

# A Pilot Study of an Instrument to Assess Undergraduates' Computational thinking Proficiency

Debby Erce Sondakh<sup>1</sup>, Kamisah Osman<sup>2</sup>, Suhaila Zainudin<sup>3</sup>

Faculty of Education, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia<sup>1,2</sup>

Faculty of Computer Science, Universitas Klabat, Manado, Indonesia<sup>1</sup>

Faculty of Technology and Information Science, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia<sup>3</sup>

**Abstract**—The potentiality of computational thinking (CT) in problem solving has gained much attention in academic communities. This study aimed at developing and validating an instrument, called Hi-ACT, to assess CT ability of university undergraduates. The Hi-ACT evaluates both technical and soft skills applicable to CT based problem solving. This paper reports a pilot study conducted to test and refine the initial Hi-ACT. Survey method was employed through which questionnaire comprising of 155 items was piloted among 548 university undergraduates. Structural equation modeling with partial least squares was applied to examine the Hi-ACT's reliability and validity. Composite reliability was used to assess internal consistency reliability, while convergent validity was evaluated using based on items' outer loadings and constructs' average variance extracted. As a result, 41 items were excluded, and an instrument to assess CT ability comprising 114 items and ten constructs (abstraction, algorithmic thinking, decomposition, debugging, generalization, evaluation, problem solving, teamwork, communication, and spiritual intelligence) was developed. The reliability and validity of the Hi-ACT in its pilot form have been verified.

**Keywords**—Computational thinking; assessment; skills; attitudes; undergraduates; self-assessment

## I. INTRODUCTION

The ability to solve a complex problem is demanded, regardless of the field in which we work. Wing [1] introduced computational thinking (CT) as problem solving approach that using the way computer scientist think. Her vision is, the set of CT skills and attitudes will be beneficial for everyone, not only computer science majors. Further studies had reinforced that CT enables one to become a technology builder rather than a mere technology consumer [2], develops logic, creativity, innovative thinking [3], and analytical skills [4]. The World Economic Forum [5] found these attributes are increasingly in demand in the digital world workplaces.

Further, recognition of CT as an essential skill for all students is expanding rapidly. Accordingly, initiatives are underway to bring CT into educational institution around the world. Among them, a number of recent studies focus on incorporating CT in classroom/curriculum [6], [7], some on creating artifacts with which to teach CT principles [8], [9], as well as on assessment [10]–[12]. Other studies [13]–[16] highlighted teachers' conception of CT.

In this work, we focus on CT assessment at the undergraduate level. Some studies have initiated CT

assessments for undergraduates. An instrument, which tries to test the correlation between CT and critical thinking has been developed [17]. Specifically, this instrument assesses simple algorithms, sorting method, file structure, and digital information storage. The author used multiple choice questions and short answer questions. However, it has not been validated.

In another study [18] a test to identify CT skill of first-year computer science students was developed. It was based on six classes of CT skills and practices defined in 'Computational Thinking Framework,' i.e. models and abstractions, patterns and algorithms, processes and transformation, tools and resources, inference and logic, and evaluations and improvements.

A paper-based test, called 'The Testing Algorithmic and Application Skills' is presented [19]. This test measures algorithmic skills, computer science terminology used, and problem solving abilities. Particularly, it comprises questions related to the students' computer usage habits, self-assessment on their knowledge in informatics, and tasks of traditional programming, numerical system calculation, handling files, word processing, and spreadsheet programming.

The aforementioned CT assessment studies mostly highlighted the skills, and little has been done to include attitudes. Contrariwise, according to Wing [1], CT comprises both skill and attitudes necessary in solving problems. There is thus a need for an instrument that includes items and constructs to measure students' CT competency, in terms of skills and attitudes. Therefore, in the light of Wing's original conception of CT as a set of skills and attitudes to solve problem, this work proposes an instrument, the "Holistic Assessment of Computational Thinking" (Hi-ACT), to test undergraduates' perceptions of their CT competency. We use the term 'holistic' to describe the inclusion of both skills and attitudes in the CT assessment framework. This paper reports a pilot study that was conducted to assess and refine the initial Hi-ACT by examining its reliability and validity. This is an extended version of the work published in [20].

## II. COMPUTATIONAL THINKING

CT has been noticed as a major research field since the publication of Wing's remarkable article in 2006. However, several researchers noted the long history of CT [21]–[24], as presented in Fig. 1. As early as 1945, George Polya emphasized the application of disciplined manner, decomposition, and generalization (reuse common techniques)

to solve the everyday problem [24]. In 1962, Perlis proposed his vision that programming concept would foster the ability to understand various topic outside computer science and become a vital part of education [21]. As noted by Denning [22], from the field of science, L.K. Wilson introduced the ‘computational science,’ a computation-based approach to exploit existing knowledge and discover the new one. Thereafter, Seymour Papert, in 1980, found that ‘thinking like a computer’ was a useful component of thinking skills to teach mathematics to children [23]. Programming symbols and representation were used in solving mathematical problems.

Further, in 2006, Wing introduced CT, a way of thinking to solve problems, design systems, understand human behavior, using computer science based concepts. Several researchers [2], [25]–[27] revisited Wing’s definition of CT provide a definite understanding of CT and to perceive its core principles. Denning [26] defined CT as a mental orientation to formulate problems through what so-called ‘conversion’. Algorithms are applied to convert some input into an output. Other studies described CT as problem solving process [2], and

the essence of is ‘thinking like a computer scientist’ [27]. Wing refined her early delineation of CT to be “the thought processes involved in formulating problems and their solutions that can be effectively carried out by an information-processing agent; a human or machine, or combinations of humans and machines” [28]. Aho [25] then simplified Wing’s refinement by defining CT as the thought processes to formulate solutions to the problems, which represented as computational steps and algorithms. Put simply, the core of CT is to approach a problem using computer scientists’ way of thinking.

CT adopts some fundamental concepts of computer science as its skills [1]. There are varying views considering CT skills (Table I). Along with the skills, attitudes are also required in CT-based problem solving [1]. Barr et al. [2] used the term ‘dispositions’ to describe the values, motivations, feelings, stereotypes and attitudes’ appropriate to CT. It, therefore, can be said that, in CT, attitudes is indeed necessary for solving problems using. Nevertheless, as shown in Table II, only a few works of literature that considered attitudes.

