The Shortest Path to Ethics in AI: An Integrated Assignment Where Human Concerns Guide Technical Decisions

Noelle Brown
University of Utah
Salt Lake City, Utah, USA
noelle.brown@utah.edu

Koriann South
University of Utah
Salt Lake City, Utah, USA
korianns@ymail.com

Eliane S. Wiese
University of Utah
Salt Lake City, Utah, USA
eliane.wiese@utah.edu

ABSTRACT

How can we teach AI students to use human concerns to guide their technical decisions? We created an AI assignment with a human context, asking students to find the safest path rather than the shortest path. This integrated assignment evaluated 120 students' understanding of the limitations and assumptions of standard graph search algorithms, and required students to consider human impacts to propose appropriate modifications. Since the assignment focused on algorithm selection and modification, it provided the instructor with a different perspective on student understanding (compared with questions on algorithm execution). Specifically, many students: tried to solve a bottleneck problem with algorithms designed for accumulation problems, did not distinguish between calculations that could be done during the incremental construction of a path versus ones that required knowledge of the full path, and, when proposing modifications to a standard algorithm, did not present the full technical details necessary to implement their ideas. We created rubrics to analyze students' responses. Our rubrics cover three dimensions: technical AI knowledge, consideration of human factors, and the integration of technical decisions as they align with the human context. These rubrics demonstrate how students' skills can vary along each dimension, and also provide a template for scoring integrated assignments for other CS topics. Overall, this work demonstrates how to integrate human concerns with technical content in a way that deepens technical rigor and supports instructor pedagogical content knowledge.

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

ethics, artificial intelligence, computing education

ACM Reference Format:
Noelle Brown, Koriann South, and Eliane S. Wiese. 2022. The Shortest Path to Ethics in AI: An Integrated Assignment Where Human Concerns Guide Technical Decisions. In *Proceedings of the 2022 ACM Conference on International Computing Education Research V.1 (ICER 2022), August 7–10, 2022, Lugano and Virtual Event, Switzerland*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3501385.3543978

1 INTRODUCTION

While the importance of teaching ethics within computing contexts is clear [1, 17, 22], how to do it is not. Current methods for integrating ethics within technical courses often involve discussing ethical issues related to technology. Consider an activity where students discuss the harmfulness of "fake news" and write an essay arguing how a social media company might suppress it, and if the company is obligated to do so [13]. This activity engages students with ethical thinking in a socio-technical context, primarily focused on the high-level goal of determining what kind of technology should be developed. Another way to connect ethical instruction with technical skills is to focus on lower-level implementation choices, and how those technical decisions respond to a human context.

We present an ethically-integrated homework assignment where students evaluate the suitability of standard algorithms for a particular human context, and propose modifications if needed. Students' responses demonstrate gaps in their technical knowledge (e.g., in defining a cost function) and in reasoning about technical choices in human contexts (e.g., determining if their chosen algorithm achieves their stated goal). Further, we also propose a rubric that can help other instructors evaluate integrated assignments. Through this work, we aim to answer the following research question: **How can an assignment simultaneously assess technical knowledge and judgements in response to human contexts?**

To answer this research question, we use our rubric, which has three dimensions: technical AI knowledge, consideration of the human factors, and how the technical decisions responded to the human considerations. Analyzing students' responses with the rubric shows that students can demonstrate different levels of knowledge along the technical and human dimensions, and also that a high score on those two dimensions independently does not always mean that students are integrating them appropriately. Further, we explore whether ethically-integrated technical assignments help instructors develop pedagogical content knowledge and improve their instruction. To examine this, our partner instructor reflected on how the assignment helped him assess students' technical understanding, and the changes he made to his instruction in response.

2 RELATED WORK

For decades, educators and researchers have explored how to teach ethics as a part of the computing curriculum [1, 22]. Many educators have successfully integrated ethics within their courses [3, 6–8, 14, 30, 31], even though widespread discussion of this topic in Machine Learning (ML) and AI education is a fairly new trend [31]. However,

ICER 2022, August 7–10, 2022, Lugano and Virtual Event, Switzerland

ACM ISBN 978-1-4503-9194-8/22/08…$15.00
https://doi.org/10.1145/3501385.3543978

there are many differing opinions on the most effective way to teach this important dimension of computing and even less clarity on how to assess it [7]. We acknowledge that “ethics” is a large field and we do not claim to address its entirety. Further, we recognize the limitations of teaching ethics in CS courses: course assignments do not encompass all of the difficulties that occur in real world settings [29], they do not address the inequalities that arise from technology [33], and do not support regulation to protect the public from harmful technologies [24].

2.1 Current Landscape of Teaching Ethics in Technical Courses

One of the most prominent debates in CS ethics education is whether ethics should be taught as its own standalone course or be taught *in situ*, integrated throughout the CS curriculum [25]. Standalone ethics courses allow for more time spent on ethics topics as the focus of the course [11]. However, this method may communicate to students that ethical issues are inherently separated from technical ones. Integrating ethics *in situ* and addressing ethical topics as they arise within a technical course has benefits including allowing students to see the technical choices they make have human consequences [31].

We argue that a more important difference than *in situ* vs. standalone is the difference between situating ethical decisions in a technical context vs. using ethical considerations to guide technical implementations. Even though there has been a large push towards *in situ* ethics education, many of the assignments and activities that teach ethics do not require students to use the technical knowledge they are learning in class. For example, students may read an article about algorithmic discrimination and be required to discuss the ethical implications [13] or discuss ethical trade-offs and decisions related to an example from science fiction [3, 16]. While these are valid ways to teach ethics as it relates to technology, these methods do not teach students to notice how their routine technical calculations and decisions can impact others in unforeseen ways. Even when technical problems are abstracted for CS classrooms, Lin [19] argues that their applications inherently embody values with political consequences. For example, the particular ordering of a search algorithm priority queue can reinforce power structures or fail to consider the needs of diverse users when applied to content moderation or road navigation [19]. Because technical components are often taught separately from the ethical issues, many instructors are hesitant to teach ethical topics in class for fear they will sacrifice valuable classroom time spent covering necessary technical concepts. Additionally, many AI instructors do not feel qualified to teach these topics [12] and there are arguments that they should not be expected to teach both technical topics and ethical frameworks [29]. To address these issues, our assignment connects human concerns with the concrete technical choices that are within the scope of the existing course content.

2.2 Teaching Ethics by Integrating Human Concerns into Technical Assignments

We recognize that "ethics" is a large field and there are different definitions of what it means to teach “ethics” within AI and CS courses. To that extent, we do not claim that our work encompasses all of ethics in AI. We argue that a prerequisite for any ethical decision is to value human concerns. An early code of ethics proposed for AI was Asimov’s 3 laws, which is predicated on a concern for human welfare (made explicit in the 0th and 1st laws, on which the others depend) [2]. Taking human concerns into account is not ethics in its entirety, but there is no ethics without it. Therefore, we see human concerns as a natural starting point. In an integrated assignment, a correct answer will involve technical choices that are influenced by human concerns. That is, changing the context or de-contextualizing the problem entirely would result in different technical choices. This criteria distinguishes between an integrated assignment and a purely technical one.

