Identifying and promoting self-regulated learning in higher education: roles and responsibilities of student tutors. Mentoring & Tutoring: Partnership in Learning 16(2), 147–161 (2008)
22. Hattie, J., Goveia, I.C.: Synlig læring: et sammendrag av mer enn 800 metaanalyser av skoleprestasjoner. Cappelen Damm akademisk (2013)
23. Johnson, D., R., J., Smith, K.: Active Learning: Cooperation in the College Classroom. Interaction Book Co., Edina, MN, 2nd ed. edn. (1998)
24. Lage, M.J., Platt, G.J., Treglia, M.: Inverting the classroom: A gateway to creating an inclusive learning environment. The Journal of Economic Education 31(1), 30–43 (2000)
25. Lan, W.: The effects of self-monitoring on students' course performance, use of learning strategies, attitude, self-judgment ability, and knowledge representation. Journal of Experimental Education 64(2), 101–116 (1996)
26. Marshall, D., Summers, M., Woolnough, B.: Students' conceptions of learning in an engineering context. Higher Education 38(3), 291–309 (1999)
27. Marton, F.: Phenomenography — Describing conceptions of the world around us. Instructional Science 10, 177–200 (1981)
28. Marton, F., Booth, S.: Learning and Awareness. Mahwaw, NJ: Lawrence Erlbaum (1997)
29. Negnevitsky, M.: Artificial Intelligence: A Guide to Intelligent Systems. Addison Wesley, 2nd edn. (2005)
30. Osen, O.L., Bye, R.T.: Reflections on teaching electrical and computer engineering courses at the bachelor level. In: Proceedings of the 9th International Conference on Computer Supported Education — Volume 2: CSEDU (CSEDU '17). pp. 57–68. INSTICC, SCITEPRESS (Apr 2017), selected for extended publication in Springer book series Communications in Computer and Information Science (CCIS)
31. Piaget, J.: Six psychological studies. Trans. A. Tenzer. (1968)
32. Prince, M.J.: Does active learning work? A review of the research. Journal of Engineering Education 93(3), 223–231 (2004)
33. Prosser, M., Trigwell, K.: Understanding learning and teaching: The experience in higher education. Buckingham: Society for Research in Higher Education and/Open University Press (1999)
34. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson: Upper Saddle River, New Jersey, 3rd (international) edn. (2010)
35. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach: Pearson New International Edition. Pearson: Upper Saddle River, New Jersey, 3rd edn. (2013)
36. Schaathun, H.G., Schaathun, W.A.: Learning mathematics through classroom interaction. In: The 18th SEFI Mathematics Working Group seminar on Mathematics in Engineering Education. pp. 155–161 (2016)
37. Schroeder, C., Scott, T.P., Tolson, H., Huang, T.Y., Lee, Y.H.: A Meta-Analysis of National Research: Effects of Teaching Strategies on Student Achievement in Science in the United States. Journal of Research in Science Teaching 44(10), 1436–1460 (2007)

38. Sotto, E.: When teaching becomes learning: A theory and practice of teaching. Bloomsbury Publishing (2007)
39. Springer, L., Stanne, M., Donovan, S.: Effects of Small- Group Learning on Undergraduates in Science, Mathematics, Engineer- ing and Technology: A Meta-Analysis. Review of Educational Research 69(1), 21–52 (1999)
40. Topping, K., Ehly, S.: Peer-assisted learning. Routledge (1998)
41. Topping, K.J.: The effectiveness of peer tutoring in further and higher education: A typology and review of the literature. Higher education 32(3), 321–345 (1996)
42. Vygotsky, L.: Mind in society. London: Harvard University Press (1978)