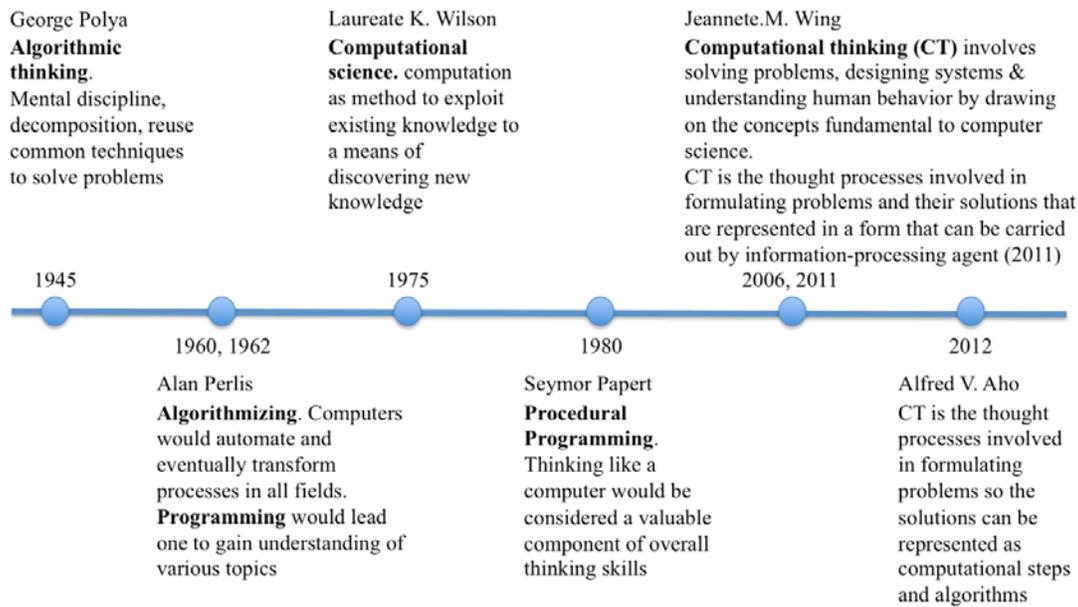


Fig. 1. Computational Thinking Evolution.

TABLE I. CT SKILLS DISCUSSED IN THE LITERATURE

	Abstraction	Algorithmic	Automation	Decomposition	Generalization	Logical reasoning	Evaluation	Debugging
Wing [1], [29]	√	√	√					
Barr et al. [2]	√	√	√	√	√	√		
Selby and Wollard [30]	√	√		√	√		√	
Grover and Pea [27]	√	√		√			√	√
Angeli et al. [31]	√	√		√	√			√
Bocconi et al. [32]	√	√	√	√	√			√
Mueller et al. [33]	√	√		√	√		√	√

TABLE II. CT ATTITUDES (SOFT-SKILLS) DISCUSSED IN THE LITERATURE

	Teamwork	Confidence	Ambiguity handling	Persistence	Coordination/Cooperation	Self-awareness	Creativity	Problem solving	Communication
Hambrusch et al. [34]	√								
Barr et al. [2]	√	√	√	√					
Kazimoglu et al. [35]					√				
Vergara et al. [36]	√					√			
Korkmaz et al. [10]					√		√	√	
Allsop [37]				√	√		√	√	√

### III. RELATED WORKS

Korkmaz et al. [10] developed the ‘Computational Thinking Scale’ (CST), an instrument comprising 29 five-point Likert type items. The CST, which was tested on undergraduate students, assesses algorithmic thinking, critical thinking, creativity, problem solving, and cooperation. In a more recent study, a scale has been developed to assess high school students’ computational thinking skill [11]. In the same way as [10], this study also develops the scale based on ISTE (2015)’s definition of computational thinking skill. The scale takes in five skills, i.e. problem solving, algorithmic thinking, critical thinking, cooperative learning, and creative thinking as the initial factors. Subsequent to validity and reliability examinations, the resulting scale consists of five-point Likert scale of 42 items categorized into four factors, i.e. problem solving, cooperative learning and critical thinking, creative thinking, and algorithmic thinking.

The aforementioned instruments evaluate some soft-skills relevant to CT (creativity, problem-solving, and cooperation); however, both fail to consider the abstraction, decomposition, pattern recognition, and generalization. In contrast, abstraction is a basic tool of reasoning in CT [38]. That is, abstraction allows one to make simpler the large and complex problems [39]. In the same way, the ability to recognize patterns and generalizing solutions is an invaluable skill for computer scientists [18]. Decomposition is needed to break-down a problem into smaller, simpler and more manageable sub-problems [39]. That is, these skills are essential in CT.

This work proposes the Hi-ACT, a CT proficiency assessment instrument that takes both skills and attitudes into consideration. The constructs measured are elaborated in detail in the next section.

### IV. DEFINING HI-ACT

There is still little unanimity on CT definition, as [18], [40] inferred. Besides, its underlying skills are still being debated and redefined. Notwithstanding, to develop the Hi-ACT, we define CT as the thought process of formulating solutions to a problem that entails some skills and attitudes. The term ‘skills’ refers to computer science-based concepts used in CT, whereby this work draws on the work of [30], [32] to define CT core skills, including abstraction, algorithmic thinking, decomposition, debugging, evaluation, and generalization. The

term ‘attitudes’ refers to soft-skills. Soft-skills are personal specific skills which include attitudes, character traits, and behaviors [41]. In this work, the attitudes were drawn from the Computer Science Curricula 2013, i.e. problem solving, ambiguity tolerance, teamwork, communication, and personal attributes [42], and the operational definition of CT attitudes previously stated by [2]. This collection of work on attitudes was synthesized into three categories: problem solving, teamwork, and communication.

Furthermore, this work suggests one additional element, i.e. spiritual intelligence, to be included as one of the attitudes (soft-skills) of CT. The justification for such inclusion is: spiritual intelligence comprising a set of abilities that encourage people’s ability to solve problems, achieve goals, and enhance decision-making capability [43]. In this way, spiritual intelligence might be beneficial to CT as a way of thinking about solving problems. Moreover, there are some attitudes that demonstrate spiritual intelligence, i.e., self-awareness, creative reasoning, integrity, and asking ‘why’ questions, which are found helpful when confronting challenging problems, including artificial intelligence problems [44]. Hence, including spiritual intelligence would be beneficial in CT-based problem solving process.