There are many quality examples of successful contextualization of technical CS concepts that require thinking about human factors (e.g., [6, 9, 21, 27]). Fiesler et al. revised technical homework assignments for an introductory CS course by situating them in ethical contexts (e.g, reading in a data file representing prospective students, creating a composite score by applying weights to numeric elements, and recommending admission based on the composite score) [7]. Follow-up questions and in-class discussions afterwards focused on the the harms of algorithmic decision-making. Student engagement was measured through self-reported perceptions of learning on these assignments compared to their other assignments. Students enjoyed the assignments, especially because they made connections between technical concepts and the real world. Contextualized assignments may promote learning by supporting these connections to personal interests and values [26], which is particularly important for broadening participating in computing [15, 32].

While Fiesler et al. [7] exemplify situating technical assignments in ethical contexts, our approach focuses on guiding technical decisions in response to human concerns. Since the Fiesler et al. [7] assignments include the specifications for the algorithms, the coding component requires only technical skills. While students discuss the implications of algorithmic decisions, they are not creating different algorithms based on their understanding of the ethical context. Therefore, while the technical aspects provide an important foundation for understanding how algorithmic decisions can be made, the ethical decisions are still separate from the technical ones. In contrast, our assignment asks students to operationalize a human goal, and select/modify an algorithm to achieve it. Further, while the ethical components of the Fiesler et al. [7] assignments were not graded, our assignment was designed to be graded and our rubrics demonstrate how to evaluate student responses.

3 THEORETICAL FRAMEWORK

What knowledge and skills do we want students to transfer from their classroom practice to real-world contexts? Many instructional approaches for ethics in CS focus on ethical reasoning about technology, such as deciding what kinds of technology should be built, or reasoning about the harms that a technology can cause. These skills may help students reason about ethical issues in a variety of technical contexts, but will likely not help students make specific implementation choices that rely on technical domain knowledge. Our approach focuses on the big idea that human considerations can and should inform all levels of technical decisions. While this overarching principle can apply to different CS domains, we contextualize it in a specific topic to support implementation choices within that domain.

We use the Knowledge-Learning-Instruction (KLI) framework to define a series of knowledge components (KCs) that represent our learning goals [18]. In the KLI framework, knowledge is represented as an action the student takes, and as a set of conditions where that action applies [18]. KCs can be isolated (representing one piece of knowledge), or integrated (representing connections among lower-level KCs) [18]. In the KLI framework, KCs are the units of transfer: if a new problem requires the KC, and the learner has acquired the action and the conditions for that action, the learner will use the KC appropriately in the new problem [18]. An instructional designer may hypothesize that a specific skill is one KC. If students do not improve after repeated practice and feedback, a possibility is that the target skill requires multiple KCs, and some are not being taught or practiced sufficiently. We use the Knowledge Integration (KI) framework to evaluate students' connections between individual concepts [4]. In the KI framework, learning is a process of connecting and integrating individual pieces of knowledge, promoting rich connections between prior ideas and new understandings. We draw on the KLI framework to identify our learning goals, and the KI framework to create rubrics for evaluation.

The following KCs show the knowledge targeted by our graph search assignment:

- KC1 Graph search algorithms find a path from a start state to a goal state (where a *path* is an ordered sequence of nodes that are connected by edges).
- KC2 Graph search algorithms construct a *search tree* where the root tree node represents the start node of the graph being searched. The root node is *expanded* by adding (as children) nodes that represent each node in the graph that the start node is connected to. This process of expansion is applied to subsequent nodes, one at a time. If there are no nodes that are candidates for expansion (e.g., because all possible paths from the corresponding node would lead back to nodes that are already on the path), then there is no path from the start node to the goal. If the node being expanded is a goal node, then the algorithm returns that path as the solution (the path is the sequence of nodes from the root of the search tree to the goal). Therefore, graph search algorithms do not direct the actions of an agent as the agent travels the graph. Rather, graph search algorithms reason over a set of paths to find a path to a goal, and in some cases, finding a path that optimizes a mathematically-defined criteria.
- KC3 Search algorithms differ in the order in which they expand nodes in the search tree.
- KC4 For Depth-first search, the next node to expand is the one which was added to the tree most recently (this ordering is reversed for Breadth-first search).
- KC5 For A^* search, ordering is determined by priority, which is calculated by a function. Let n represent a node in the graph being searched (where n is reachable from the start state). The function is of the form:

  f(path from start to n) = g(path from start to n) + h(n)

  Where g(path from start to n) is the total cost of the path from the root of the search tree to that node, and $h(n)$ is the estimated cost of getting from n to the goal state (called a *heuristic function*).
- KC6 The heuristic function h(n) assigns an estimated cost value to all nodes on the graph being searched.
- KC7 In Uniform Cost Search, the function is of the form:

  f(path from start to n) = g(path from start to n)

  That is, it takes into account the cost of the path from the start to the node, but does not take into account any estimate of the cost from that point to the goal.
- KC8 For A^* Search to remain optimal, the heuristic value h(n) needs to be both admissible (the estimated cost is never more than the true cost, $h(n) \leq$ cost(n to Goal)) and consistent (the estimated cost for a node is never greater than the estimated cost from a neighboring node to the goal plus the cost of reaching the neighboring node, $h(n) \leq h(c)$ + cost(n to c)).
- KC9 In Greedy Best-First Search, f(path from start to n) = h(n) (that is, it takes into account the estimated cost from the node to the goal, but not the cost that has been accumulated from the start to the node). This is ordered by goal proximity, or forward cost.
- KC10 In the traditional forms of UCS and A^* Search, g() is a cost function, where the cost of a path is calculated by summing the costs of each of the edges within the path. Therefore, among all possible paths from the start to the goal, the traditional forms of UCS and A^* Search are guaranteed to find the path with the lowest total cost.
- KC11 In the traditional forms of UCS and A^* Search, g() has two key properties: (1) it can be calculated over a path as it is constructed on the search tree (that is, it is incremental), and (2) adding an edge to the path does not change the cost of the previously-constructed portions of the path (it is cumulative). (Note: at this point in the course, students have learned that proofs of optimality for these algorithms hold when the cost function has these properties; proofs regarding search over cost functions without these properties have not been discussed.)
- KC12 *Optimal* has a technical definition in graph search: the optimal path is the one that has the best value as calculated by the objective function (*best* is always the lowest for cost functions, and can be the highest or lowest for objective functions in general).
- KC13 Technical optimality as defined by an objective function is not always the best choice in human contexts because an objective function may not account for important human considerations.
- KC14 Creating an objective function that accounts for human needs requires identifying those needs within the problem context.
- KC15 An appropriate objective function for a human context must align with the human needs. That is, the optimal path as defined by the objective function must be the best path for the human to take in that context.

One of the aspects that makes this assignment more rigorous and nuanced when it is contextualized is that the lowest cumulative path (which is optimal for traditional accumulation problems) may not be the appropriate path for this context. When we consider the additional constraint brought forward by the human context where we do not want the cost of a single segment to be too high, the traditional algorithmic assumptions and calculations are no longer technically optimal or appropriate. Students must critically evaluate their technical decisions in human contexts, rather than simply going through a set of routine steps to complete the calculations.