In summary, this work develops an instrument, the Hi-ACT, which evaluates undergraduates’ perceptions of their CT competency. Based on the following literature [2], [28], [45]–[48], [29]–[32], [35], [39], [43], [44], the authors define each associated CT skills included in the Hi-ACT.

- Abstraction: the ability to simplify a problem by removing unnecessary details or information, then create a representation of the solution.
- Algorithmic Thinking: the ability to thinking algorithmically in formulating the instructions (procedure) through logical thinking to solve a problem.
- Decomposition: the ability to simplify a problem by dividing it into smaller, simpler, and easier to manage sub-problems.
- Debugging: the ability to identify and remove errors in the designed solutions (the algorithm).

- Evaluation: the ability to assess the solution's correctness, performance, resource usage, and the action of refining to improve the solution's quality.
- Generalization: the ability to identify similar patterns between the problems and generalizing solutions of previous problems to similar ones.
- Problem solving: the characters applicable to problem solving process, including self-confidence, persistence, ambiguity handling, and willingness to solve the problem.
- Teamwork: the ability to work in a team.
- Communication: the ability to exchange information and knowledge, by means of verbal and non-verbal, within the member of teamwork.
- Spiritual intelligence: copes with the ability to use spiritual abilities, including self-awareness, integrity, and creative reasoning, in enhancing an individual's personal characters to facilitate problem-solving process.

## V. METHODOLOGY

### A. Hi-ACT Initial Instrument

The Hi-ACT was firstly designed with 172 7-point Likert scale candidate items. These items address one of the sub-construct presented in Table III. Sub-construct is construct categories among the candidate items that are defined to ensure the items' convergent validity. Further, the first version underwent a content validation process through experts' judgment in a three-round Fuzzy Delphi Study, as reported in [49]. As a result, the initial Hi-ACT comprising 155 items was ready for validity and reliability assessment.

### B. Participants

The initial Hi-ACT was administered on a total sample of 713 undergraduate students, from STEM and non-STEM major of specializations. The participants were recruited from different departments (Computer Science, Economics, Social Sciences and Humanities, Design, Linguistic, Natural Sciences, Health, Engineering, Law, Medicine, and Education), from two universities located in two different cities in Indonesia and one university located in Malaysia. After removing the surveys that have not been completely filled in, the final usable sample size is 548. Prior to data collection, universities' approval was obtained. All participants were notified of their voluntary participation, anonymity and confidentiality were assured. The percentage of participants in term of gender was equal, 274 (50%) were male, and 274 (50%) were female. Regarding the major of specialization, from the total sample, 363 (66%) were registered as STEM-based.

### C. Data Analysis

This work aimed at refining the initial Hi-ACT by examining its validity and reliability. To do so, the structural equation modeling with partial least squares (PLS-SEM) was chosen. This choice was made for two reasons. First, factor analysis is a common statistical method for conceptualizing the constructs when refining a new instrument [50]. Exploratory

factor analysis is specifically intended to refine a set of items in a new instrument. In that regard, as argued by Hair, Hult, Ringer and Sarstedt [51], PLS-SEM is mainly used to develop theories in an exploratory study.

Second, PLS-SEM is suitable for a complex model [52]. Based on the literature analyzed in this work and the result of content validation, Hi-ACT comes up as a multi-dimension construct, i.e. the constructs and sub-constructs described in Table III. Thus, Hi-ACT was modeled as reflective-formative higher-order constructs, as shown in Fig. 2. This model comprises of 29 first-order constructs, i.e. the sub-constructs (AR, AC, ATPr, and so forth) and ten second-order constructs, i.e. the constructs (Abstraction, Algorithmic Thinking, Decomposition, and so forth). Finally, the second-order constructs are formative to the Hi-ACT construct. Each of the 155 items in the initial Hi-ACT was modeled as a reflective indicator of one of the 29 first-order constructs.

TABLE III. CT SKILLS AND ATTITUDES IN HI-ACT

Constructs	Sub-construct
Abstraction	Remove unnecessary detail (AR)
	Choose the right model (AC)
Algorithmic Thinking	Procedural thinking (ATPr)
	Sequence action (ATS)
	Conditional (ATC)
	Repetition (ATR)
	Parallelism (ATPa)
	Logical thinking (ATL)
Decomposition	Divide and conquer (DD)
	Modularizing (DM)
Debugging	Debugging (DE)
Evaluation	Performance evaluation (EP)
	Iterative refinement (EI)
	Optimizing (EO)
Generalization	Pattern recognition (GP)
	Reuse (GU)
	Remix (GM)
Problem solving	Confidence (PSC)
	Persistent (PSP)
	Ambiguity handling (PSA)
	Willingness (PSW)
Teamwork	Cooperation (TCp)
	Coordination (TCd)
	Participation (TP)
	Conflict management (TCM)
Communication	Communication (COM)
Spiritual intelligence	Self-awareness (SIS)
	Integrity (SII)
	Creative reasoning (SIC)

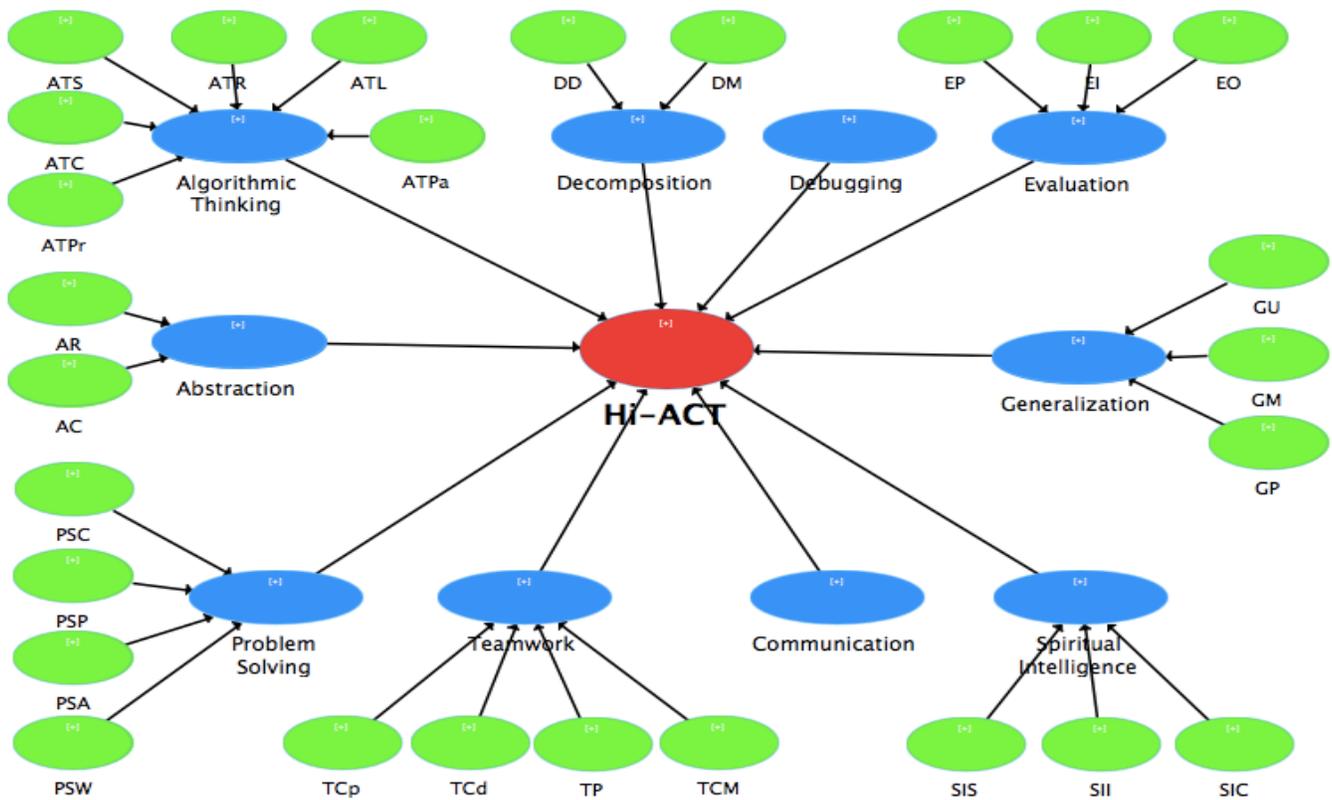


Fig. 2. The Hi-ACT.

In order to evaluate the instrument validity and reliability, the first-order constructs were evaluated. Evaluating reflective first-order constructs involves the examination of reliability and construct validity (convergent validity), which we deemed acceptable to analyze the result of new instrument pilot study. By referring to Hair, Hult, Ringer and Sarstedt [51], the following analyses were conducted:

- Internal consistency reliability as an evaluation of reliability. The internal consistency reliability was assessed by composite reliability (CR). It is desirable to have value within the range of 0.7 to 0.9, and it should not exceed 0.9.
- Convergent validity was assessed based on two criteria, i.e. items outer loadings and constructs' average variance extracted (AVE). Item's outer loading should be  $\geq 0.7$  and the AVE to be  $\geq 0.5$ . When the AVE does not meet the required threshold, the item with the smallest loadings should be removed.

## VI. RESULT

Table IV presents the internal consistency and convergent validity results. In the first run, the constructs CR value ranged from 0.7 to 0.93 exceed the threshold of 0.7. However, the TCp, SII, and SIC constructs' CR are higher than 0.9. CR value above 0.9 indicates that all the indicators measuring the same phenomenon, which is not a valid measure of the construct, and therefore is not desirable [51]. The second test was convergent validity. The loadings of all items ranged from 0.55 to 0.85, while the AVE ranged from 0.43 to 0.66. The AR, AC, ATS, ATR, DD, COM, SIS, SII, and SIC constructs' AVEs fall short of the threshold value of 0.5, indicating that the conditions of convergent validity were not met. Accordingly, items with low loadings were eliminated.

In summary, the initial instrument was refined by removing 38 items to improve each particular construct's AVE, and three items to shrink the CR value of TCp constructs to 0.9. Hence, the total of items eliminated was 41. This increased the AVEs while keeping the CRs in the threshold (Table V), then subsequently support the internal consistency and convergent validity conditions.