We expect students to transfer their KCs from the information they were taught in class to a novel human context [5]. In this case, we expect students to be able to calculate the optimal path for various graph search algorithms and critically evaluate whether the chosen path is appropriate for the new context. By understanding this, instructors can pinpoint what skills students currently possess and what KCs are missing to better support transfer. It also allows us to quantify how students' technical reasoning changes in contexts that require ethical reasoning skills when they make these connections and whether these connections can strengthen students' technical understanding of AI topics.

4 METHODS

The instructor of the introductory undergraduate Artificial Intelligence course at our university wanted to integrate ethics into his class. We collaborated to create a graph search assignment that involved human concerns. The instructor drew from our prototype to create a version that met the objectives of his course and his own instructional aims. The instructor assigned the integrated problems as part of a homework on graph search, worth a third of the assignment. While the course syllabus mentioned ethics, prior to this assignment there were no formal discussions about ethics or assignments related to it. After the conclusion of the course, we analyzed de-identified responses to the integrated problems. Since the updates to the AI assignment were done for instructional purposes and since our analysis was done on de-identified data, our IRB determined this work to be non-human subjects research (protocol #00147387). In total, 120 students completed the homework assignment. Students in this class are typically third year undergraduate students majoring in CS, Computer Engineering, or Data Science (CS is the majority). While maintaining the sentiment of the original responses, all student examples presented in this paper have been slightly modified to preserve students' privacy.

4.1 Integrated Assignment where Human Concerns Guide Technical Decisions

Traditional graph search problems ask students how different algorithms would find the shortest or cheapest path from a start node to a goal node. A traditional cost function would sum the edge weights along the path, and the search algorithm would find the optimal path by exploring paths on the priority queue with the lowest cumulative cost. The integrated assignment presents a novel context: a human needs to travel between cities, but at random intervals requires medical care that can only be found in the cities (not on the roads). The shortest overall path is not necessarily the best. One approach to this context is to treat it as a bottleneck problem and find the path that minimizes the length of the longest road. At this point in the course, the algorithms that students had learned were appropriate for accumulation problems, not bottleneck problems. Therefore, the instructor thought this problem incorporated technical rigor by requiring students to consider the conditions for which an algorithm was appropriate. Defining an objective function for this context is not straightforward: even framing this situation as a bottleneck problem has drawbacks. The core task in this assignment is to identify the limitations of the traditional cost function and propose a new one that is better suited to this context.

4.1.1 *Assignment.* The assignment prompt:

> For this section, we are asking for thoughtful consideration of the questions. Different answers can be correct, as long as they consider the algorithms in light of human-centric factors.
> Consider the following graph (Figure 1), which shows the connectivity via roads between five cities. The edges and their labels represent the existence and length of the roadways in between the cities.

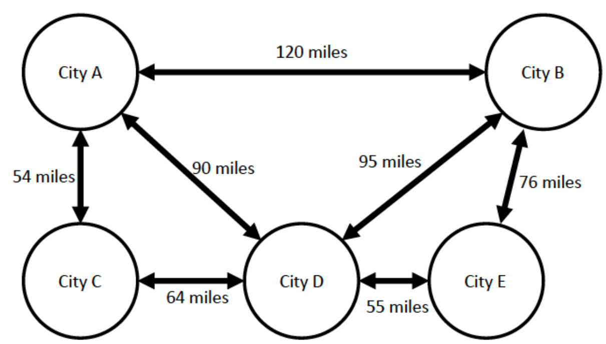

Figure 1: State space graph. The optimal path under the traditional cost function with UCS is $A \rightarrow B$ with a cost of 120. However, this path could be unsafe for Jessie who might need emergency medical treatment and should avoid long roads.

> Jessie, who resides in City A, needs to travel to City B. Jessie has a medical condition that frequently produces life-threatening symptoms at unpredictable times that requires treatment in an emergency room. There are emergency rooms in each of the five cities, but medical care is non-existent on the roads themselves. Please consider and thoughtfully answer the following questions:
> (1) Is shortest path the right cost function when considering the optimal route for Jessie from City A to City B? Why or why not?
> (2) Please describe an alternative cost function that may be appropriate for this problem and justify the choice.
> (3) What is an optimal path from A to B in the above graph under your alternative cost function?

(4) Which, if any, of the search algorithms we've covered will still be optimal for your cost function, both for this specific graph and for an arbitrary graph? Justify your answer.
(5) What is another example of a human-centric factor that demonstrates the complex nature of defining an "optimal" path in real life?

4.2 Data Collection and Analysis

Student responses on the homework assignment were collected and graded by the instructor as a routine part of the coursework. The 120 completed homework assignments were first de-identified by the research team and randomly split into six batches, each consisting of 20 homework assignments. The written responses on the first four batches were inductively coded by the first author to capture the range of student ideas. The second author coded the remaining two batches in the same manner. The codes were then compared and the first two authors worked together to develop a final codebook with 30 codes [20], which was a sufficient number of codes to capture the range of student responses on the homework assignments. The first two authors used the final codebook to code the remaining assignments such that all assignments were coded by both authors. All disagreements were resolved through discussion.

4.2.1 Rubric Development. The codes were grouped into three overall themes (technical AI content, human factors, and the integration of technical and human factors) which were used by the first author to develop three rubrics. For example, the code *A^* without heuristic* (referring to a response that suggested implementing the A^* algorithm but did not define a heuristic to use with it) was categorized under *technical AI content*, as this describes a specific technical misconception (KC8). The code *Considers other human factors* (that is, the student describes another human factor in their answer beyond the human factors that are explicitly mentioned in the problem) was categorized under *human factors* since this relates to the human considerations of the problem. Codes in the *technical AI content* and *human factors* categories were analyzed together to verify consistency and create the *integrated* theme. The process of reflecting on the themes presented by the codes led us to the higher-level themes expressed in the rubrics. Each rubric was refined after each batch of assignments was scored to ensure it categorized all responses. The first author re-scored all assignments with the final version of each rubric. The rubric aims to evaluate whether students have the individual pieces of knowledge necessary to solve the problems (KCs), but also how well they can integrate these ideas and make connections [4]. We use the rubrics to organize and present our results.

5 RESULTS

We present our results through our three rubrics:

- Technical: Demonstration of AI knowledge targeted by the course, including fully describing an algorithm (including the cost function and the determination of expansion order of the search tree, as necessary).
- Human Factors: Identification of human concerns based on the problem context.
- Integrated: Alignment between the proposed algorithm and the identified human concerns.

While we define each rubric level for our specific assignment, the broader themes of each rubric can be applied across contexts.

5.1 Technical

Students may have an intuitive sense of what path might be best for Jessie in the specific graph provided with the problem. The key technical task is to translate that intuition into a precisely-defined algorithm. Students may choose an algorithm they have encountered in class (e.g., UCS or A^*) and propose a modification to the objective function, order of expansion of the search tree, or both. The technical rubric does not examine if the student's algorithm is the most appropriate for Jessie, only if the algorithm is defined in sufficient detail to be executed. We describe student understanding of the technical components in Table 1. In this category, a score of 1 means that the technical learning objectives were not met and there is no evidence that the student has sufficient knowledge of the component. A score of 2 implies that the student has partially met the learning objective but may not fully understand the technical content being assessed. A score of 3 evidences a student's full understanding of the technical component.

5.1.1 Technical: Rubric Level 1. A response at this level does not identify a cost function that can be calculated and does not suggest how an algorithm might be modified to accomplish the stated technical goal, even though an overall objective may be present.

For example, consider the responses that suggested using the number of cities that Jessie passes through as the cost function (Table 1, Level 1). At first glance, finding the route that maximizes the number of cities may seem technically straightforward. However, the student does not explain how to define this objective as a calculable cost function, and there is no evidence that the student understands the technical challenges involved in implementing this idea. As the instructor points out in the feedback returned to the student, "your cost function requires knowing the total number of cities in the path. It's not clear how you would use this information in the incremental construction of a path (as is done in search). So we'd need to describe more how the values in our priority queue are computed in order to execute search as we've described." This student may begin to think about an overall objective (KC12, KC13, and KC14), but they do not demonstrate knowledge of any other KCs.

Consider another common student example, where a student identifies a general objective to keep Jessie close to medical care at all times (KC14), but is unable to articulate what that would look like in a technical implementation and gives no description of an algorithmic step to accomplish this objective (KC2 - KC11). The student may suggest using an algorithm to accomplish their identified non-technical objective, but they do not identify all of the necessary components for a technical implementation (e.g. a heuristic function, KC6 and KC9):

> You can use a greedy algorithm that will always take the shortest road, keeping Jessie as close to a city as possible.

Table 1: Technical Rubric with Student Examples

Level 1: Response does not have a well-defined technical solution (32%).
Student may identify a general objective to optimize, but gives no indication of how to implement this mathematically or algorithmically in an incremental construction of a path as required for graph search algorithms:

- A cost function could consider a path that travels through as many cities as possible while traveling towards City B. While longer, this would be safer since Jessie travels through multiple cities with hospital care on the way to the destination.
- It is better to consider the route that keeps Jessie the closest to medical help as possible.

Level 2: Response has a partial technical implementation (24%).
Student identifies a general objective to optimize, and references an algorithmic step, but does not identify a full mathematical implementation:

- A cost function could be to take the shortest path between cities. That is, when given the choice between any number of cities, always take the shortest road to avoid longer roads.
- An alternative cost function would be a greedy approach that always selects the shortest road to the next city while moving towards the goal city.

Level 3: Response demonstrates full understanding of the technical material (44%).
Student defines a cost function that is mathematically realistic to calculate and viable to implement (a student in this level does not necessarily have to define a cost function that maintains optimality for their chosen algorithm):

- An alternative cost function could be to exponentially punish longer edge traversals using the function: Edge Cost = Distance^k, where k represents the risk of traveling long distances. If the risk was low, you could use something like: $k = 2$ or, if the risk was very high, you could ramp up the punishment of traveling long distances using: $k = 10$.
- An alternative cost function could penalize longer roads by adding an additional cost of 1000 to any distance that is greater than 100 miles.
- A cost function could be the lowest average distance between cities in the path. This way, Jessie would be able to access medical care in the cities much more frequently.

5.1.2 Technical: Rubric Level 2. A student on this level is able to identify an objective that could theoretically work for a graph search algorithm (KC1, KC12, and KC13), and they suggest how this might work algorithmically. However, they do not articulate what this would look like mathematically in a technical implementation of the algorithm as defined in KC2. For a student to demonstrate conceptual understanding of how a chosen cost function might work for a graph search algorithm, they must begin to describe the algorithmic process of implementing their cost function.

For example, a student may calculate the cost for one specific edge, may suggest which choice an algorithm would make, or may note how the algorithm would keep track of which nodes have already been explored or taken off of the priority queue. Each of these demonstrates partial understanding of KC2. Even if a student is unable to articulate how this objective function would be defined mathematically to be implemented in code or in an AI system, a student that describes an algorithmic step (e.g. take the shortest segment when given the choice of two or more segments) demonstrates that they are starting to think about a technical implementation. However, since a student at this level does not mathematically define the cost function or explain how the algorithm would need to be modified in cases where the cost function would not work with the traditional algorithm, they only partially meet the technical learning objective for KC2. This is the first time that students are asked to come up with their own cost function, rather than being given one as defined by the algorithm. For this reason, students may not be reasonably expected to identify a cost function that is realistic and executable.

5.1.3 Technical: Rubric Level 3. A student who can articulate a cost function that can be technically calculated has demonstrated a flexible understanding of the technical material (KC1 - KC11). For example, a suggestion of punishing longer edge traversals by raising the cost to a power relative to Jessie's risk tolerance is both technically calculable and implementable with algorithms that the student has identified (Table 1, Level 3). This student uses their cost function implemented with a UCS algorithm to successfully calculate the optimal path under their conditions, demonstrating understanding of KC10 and KC11:

> Using a high punishment value of: Edge Cost = Distance^{10}, the optimal path would be: $A \rightarrow C \rightarrow D \rightarrow E \rightarrow B$. A uniform cost search would be able to solve this problem with this cost function because once the edge weights are recalculated, the problem is still a shortest path traversal.

A student that excels in this level of the rubric is able to reason through the set of paths explored in graph search and may justify necessary changes to an algorithm to be viable with their cost function (KC2, KC8, KC10, and KC11). While technically feasible, this solution is not necessarily the best solution for Jessie (as discussed further in Section 5.3).

5.2 Human Factors

Students should acknowledge that there are human concerns they need to carefully consider in this problem, specifically that it is risky for Jessie to drive on long stretches of roads in case they need emergency medical care, which is unavailable on the roads. In this example, students should recognize that the shortest path is not the best path for Jessie to take, and the algorithm should prioritize safety over speed or distance. The human factors rubric (Table 2) evaluates students' consideration of the human factors that should be examined when solving this problem. Some students were able to identify the complexity of the problem presented by the human needs since humans can be unpredictable, their needs may change, or they may change their priorities, which is categorized as a level 3 on this rubric. Others assumed Jessie's wants or needs, or they did not consider outside factors that may need to be taken into consideration when building technology to prioritize human safety (level 2). A level 1 on this rubric would imply that a student did not take any human factors into consideration.

5.2.1 Human Factors: Rubric Level 1. At this level, a response does not consider the human factors presented in the problem (e.g., finding the shortest path, regardless of Jessie's needs). However, all students in our sample considered Jessie's needs to some extent, so there are no student examples of this rubric level.