TABLE IV. RELIABILITY AND CONVERGENT VALIDITY RESULTS

Construct	Item Code	Item	Loading	AVE	CR	Reason for Exclusion
AR	AR-1	Reducing complexity by removing unnecessary detail	0.61	0.45	0.83	*
	AR-2	Evaluate what is valuable information and what is not	0.69			
	AR-3	Filtering the information when developing solution	0.69			
	AR-4	Separate the important from the redundant information	0.71			
	AR-5	Add or remove details to clarify a problem	0.62			*
	AR-6	Find appropriate level of detail to define and solve a problem	0.67			*
AC	AC-1	Understand that a model depicts general idea of the problem	0.72	0.48	0.74	
	AC-2	Create a model to solve problem	0.72			
	AC-3	Choosing a way to represent an artifact, allow it to be manipulated in useful ways	0.64			*
ATS	ATS-1	Formulate instructions to achieve a desired effect	0.65	0.46	0.81	*
	ATS-2	Create a set of precise steps to solve a problem	0.72			
	ATS-3	Elaborate particular activity or task as a series of individual steps or instructions	0.69			
	ATS-4	Explicitly wording the steps to solve problem	0.64			*
	ATS-5	Put instructions in the correct sequence	0.67			*
ATPr	ATPr-1	Select and execute appropriate steps to solve problem	0.76	0.54	0.82	
	ATPr-2	Identify the steps required to solve a problem	0.75			
	ATPr-3	Identify the sequence of steps including possible decisions and branching	0.77			
	ATPr-4	Understand normal and exceptional behaviors of a solution	0.65			*
ATC	ATC-1	Make decisions based on certain conditions	0.71	0.54	0.78	
	ATC-2	Think of possibility of different procedures for a problem	0.77			
	ATC-3	Produce many options while thinking of the possible solution regarding a problem	0.73			
ATR	ATR-1	Implement the same design plan for a specified number of times	0.56	0.43	0.75	*
	ATR-3	Repeat design processes to refine solutions until the ideal result is achieved	0.65			*
	ATR-4	Identify all possibilities of procedures that can be executed more than once	0.70			
	ATR-5	Decide, based on certain condition, when to execute a procedure and when to stop	0.70			
	ATPa	ATPa-1	Running different sequences of instructions at the same time			0.76
ATPa-2	Dividing up resources and task in such a way to be processed in parallel	0.85				
ATL	ATL-1	Explain why something happens	0.73	0.54	0.87	
	ATL-2	Infer a conclusion based on existing knowledge	0.76			
	ATL-3	Explain how a conclusion is drawn	0.72			
	ATL-4	Provide the reason for my own thinking	0.72			
	ATL-5	Using existing knowledge to make reliable predictions	0.73			
	ATL-6	Elaborate logical connections between cause and effect	0.73			
DD	DD-1	Breaking apart problem into smaller subproblems to make it easier to solve	0.70	0.48	0.79	
	DD-2	Breaking down a problem into simpler version enables the same problem to be solved in the same way	0.72			
	DD-3	Apply the order of mathematical operations properly	0.62			*
	DD-4	Do classification	0.72			
DM	DM-1	Combine smaller parts to produce something larger	0.78	0.61	0.76	
	DM-2	Develop a solution by assembling together the smaller parts	0.77			
DE	DE-1	Think of anticipation plan for a problem	0.78	0.6	0.75	
	DE-2	Recognize problem when procedures do not correspond to solutions	0.77			
EP	EP-1	Determine whether the procedures in a solution is complete to solve the problem	0.75	0.53	0.85	
	EP-2	Assess whether the solution is suitable for solving the problem	0.73			
	EP-3	Assess whether the solution does the right thing	0.72			
	EP-4	Compare the performance of different procedures that solve the same problem	0.74			

	EP-5	Assess whether the solution is easy for people to use	0.68			*
EI	EI-1	Refine the solution procedure to improve its precision	0.78	0.58	0.8	
	EI-2	Evaluate solution against the success criteria	0.74			
	EI-3	Adjust the design and implementation of a solution when necessary	0.75			
EO	EO-1	Analyze the solution for efficient use of resources	0.79	0.56	0.84	
	EO-2	Develop a solution that can utilize the available resources	0.79			
	EO-3	Adapt the solution that can utilize the available resources	0.76			
	EO-4	After the problem solved, I analyze what went right and what went wrong	0.65			*
GU	GU-1	Applying an existing solution in a given problem to cover more possibilities	0.73	0.5	0.8	
	GU-2	Use sequence of instructions previously employed to solve a new problem	0.74			
	GU-3	Transfer ideas and solutions from one problem to another	0.70			
	GU-4	Building on other people's work	0.66			*
GM	GM-1	Embed other's work into my own work in a meaningful way	0.72	0.54	0.82	
	GM-2	Efficient in researching relevant information	0.75			
	GM-3	Constructively builds on contributions of others & integrates my own works with others'	0.75			
	GM-4	Combine and build on the ideas of others	0.72			
GP	GP-1	Identify patterns, similarities, and connections between prior and current problems	0.76	0.54	0.7	
	GP-2	Solve similar problems with the same set of steps	0.72			
PSC	PSC-1	I am a good problem solver	0.73	0.59	0.89	
	PSC-4	Confident to solve most problems	0.77			
	PSC-5	Given enough time, I believe I can solve most problems that confront me	0.77			
	PSC-7	Can solve new and difficult problems	0.79			
	PSC-8	Have a systematic method for comparing alternatives and making decisions	0.80			
	PSC-9	When I make plans to solve a problem, I am almost certain that I can make them work	0.78			
PSP	PSP-1	Can work a long time on a problem	0.68	0.55	0.88	*
	PSP-2	Keep working on a problem until I get the right answer	0.73			
	PSP-3	Keep trying even when a task becomes difficult	0.76			
	PSP-4	When a solution to a problem was unsuccessful, I will examine why it didn't work	0.75			
	PSP-5	When I'm confronted with a complex problem, I develop a strategy to collect information so that I can define exactly what the problem is	0.76			
	PSP-6	When my first effort to solve a problem fail, I still have certainty about my ability to handle the situation	0.76			
PSA	PSA-1	Anticipate impact of change and direct myself and others in smoothly shifting gears	0.74	0.52	0.89	
	PSA-3	Can guide others to cope with ambiguous situation effectively	0.7			
	PSA-4	Rise to challenge, accepting risk and uncertainty as normal	0.72			
	PSA-5	Remain calm and focus during time of change	0.74			
	PSA-6	Willing and open to change	0.69			*
	PSA-7	Adaptable with the unknown	0.68			*
	PSA-8	Have no problems with demonstrating the solution of a problem in my mind	0.71			
	PSA-10	Apply the solution I plan respectively	0.78			
PSW	PSW-1	Like to try to solve problems	0.84	0.66	0.85	
	PSW-2	It is fun to try to solve problems	0.79			
	PSW-3	Will try to solve almost any problems	0.81			
TCp	TCp-1	Enjoy working together with other	0.7	0.54	0.93	/
	TCp-2	Share the power with others	0.78			
	TCp-3	Understand that there are shared knowledge and skills between team members	0.79			
	TCp-4	Create a cooperative atmosphere among the members when addressing problems	0.79			
	TCp-5	Listen to and consider other's opinions	0.73			
	TCp-6	Willing to ask others for help	0.69			/
	TCp-7	Trust other team members	0.6			/