5.2.2 Human Factors: Rubric Level 2. At this level, a response considers the human needs as presented explicitly in the prompt, for the specific situation at hand. For example, students may propose a path that always keeps Jessie as close as possible to a city (Table 2, Level 2). These students recognize that the shortest overall path is not always best for Jessie, and that shorter roads will help Jessie remain close to a hospital (KC14). While this reasoning makes sense for the specific graph in the question (Figure 1), it does not hold for all contexts. In Figure 2, for example, the shortest overall path is one segment of 30 miles, while the alternative path is four segments of 29 miles each. Even if safety is Jessie's only concern, the trade-off of marginally shorter segments is likely not worth the risks of adding 86 miles to the journey. While a student at this level of the rubric does not consider factors that are not explicitly presented in the problem, they still demonstrate understanding of KC14.

5.2.3 Human Factors: Rubric Level 3. At this level, a response considers human needs beyond those stated explicitly in the problem, and beyond the specific graph provided. The needs of humans are often unpredictable and nuanced, and a student at this level of the rubric recognizes this and takes this into consideration when describing their justification or solution (KC14). For instance, while Jessie needs to remain close to medical care, there may be a trade-off between how long the total path takes and the safety of the path. Jessie may be willing to risk driving a little further if it saves quite a bit of time on the overall trip. A student that recognizes these complexities and nuances demonstrates that they understand humans may change their definition of "optimal" depending on their priorities, circumstances, or even feelings in the moment. This

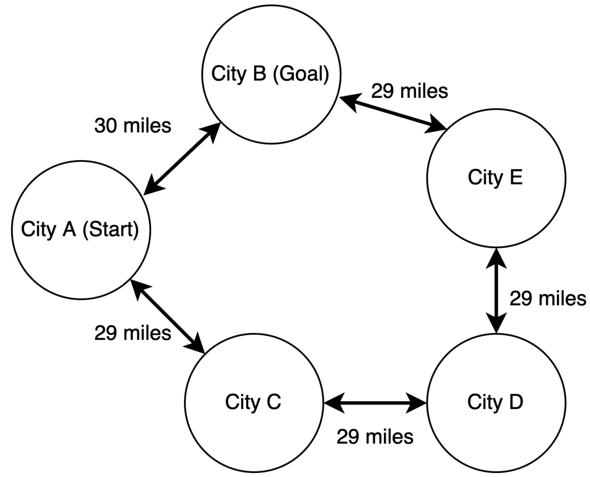

Figure 2: State space graph. Using a student's defined rule of always taking the shortest road to the next city would return the path $A \rightarrow C \rightarrow D \rightarrow E \rightarrow B$, even though a more reasonable path would be $A \rightarrow B$.

makes technical calculations all the more complex and challenging, a realistic problem that AI practitioners will face in the real world.

5.3 Integrated

The integrated rubric (Table 3) evaluates a student's ability to align their technical choices with human goals. In this category, a student not only needs to be able to complete the technical AI calculations and sufficiently identify the human needs in the problem, but they also need to be able to conciliate their technical and human-centered goals. Level 1 responses have technical and human components that do not align, either because the solution is well-defined technically but does not accomplish the stated goal (1A), or because the technical solution is not well-defined (1B). In both cases, human factors are identified. Level 2 responses define a technical solution that is consistent with the identified human goals.

5.3.1 Integrated: Rubric Level 1A. A student on this level identifies a reasonable technical choice that is, however, not consistent with their intent based on the human factors that they identify. This type of response demonstrates that a student does not understand the potential impact of their technical choices in a human context. Such a response would score high on the technical rubric, but may receive any score on the human factors rubric. The key feature of this level is that a student's answer indicates that their technical implementation does not match the human factors that they identified as important.

For instance, consider a student that wants to keep Jessie on a path that keeps them close to medical care, demonstrating understanding of KC13 and KC14 (Table 3, Level 1A). This student suggests a cost function and algorithm that minimize the average distance between all cities on a path (a technically viable and calculable solution). However, this solution will not always make sense for Jessie. For example, under this cost function, a path with one 10-mile segment and 20 1-mile segments would be selected over

Table 2: Human Factors Rubric with Student Examples

Level 1: Response does not consider human factors or context (0%).
Student does not think critically about Jessie's needs.

Level 2: Response considers human needs as they are explicitly presented, but questions are answered in a vacuum and the nuances of human needs are not critically considered (i.e. student treats this as a "toy" example. The student only considers what is presented in the problem, even though a real-world problem would have much more nuance than is made explicit in the question) (84%).
Student considers Jessie's needs as they are explicitly presented:

- The shortest path would not be the right cost function when considering Jessie's circumstances since the shortest path (City A straight over to City B) has the longest stretch of road at 120 miles. This would be an issue since Jessie may need emergency care at unpredictable times, and the shortest path could put them too far away from a hospital. A better cost function would take Jessie's needs to remain close to a city into consideration.
- It would be safer for Jessie if there would be shorter distances between cities rather than longer distances so that Jessie can get access to a emergency room in the nearby city quickly. Therefore, we can reward the cost function by entering a city.

Level 3: Response mentions that the definition of optimal will change in varying human situations and technical questions are answered in context beyond what is stated in the problem (16%).
Student considers that humans are unpredictable and therefore Jessie's needs may change:

- My answer depends on Jessie's priorities. If they want to travel from City A to City B as quickly as possible, then the shortest path is the best cost function to use for this problem. However, if they want to travel as safely as possible by staying as close to medical care as they can, then the shortest overall path from City A to City B isn't the best option since this is the longest path without medical care. For me personally, I would prioritize a safe trip rather than a quick but risky trip.
- Let's assume that Jessie can travel for a certain distance and still be saved, say 30 miles. Also assume that cities all have a medevac helicopter that can travel 200 miles in that same time, but at a much increased cost to Jessie (say equivalent to 10,000 miles driven, no matter the flight distance, due to the insurance copay). Also assume that Jessie has a relatively low chance of an incident, say 0.001 per mile (0.1%). For ease of use, consider the cost of 1 mile to be 1 unit. So the cost in total for a road of length d is:

$$\begin{cases} d & \text{if } d \le 60 \\ 60 + 10,000 * 0.001 * (d - 60) = 10d - 540 & \text{if } d > 60 \end{cases}$$

  This ensures Jessie could turn around and go back home on the first 30 miles, continue on in the last 30 and that each mile out of driving range of a hospital ends up costing 10 units.
- It depends - if Jessie is very cautious then they may want to stay close to a hospital at all times and always move to the closest city. At 60mph, it would take about 8 hours to drive from City A to City C, D, E, then B and back but Jessie would only spend about an hour driving between each city. Then, we can define a cost function that minimizes the time taken to drive between cities.

a path with just one 10-mile segment. Despite individual understanding of some KCs, the student's solution does not demonstrate successful mastery of KC15.

5.3.2 Integrated: Rubric Level 1B. A student on this level considers the impacts of their technical choices and makes decisions that minimize human harm, but does not identify how to execute their ideas technically. While the student would score high on the human factors rubric and may demonstrate understanding of KC13 and KC14, they do not suggest a viable technical solution and therefore would score low on the technical rubric (KC2 - KC11).

Consider the student that recognizes Jessie's need to be close to a city, but suggests "the closest city" as an alternative cost function (Table 3, Level 1B). This student notes that they need to identify a cost function and algorithm that will ensure Jessie is never too far from medical care in case of an emergency (KC14). However, they do not describe a technical implementation of this, as their suggestion is not a usable cost function, demonstrating a misconception of KC2, KC5, KC10, and KC11. Another student may carefully consider the trade-offs involved in various technical choices when considering Jessie's complex needs:

> Developing a complicated function that minimizes Jessie's risks is no easy task, and without proper data, it's impossible to come up with the least-risky path for Jessie. While the goal seems easy to define, we cannot come up with metrics that ensure Jessie's safety while taking other factors into account. We could choose different metrics to prioritize, but Jessie's preferences,

Table 3: Integrated Rubric with Student Examples

Level 1A: Response is well-defined technically, but the implementation of the technical decision does not match the student's intention based on the human factors (15%).
Student defines a reasonable technical cost function and algorithm, but the implementation of this does not match their intent for optimizing this problem based on Jessie's needs or desires:

- A cost function could be the lowest average distance between cities in the path. This way, Jessie would be able to access medical care in the cities much more frequently.
- Choose a cost function that penalizes heavily against longer roads to avoid the long road. Perhaps cost = miles^2.

Level 1B: Response carefully considers the human factors involved in the problem, but there is no identification of how to execute this technically (56%).
Student has thoughtfully considered Jessie's needs or desires, but does not describe how this could be done in a technical implementation:

- A cost function could be the closest city. While the path would be longer, Jessie would be safer in case of medical emergency.
- A better cost function would be the travel time for Jessie, accounting for traffic, state of the road, etc.
- A cost function could be to use the road distance. Minimizing the road distance keeps Jessie close to a city.

Level 2: Response carefully considers the human factors involved in the problem, and identifies how to execute this technically in a way that is consistent with the student's intent based on the human needs (29%).
Student has thoughtfully considered Jessie's needs or desires and defines a realistic and implementable technical solution that is consistent with the student's intent based on the human needs:

- A cost function would need to take Jessie's risk of life-threatening symptoms into account. For example, using the cube of the distance as the cost function makes longer roads much more costly than shorter roads. Using this, the optimal path under UCS would be $A \rightarrow C \rightarrow D \rightarrow E \rightarrow B$. However, the value of the exponent can change according to how much risk Jessie wants to take. UCS would still be optimal, since this cost function strictly applies to the distances between cities and does not take anything else into account. It would still find the optimal path assuming Jessie wants to take the least risky path possible.
- We want to define a cost function that punishes large distances between cities. Instead of summing the distances between the start and the path, it might be better to sum the distances raised to the power of n where n would be a hyperparameter. At first I was going to set $n = 2$, but then the optimal path is still $A \rightarrow B$, so we could set this value higher (say, $n = 3$). However, we don't have any details about Jessie's condition or how urgent the care is. Although $A \rightarrow B$ has a long road on it, taking a longer overall path through shorter roads covers much more distance overall which increases the chance an emergency would happen on the drive.
- To ensure Jessie stays within a safe distance from medical care, we could assign a cost of infinity to roads that are too long. For example, assume Jessie can be saved as long as they are at most 50 miles away from the nearest hospital in either direction. Then, we could define a cost function as:

$$\text{cost(City A, City B)} = \begin{cases} dist\text{(City A, City B)} & \text{if } \frac{dist\text{(City A, City B)}}{2} \leq 50 \\ \infty & \text{otherwise} \end{cases}$$

willingness to take risks, or other priorities make it challenging to define a single optimal solution. When risk and odds are introduced into a problem, the best we can do is maximize those odds in our favor - but without perfect data, all we have are effectively better guesses.

This student considers the nuances of human wants and needs (KC14), but similarly does not identify a viable technical solution (KC2 - KC11).

5.3.3 Integrated: Rubric Level 2. A successful student is able to consider various technical choices while minimizing any harm to Jessie. The implementation of their technical solution is consistent with their thoughtful consideration of the human factors involved in the problem (KC15). Consistency is important: the student may score highly on the individual technical and human factors rubrics but score poorly on the integrated rubric if they do not conciliate the technical and human-centered concerns. The students with a score of 2 in Table 3 articulate Jessie's need to avoid long stretches of roads and recommend reasonable and calculable cost functions, receiving high technical and human factors scores. Further, the first two students' technical suggestion to use the cube of the cost as the new cost function (Table 3, Level 2) is *consistent* with their goal of keeping Jessie off of longer stretches of roads (as noted in the

Table 4: Percentage of assignments that were scored in each level of the technical and human factors rubrics (*Note*: Human Factors Level 1 has been removed as no students scored in this level of the rubric).

		Human Factors	
		2	3
Technical	1	23.5%	8%
	2	23.5%	1%
	3	37%	7%

Table 5: Percentage of assignments that were scored in each level of the technical, human factors, and integrated rubrics (*Note*: Human Factors Level 1 has been removed as no students scored in this level of the rubric).

		Integrated		
		1A	1B	2
Technical	1	0%	32%	0%
	2	0%	24%	0%
	3	15%	0%	29%
Human Factors	2	14%	47%	23%
	3	1%	9%	6%

second response, squaring the distances is not sufficient to find the optimal path for the given graph). All responses in this level provide technical details to reasonably operationalize the human goals.

5.4 Scores across Rubric Dimensions

Table 4 shows the scores on the technical rubric crossed with scores on the human factors rubric, and Table 5 shows the scores of the integrated rubric crossed with the other two. Table 5 demonstrates that the assignment revealed varying levels of both technical knowledge and considerations of the human factors (note that no student completely disregarded the human factors, so there were no scores of 1 on that rubric).

The integrated rubric evaluates students' ability to craft a technical solution that responds to the identified human needs. Therefore, a 3 on the technical rubric (a well-defined technical solution) and at least a 2 on the human factors rubric (considers human needs as presented) are necessary for a high score on the integrated rubric. However, they are not sufficient, because a response may present a well-defined technical solution that does not match the students' stated intentions regarding the human factors. For example, a student may state that Jessie should avoid long road segments (scoring 2 on the human factors rubric), and may propose to use UCS with a cost function that takes the square of the each road segment (scoring a 3 on the technical rubric because the modification to the traditional cost function is well-defined and the path returned by UCS is guaranteed to be optimal under the new cost function). However, the path returned for the graph given in the problem is still the direct route from A to B, traveling the longest segment of the graph. Since the algorithm does not implement the student's stated goal of avoiding long roads, it does not score highly on the integrated rubric. To sufficiently punish longer segments of this graph, the function needs to use at least a cubic. Overall, 44% of students scored high enough on the technical rubric to be eligible for the highest integrated score (all students scored the necessary 2 or 3 on human factors). However, only 29% of students scored a 3 on on the integrated rubric, demonstrating how it adds insight above the isolated rubrics.

6 DISCUSSION

Evaluation of technical AI knowledge and human factors individually is not sufficient for what we aim to teach students when we consider ethics in AI (as defined in Section 2.2) - that technical implementations may have human impacts. This is made explicit by the students that scored highly on both the technical and human factors rubrics, but not on the integrated rubric. Even if an AI practitioner deeply understands the technical components and considers human needs in their work, they could still cause harm if their technical choices do not align with their human considerations. For this reason, it is critical to integrate ethical and technical concepts in the classroom.