	TCp-8	Set aside differences when work with others to achieve a common goal	0.71			
	TCp-9	Like to experience cooperative learning together with team members	0.76			
	TCp-10	In cooperative learning, I think I attain more successful results	0.73			
	TCp-11	Solve problems related to group project in cooperative learning	0.76			
	TCp-12	More ideas occur in cooperative learning	0.76			
TCd	TCd-2	Work together harmoniously	0.77	0.59	0.81	
	TCd-3	Communicates actively and constructively	0.78			
	TCd-4	Acknowledge contribution of others	0.75			
TP	TP-1	Stay focus on the task during group work	0.75	0.58	0.87	
	TP-2	Fulfill the role assigned by the group	0.75			
	TP-3	Participate actively and accept a fair share of group work	0.81			
	TP-4	Work skilfully on the task assigned to me and complete them on time	0.72			
	TP-5	Share responsibilities for the team's success or failure	0.77			
TCM	TCM-1	Respond to and manage direct/indirect conflict constructively and effectively	0.71	0.51	0.86	
	TCM-2	Fully accept each other's strengths and weaknesses	0.73			
	TCM-3	Try to achieve harmony by avoiding conflict	0.7			
	TCM-4	Takes criticism in a friendly way	0.73			
	TCM-5	Avoid using put-down or blaming others	0.68			*
	TCM-6	Accept compromise to deal with conflict	0.74			
COM	COM-1	Like to share feelings, ideas, or opinions with others	0.66	0.48	0.88	*
	COM-2	Speak clearly with acceptable vocabulary	0.73			
	COM-3	Use a variety of communication means (written message, e-mail, phone, informal discussion)	0.68			
	COM-4	Limit length of comments so others can talk	0.64			*
	COM-5	Listen to everyone and respect their view	0.68			
	COM-6	Contribute appropriately in healthy debate	0.74			
	COM-7	Ensure consistency between words and tone	0.72			
	COM-8	Ensure consistency between facial expression and body languages	0.68			*
SIS	SIS-1	Aware of one's abilities and weaknesses	0.63	0.46	0.89	*
	SIS-2	Live with self-respect	0.66			*
	SIS-3	Satisfied with who I am	0.55			*
	SIS-4	Do any work with self-confidence	0.66			*
	SIS-5	Can decide my own goal	0.71			
	SIS-6	Consistently look for and try to discover my blind spots	0.68			*
	SIS-7	In negotiating, I try to see things from other's perspective even when I disagree	0.68			
	SIS-8	During an activity or conversation, I monitor and notice my thoughts and emotions	0.73			*
	SIS-9	My actions are aligned with my true essential nature	0.74			
	SIS-10	Aware of one's inner truth, what one know inside to be true	0.73			
SII	SII-1	Proud of one's country culture	0.58	0.49	0.93	*
	SII-2	Believe that character is one's real strength	0.68			*
	SII-3	Be aware of my own values and beliefs	0.74			
	SII-4	Keep the promises given to others	0.73			
	SII-5	My actions are aligned with my values	0.76			
	SII-6	I accept myself as I am with all my problems and limitations	0.73			
	SII-7	Know how to be myself when interacting with others	0.74			
	SII-8	Help and support others	0.69			*
	SII-9	Respect and trust others	0.66			*
	SII-10	Being open and honest with others	0.64			*
	SII-11	Put one's consciousness in a positive direction	0.72			
	SII-12	Live one's values in relationship with others	0.72			

	SII-13	Act with honesty and truthfulness	0.67			*
	SII-14	Keep working diligently even when no one is watching	0.68			
SIC	SIC-1	In solving problems, I draw on my own ability to hold, accept and go beyond paradoxes	0.73	0.47	0.91	
	SIC-2	Can integrate the seemingly contradictory points of view	0.76			
	SIC-3	Find it challenging to find out what the truth is	0.70			
	SIC-4	Can think of an answer to a problem, even though at first apparently no solution	0.70			
	SIC-5	Can offer new ways of viewing to a problem	0.71			
	SIC-6	Can find an unusual way to solve a problem	0.62			*
	SIC-7	Eager to know many things	0.69			
	SIC-8	Willing to change my mind and try something else	0.65			*
	SIC-9	Willing to admit when I made a wrong decision	0.66			*
	SIC-10	Can improve the original idea	0.64			*
	SIC-11	Can express one's ideas well	0.67			*

\* Remove to refine the construct's AVE  
/ Remove to refine the construct's internal consistency reliability

TABLE V. RELIABILITY AND CONVERGENT VALIDITY RESULTS AFTER ITEMS REMOVAL

Construct	Remaining Items	Loading	AVE	CR
AR	3	0.72 – 0.74	0.53	0.77
AC	2	0.72 – 0.75	0.54	0.70
ATS	2	0.68 – 0.74	0.50	0.67
ATPr	3	0.76 – 0.78	0.59	0.81
ATC	3	0.70 – 0.78	0.54	0.78
ATR	2	0.69 – 0.72	0.50	0.67
ATPa	2	0.75 – 0.86	0.65	0.79
ATL	6	0.72 – 0.76	0.54	0.87
DD	3	0.71 – 0.74	0.53	0.77
DM	2	0.77 – 0.79	0.61	0.76
DE	2	0.77 – 0.78	0.60	0.75
EP	4	0.72 – 0.76	0.54	0.83
EI	3	0.74 – 0.79	0.58	0.80
EO	3	0.78 – 0.81	0.64	0.84
GU	3	0.71 – 0.77	0.56	0.79
GM	4	0.71 – 0.75	0.54	0.82
GP	2	0.71 – 0.77	0.54	0.70
PSC	6	0.73 – 0.80	0.59	0.90
PSP	5	0.72 – 0.79	0.58	0.87
PSA	6	0.70 – 0.77	0.53	0.87
PSW	3	0.79 – 0.84	0.66	0.85
TCp	9	0.71 – 0.79	0.57	0.90
TCd	3	0.75 – 0.78	0.59	0.81
TP	5	0.72 – 0.8	0.58	0.87
TCM	5	0.69 – 0.73	0.51	0.84
COM	5	0.68 – 0.74	0.5	0.83
SIS	4	0.70 – 0.75	0.54	0.82
SII	8	0.68 – 0.76	0.53	0.90
SIC	6	0.69 – 0.77	0.52	0.87

## VII. DISCUSSION

This pilot study yielded preliminary proof of Hi-ACT's potential psychometric properties, a scales aimed at assessing undergraduate CT skills more comprehensively, by incorporating both skills and attitudes. A total of ten constructs and 114 items were extracted for Hi-ACT. Within this frame, the factor loadings for all items ranged from 0.68 to 0.86. These values indicating that the items of each specific construct have much in common, and they are contributing to measuring each associated sub-construct. The convergent validity of the construct level (AVE) was confirmed with values were ranged from 0.5 to 0.66, satisfying the required threshold of 0.5. The internal consistency reliability was maintained in an acceptable range. Within the range from 0.67 to 0.9, the CR values exceeding 0.7 were obtained for most of the sub-constructs. Thus, indicating that high internal consistency was achieved.