6.1 Classroom Implications and Instructor PCK

The assignment and process of evaluating the assignment generated instructor pedagogical content knowledge (PCK) by revealing nuances of student misconceptions about graph search algorithms. According to the instructor, one of the most enlightening factors of this process was that the homework revealed students' technical misconceptions in a way that was unique to this type of assignment. The concept that certain algorithms work by calculating the cumulative cost of the path was not clear to students (KC10 and KC11). It is often assumed that students understand that algorithms such as UCS calculate a cumulative path cost because they are only exposed to cumulative versions of these algorithms in class. However, on a typical technical assignment, students are not required to question whether the assumptions of each algorithm are met. They simply need to calculate the optimal path given all of the necessary information. This practice may have masked these misunderstandings.

The instructor also noticed that many students did not specify how their suggestions might be implemented technically. E.g., a student that suggested finding a path that minimizes the maximum segment length on the path without suggesting how an algorithm might be modified to accomplish this goal. While this is a reasonable strategy for solving the problem, it is unclear that the student understands how such a path might be constructed by a graph search algorithm. While these poorly defined solutions could often be translated into implementable technical definitions of a cost function by an expert, it was unclear that the student would be able to do this. By addressing the misconceptions revealed through this integrated assignment, the instructor can ensure deeper understanding during instruction [28]. Specifically, this assignment revealed that many students do not have a flexible enough understanding of the technical detail required to implement a graph search algorithm (KC2 - KC11) or to define a technical objective that is consistent with their identified goals based on the human needs (KC12 - KC15).

Based on the students' responses, the instructor adjusted his instruction to clarify common misconceptions (including how the different algorithms consider the ordering of nodes for expansion, and that a cost function is defined over paths and is not the same as the edge weights).

Another consideration for instructors is the challenge of grading these types of assignments. As presented by this work, integrated assignments necessitate integrated evaluation. However, reasoning through a student's technical (or non-technical) implementation to determine if it is consistent with their defined human goals is a nontrivial task. While integrating human concerns within a technical assignment has benefits including deepening technical rigor and requiring students to think critically about their solutions, it also places more burden upon the instructor to evaluate these solutions. Future work might explore ways to make grading integrated assignments more scalable.

6.2 Re-contextualizing Technical Assignments in AI Courses

Our hope is that this work provides a blueprint for other AI instructors to replicate our process of re-contextualizing a technical assignment. By replicating this process, AI instructors can integrate human concerns into technical CS courses with less effort than building an assignment from scratch. Additionally, re-contextualization can help students begin to recognize that their technical calculations may have real-world implications, and it is their responsibility as a professional to critically evaluate their technical decisions. Impressing this responsibility upon CS students is important because they often do not believe that they are going to be responsible for making ethical decisions in their career and they feel that ethics is inherently separate from the technical content they are learning in class [23].

6.3 Future Work

Our findings can be used to improve future iterations of this assignment. For instance, we will change the wording of "cost function" to "objective function" to clarify that there are a variety of ways students can define a technical objective to solve this problem. Additionally, it was not apparent whether students calculated the optimal path using their cost function and chosen algorithm or if they simply looked at the graph and determined a path they thought would be optimal for Jessie. For this reason, we will clarify the expectations in future iterations of the assignment and give students the opportunity to suggest which path they would recommend that Jessie take based on both their human judgment and their technical calculations.

Instructors and researchers recognize the difficulty of creating an assignment that integrates ethics while assessing technical AI knowledge. It can be challenging to find ethical contexts to situate abstract concepts and time consuming to create new assignments of this type. Future work could explore ways to overcome the challenges involved in incorporating these assignments across the curriculum.

6.4 Limitations

While this method does not encompass everything that it means to integrate ethics within technical courses, this is in line with and builds upon previous research in the field that connects technology and ethics [7]. We do not claim that student responses to this assignment are representative of the responses of all students on ethically-integrated AI assignments. This assignment was administered in a single course in an online format due to the COVID-19 pandemic. Additionally, we are limited to the students' written responses on these questions which do not wholly reveal their reasoning. For example, students may not be able to or be motivated to articulate their thought process. Another limitation is that the integrated questions may have caused extraneous cognitive load or burden that we cannot identify from students' answers alone. There is also a potential that we are not measuring what we truly care about, which is understanding whether students will be prepared to consider the human impacts of their technical decisions beyond the classroom.

Additionally, this is still a simplified example that does not reflect all of the complexities AI practitioners would have to consider while solving real-world problems with human components. For example, if an AI practitioner were actually coding an assistive AI application to recommend a driving path for Jessie, they would need to consider many other factors involved in driving (e.g. traffic, speed limits, road quality, etc.) as well as the best way to present this information to the human. Although this work explores student perceptions of their technical choices as they impact humans, there is still the potential that students will make harmful decisions even if they recognize the ethical concerns involved in their work [10]. Even so, we think there is value in presenting students with any example of how their technical choices could have real impact. The simplified example gives them an opportunity to think through their solutions in a low-stakes environment that allows them to learn and grow.

7 CONCLUSION

In this paper, we have demonstrated successful integration of ethical components into a technical AI homework assignment and successful evaluation of students' responses to the new assignment. Our re-contextualized AI homework assignment requires students to think through whether their proposed solutions may cause human harm and ensure that their technical implementations account for human factors. We showed that integrating human factors can deepen the technical rigor of an assignment. However, an integrated assignment necessitates integrated evaluation. Integrated evaluation requires us to simultaneously assess ethical components and technical content. We can do this by assessing whether a student is able to appropriately alter their technical choices based on the human factors they identify. We developed three rubrics to evaluate this knowledge by qualitatively analyzing student responses to the integrated homework assignment. The technical rubric evaluates students' technical AI knowledge (what would be evaluated when grading traditional AI problems), the human factors rubric evaluates students' ability to consider the human needs presented in the problem (what would be evaluated when grading traditional "ethics" problems), and the integrated rubric evaluates the alignment of these two components (how one impacts the other).

ACKNOWLEDGMENTS

We thank Alan Kuntz, our partner instructor. This work is supported in part by the National Science Foundation (NSF) award SaTC-EDU 2041960.

REFERENCES

[1] ACM Code of Ethics 2018. Code 2018 - ACM ethics. https://ethics.acm.org/code-of-ethics/code-2018/

[2] A. Asimov, M. H. Greenberg, and P. S. Warrick. 1983. *Machines That Think.* Holt. Rinehart, and Wilson.

[3] Emanuelle Burton, Judy Goldsmith, and Nicholas Mattei. 2016. Using "the machine stops" for teaching ethics in artificial intelligence and computer science. *Association for the Advancement of ArtificialIntelligence* (2016).

[4] Jennifer L Chiu and Marcia C Linn. 2014. Supporting Knowledge Integration in Chemistry with a Visualization-Enhanced Inquiry Unit. *J. Sci. Educ. Technol.* 23, 1 (April 2014), 37–58.

[5] National Research Council. 2000. *Learning and Transfer: Chapters 2 and 3 How People Learn: Brain, Mind, Experience, and School: Expanded Edition.* The National Academies Press, Washington, DC.

[6] Stacy A Doore, Casey Fiesler, Michael S Kirkpatrick, Evan Peck, and Mehran Sahami. 2020. Assignments that Blend Ethics and Technology. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) *(SIGCSE '20)*. Association for Computing Machinery, New York, NY, USA, 475–476.

[7] Casey Fiesler, Mikhaila Friske, Natalie Garrett, Felix Muzny, Jessie J Smith, and Jason Zietz. 2021. Integrating Ethics into Introductory Programming Classes. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education.* Association for Computing Machinery, New York, NY, USA, 1027–1033.

[8] Casey Fiesler, Natalie Garrett, and Nathan Beard. 2020. What Do We Teach When We Teach Tech Ethics? A Syllabi Analysis. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) *(SIGCSE '20)*. Association for Computing Machinery, New York, NY, USA, 289–295.

[9] Sorelle Friedler, Kathi Fisler, Suresh Venkatasubramanian, and Seny Kamara. [n.d.]. Responsible Problem Solving course materials repository. https://responsibleproblemsolving.github.io/

[10] Colin M Gray, Shruthi Sai Chivukula, Kassandra Melkey, and Rhea Manocha. 2021. Understanding "Dark" Design Roles in Computing Education. In *Proceedings of the 17th ACM Conference on International Computing Education Research* (Virtual Event, USA) *(ICER 2021)*. Association for Computing Machinery, New York, NY, USA, 225–238.

[11] Nancy Green. 2021. An AI Ethics Course Highlighting Explicit Ethical Agents. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society.* Association for Computing Machinery, New York, NY, USA, 519–524.

[12] Colleen Greer and Marty J Wolf. 2020. Overcoming barriers to including ethics and social responsibility in computing courses *(ETHICOMP 2020)*.

[13] Barbara J. Grosz, David Gray Grant, Kate Vredenburgh, Jeff Behrends, Lily Hu, Alison Simmons, and Jim Waldo. 2019. Embedded EthiCS: Integrating Ethics across CS Education. *Commun. ACM* 62, 8 (July 2019), 54–61. https://doi.org/10.1145/3330794

[14] Diane Horton, Sheila A McIlraith, Nina Wang, Maryam Majedi, Emma McClure, and Benjamin Wald. 2022. Embedding Ethics in Computer Science Courses: Does it Work?. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1* (Providence, RI, USA) *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 481–487.

[15] Nazish Zaman Khan and Andrew Luxton-Reilly. 2016. Is Computing for Social Good the Solution to Closing the Gender Gap in Computer Science?

[16] Shamika Klassen and Casey Fiesler. 2022. "Run Wild a Little With Your Imagination": Ethical Speculation in Computing Education with Black Mirror. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1* (Providence, RI, USA) *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 836–842.

[17] Amy J Ko, Alannah Oleson, Neil Ryan, Yim Register, Benjamin Xie, Mina Tari, Matthew Davidson, Stefania Druga, and Dastyni Loksa. 2020. It is time for more critical CS education. *Commun. ACM* 63, 11 (Oct. 2020), 31–33.

[18] Kenneth R Koedinger, Albert T Corbett, and Charles Perfetti. 2012. The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. *Cognitive Science* 36, 5 (2012), 757–798.

[19] Kevin Lin. 2021. Do Abstractions Have Politics? Toward a More Critical Algorithm Analysis. In *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT).* IEEE, 1–5.

[20] K M MacQueen, E McLellan-Lemal, K Bartholow, and B Milstein. 2008. Team-based codebook development: Structure, process, and agreement. *Cultural Anthropology Methods* 10, 2 (Jan. 2008), 119–135.

[21] James W Malazita and Korryn Resetar. 2019. Infrastructures of abstraction: how computer science education produces anti-political subjects. *Digital Creativity* 30, 4 (Oct. 2019), 300–312.

[22] C Dianne Martin, Chuck Huff, Donald Gotterbarn, and Keith Miller. 1996. Implementing a tenth strand in the CS curriculum. *Commun. ACM* 39, 12 (Dec. 1996), 75–84.

[23] Nora McDonald and Shimei Pan. 2020. Intersectional AI: A Study of How Information Science Students Think about Ethics and Their Impact. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW2 (Oct. 2020), 1–19.

[24] Donald McMillan and Barry Brown. 2019. Against Ethical AI. In *Proceedings of the Halfway to the Future Symposium 2019* (Nottingham, United Kingdom) *(HTTF 2019, Article 9)*. Association for Computing Machinery, New York, NY, USA, 1–3.

[25] Koby Mike. 2020. Data Science Education: Curriculum and pedagogy. In *Proceedings of the 2020 ACM Conference on International Computing Education Research* (Virtual Event, New Zealand) *(ICER '20)*. Association for Computing Machinery, New York, NY, USA, 324–325.

[26] Hilarie Nickerson, Catharine Brand, and Alexander Repenning. 2015. Grounding Computational Thinking Skill Acquisition Through Contextualized Instruction. In *Proceedings of the eleventh annual International Conference on International Computing Education Research* (Omaha, Nebraska, USA) *(ICER '15)*. Association for Computing Machinery, New York, NY, USA, 207–216.

[27] Evan M. Peck. [n.d.]. Ethical reflection modules for CS 1. https://ethicalcs.github.io/

[28] Yizhou Qian and James Lehman. 2017. Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. *ACM Trans. Comput. Educ.* 18, 1 (Oct. 2017), 1–24.

[29] Inioluwa Deborah Raji, Morgan Klaus Scheuerman, and Razvan Amironesei. 2021. You Can't Sit With Us: Exclusionary Pedagogy in AI Ethics Education. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (Virtual Event, Canada) *(FAccT '21)*. Association for Computing Machinery, New York, NY, USA, 515–525.

[30] Jean Michel Rouly, Huzefa Rangwala, and Aditya Johri. 2015. What Are We Teaching? Automated Evaluation of CS Curricula Content Using Topic Modeling. In *Proceedings of the eleventh annual International Conference on International Computing Education Research* (Omaha, Nebraska, USA) *(ICER '15)*. Association for Computing Machinery, New York, NY, USA, 189–197.

[31] Jeffrey Saltz, Michael Skirpan, Casey Fiesler, Micha Gorelick, Tom Yeh, Robert Heckman, Neil Dewar, and Nathan Beard. 2019. Integrating Ethics within Machine Learning Courses. *ACM Trans. Comput. Educ.* 19, 4 (Aug. 2019), 1–26.

[32] Kimberley Scott, Gregory Aist, and Xiaolong Zhang. 2014. Designing a Culturally Responsive Computing Curriculum For Girls. *International Journal of Gender, Science and Technology* 6, 2 (2014).

[33] Mona Sloane. 2019. Inequality Is the Name of the Game: Thoughts on the Emerging Field of Technology, Ethics and Social Justice. In *Weizenbaum Conference.* DEU, 9.