Six sub-constructs, i.e., ATS, ATR, TCM, COM, SII, and SIC, have items with factor loadings less than the threshold value of 0.7. Low factor loadings might contribute to low CR and AVE. Particularly for ATS and ATR sub-constructs, the CR values (0.67) were slightly lower than other sub-constructs. The CR value of 0.67, indicating that the items only have shared common variance of 45%, which implies that the items in each construct are slightly weak to measure the construct. It could be that these two sub-constructs have very few items compared to other sub-constructs. Each sub-constructs has two items, and one of them has factor loading less than 0.7; ATC3 (0.68) and ATR4 (0.69), which leads to slightly low item reliability. Nevertheless, the CR value above 0.6 is considered acceptable in an exploratory study [51]. Moreover, the AVE of both sub-constructs achieved the value of 0.5. This indicates that, on average, each sub-construct accounts for a minimum of 50% of the variance of its items. Thus, the validity of the items and the sub-construct is indisputable. The COM sub-construct also holds two items with factor loadings lower than 0.7, i.e. COM-3 (0.68) and COM-5 (0.67). However, removing one of these items led to a fall in sub-construct's convergent validity (decreasing the AVE). Also, COM has other strength statistics, i.e. CR value of 0.83. Accordingly, the items were retained. For the same reason, the items with factor loadings below 0.7 in TCM, SII, and SIC sub-constructs were retained.

## VIII. CONCLUSION

The Hi-ACT which that evaluates undergraduates' perceptions of their CT competency was developed. A pilot study was carried out to refine the initial instrument. Based on the responses of 548 university undergraduates to 155 items, an instrument comprising 114 items was established. The findings of statistical test of internal consistency and convergent validity reveal that the Hi-ACT in its pilot form is valid and reliable to measure university undergraduates' CT competency. In future studies, we plan to proceed with further instrument evaluation to provide further evidence of construct validity and discriminant validity.

Furthermore, the Hi-ACT makes a notable contribution to CT literature. It extends the CT assessment study by verifying ten primary constructs and 29 sub-constructs, which delineate the skills and attitudes applicable in CT-based problem solving

process. These CT concepts did not comprehensively address in most previous CT assessment studies. Accordingly, findings of this work bring forth comprehensiveness to CT theoretical work, specifically in undergraduate context. It also results in a set of indicators that useful in measuring CT competency holistically.

## REFERENCES

- [1] J. M. Wing, "Computational thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–35, 2006.
- [2] D. Barr, J. Harrison, and L. Conery, "Computational thinking: A digital age skill for everyone," *Learn. Lead. with Technol.*, vol. 38, no. 6, pp. 20–23, 2011.
- [3] M. G. Voskoglou and S. Buckley, "Problem solving and computers in a learning environment," *Egypt. Comput. Sci. J.*, vol. 36, no. 4, pp. 28–46, 2012.
- [4] K. Yevseyeva and M. Towhidnejad, "Work in progress: Teaching computational thinking in middle and high school," in *2012 Frontiers in Education Conference Proceedings*, 2012, pp. 1–2.
- [5] World Economic Forum, "The future of jobs report 2018," Geneva, Switzerland, 2018.
- [6] N. Talib, S. F. M. Yasin, and K. M. Mohd, "Teaching and Learning Computer Programming Using Gamification and Observation through Action Research," *Int. J. Acad. Res. Progress. Educ. Dev.*, vol. 6, no. 3, pp. 1–11, 2017.
- [7] F. Kalelioglu, Y. Gülbahar, and V. Kukul, "A Framework for Computational Thinking Based on a Systematic Research Review," *Balt. J. Mod. Comput.*, vol. 4, no. 3, pp. 583–596, 2016.
- [8] L. Shanmugam, S. F. Yassin, and F. Khalid, "Enhancing Students' Motivation to Learn Computational Thinking through Mobile Application Development Module (M-CT)," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 5, pp. 1293–1303, 2019.
- [9] C. K. Chang, Y. T. Tsai, and Y. L. Chin, "A Visualization Tool to Support Analyzing and Evaluating Scratch Projects," in *Proceedings - 6th IIAI International Congress on Advanced Applied Informatics 2017*, 2017, pp. 498–502.
- [10] Ö. Korkmaz, R. Cakir, and M. Y. Ozden, "A validity and reliability study of the computational thinking scales (CTS)," *Comput. Human Behav.*, vol. 72, pp. 558–569, 2017.
- [11] M. Yagci, "A valid and reliable tool for examining computational thinking skills," *Educ. Inf. Technol.*, vol. 24, no. 1, pp. 929–951, 2018.
- [12] M. Román-gonzález, J. C. Perez-Gonzalez, and C. Jimenez-Fernandez, "Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test," *Comput. Human Behav.*, vol. 72, no. July 2017, pp. 678–691, 2016.
- [13] M. Bower, L. N. Wood, J. W. M. Lai, C. Howe, and R. Lister, "Improving the computational thinking pedagogical capabilities of school teachers," *Aust. J. Teach. Educ.*, vol. 42, no. 3, pp. 53–72, 2017.
- [14] S. Gretter and A. Yadav, "Computational thinking and media & information literacy: An integrated approach to teaching twenty-first century skills," *TechTrends*, vol. 60, no. 5, pp. 510–516, 2016.
- [15] T. Rahayu and K. Osman, "Knowledge Level and Self-confidence On The Computational Thinking Skills Among Science Teacher Candidates," *J. Ilm. Pendidik. Fis.*, vol. 08, no. April, pp. 117–126, 2019.
- [16] S. Senin and N. M. Nasri, "Teachers' Concern towards Applying Computational Thinking Skills in Teaching and Learning Teachers' Concern towards Applying Computational Thinking Skills in Teaching and Learning," *Int. J. Acad. Res. Bus. Soc. Sci.*, vol. 9, no. 1, pp. 296–310, 2019.
- [17] J. Walden, M. Doyle, R. Gams, and Z. Hart, "An informatics perspective on computational thinking," in *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, 2013, pp. 4–9.
- [18] L. Gouws, K. Bradshaw, and P. Wentworth, "First year student performance in a test for computational thinking," in *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, 2013, pp. 271–277.

- [19] M. Csernoch, P. Biró, J. Máth, and K. Abari, "Testing algorithmic skills in traditional and non-traditional programming environments," *Informatics Educ.*, vol. 14, no. 2, pp. 175–197, 2015.
- [20] D. E. Sondakh, K. Osman, and S. Zainudin, "Holistic Assessment of Computational Thinking for Undergraduate: Reliability and Convergent Validity," in *Proceedings of The 11th International Conference on Education Technology and Computers*, 2019, pp. 241–245.
- [21] M. Guzdial, "Education paving the way for computational thinking," *Commun. ACM*, vol. 51, no. 8, p. 25, 2008.
- [22] P. J. Denning, "Computing is a natural science," *Commun. ACM*, vol. 50, no. 7, p. 13, 2007.
- [23] B. Czerkawski and E. Lyman III, "Exploring issues about computational thinking in higher education," *TechTrends*, vol. 59, no. 2, pp. 57–65, 2015.
- [24] A. Yadav, J. Good, J. Voogt, and P. Fisser, "Computational thinking as an emerging competence domain," in *Competence-based Vocational and Professional Education Technical and Vocational Education and Training: Issues, Concerns and Prospects*, M. Mulder, Ed. Springer, 2017, pp. 1051–1067.
- [25] A. V. Aho, "Computation and computational thinking," *Comput. J.*, vol. 55, no. 7, pp. 833–835, 2012.
- [26] P. J. Denning, "The profession of IT beyond computational thinking," *Commun. ACM*, vol. 52, no. 6, p. 28, 2009.
- [27] S. Grover and R. Pea, "Computational thinking in K-12: A review of the state of the field," *Educ. Res.*, vol. 42, no. 1, pp. 38–43, 2013.
- [28] J. M. Wing, "Research notebook: Computational thinking—what and why?," *The Link Magazine*, 2011.
- [29] J. M. Wing, "Computational thinking and thinking about computing," *Philos. Trans. R. Soc.*, vol. 366, pp. 3717–3725, 2008.
- [30] C. C. Selby and J. Woollard, "Computational thinking: the developing definition," 2013.
- [31] C. Angeli et al., "A K-6 computational thinking curriculum framework: implications for teacher knowledge," *Educ. Technol. Soc.*, vol. 19, no. 3, pp. 47–57, 2016.
- [32] S. Bocconi, A. Chiocciariello, G. Dettori, A. Ferrari, and K. Engelhardt, "Developing Computational Thinking in Compulsory Education: Implications for policy and practice," 2016.
- [33] J. Mueller, D. Beckett, E. Hennessey, and H. Shodiev, "Assessing computational thinking across the curriculum," in *Emerging Research, Practice, and Policy on Computational Thinking*, P. J. Rich and C. B. Hodges, Eds. Cham, Switzerland: Springer International Publishing, 2017, pp. 251–267.
- [34] S. Hambruch, C. Hoffmann, J. T. Korb, M. Haugan, and A. L. Hosking, "A multidisciplinary approach towards computational thinking for science majors," in *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, 2009, vol. 41, pp. 183–187.
- [35] C. Kazimoglu, M. Kiernan, L. Bacon, and L. MacKinnon, "Learning programming at the computational thinking level via digital game-play," *Procedia Comput. Sci.*, vol. 9, pp. 522–531, 2012.
- [36] C. E. Vergara et al., "Towards a framework for assessing computational competencies for engineering undergraduate students," in *122nd ASEE Annual Conference & Exposition*, 2015, pp. 26.1589.1-26.1589.14.
- [37] Y. Allsop, "Assessing computational thinking process using a multiple evaluation approach," *Int. J. Child-Computer Interact.*, vol. 19, pp. 30–55, 2018.
- [38] C. Varela, C. Rebollar, O. García, E. Bravo, and J. Bilbao, "Skills in computational thinking of engineering students of the first school year," *Heliyon*, vol. 5, no. 11, 2019.
- [39] P. Curzon, M. Dorling, T. Ng, C. Selby, and J. Woollard, "Developing computational thinking in the classroom: A framework." p. 5, 2014.
- [40] F. Kalelioglu, Y. Gulbahar, and V. Kukul, "A framework for computational thinking based on a systematic research review," *Balt. J. Mod. Comput.*, vol. 4, no. 3, pp. 583–596, 2016.
- [41] M. M. Robles, "Executive perceptions of the top 10 soft skills needed in today's workplace," *Bus. Commun. Q.*, vol. 75, no. 4, pp. 453–465, 2012.
- [42] ACM & IEEE Computer Society, "Computer Science Curricula 2013 Curriculum guidelines for undergraduate degree programs in Computer Science," 2013.
- [43] D. A. Sisk, "Spiritual intelligence: developing higher consciousness revisited," *Gift. Educ. Int.*, vol. 32, no. 3, pp. 194–208, 2016.
- [44] M. Kadkhoda and H. Jahanic, "Problem-solving capacities of spiritual intelligence for artificial intelligence," *Procedia - Soc. Behav. Sci.*, vol. 32, no. 2012, pp. 170–175, 2012.
- [45] J. L'Heureux, D. Boisvert, R. Cohen, and K. Sanghera, "IT problem solving: an implementation of computational thinking in information technology," in *Proceedings of the 13th Annual Conference on Information Technology Education*, 2012, pp. 183–188.
- [46] N. Kourmoussi, V. Xythali, M. Theologitou, and V. Koutras, "Validity and reliability of the problem solving inventory (PSI) in a nationwide sample of Greek educators," *Soc. Sci.*, vol. 5, no. 2, p. 25, 2016.
- [47] E. Britton, N. Simper, A. Leger, and J. Stephenson, "Assessing teamwork in undergraduate education: a measurement tool to evaluate individual teamwork skills," *Assess. Eval. High. Educ.*, vol. 42, no. 3, pp. 378–397, 2015.
- [48] P. S. Strom and R. D. Strom, "Teamwork skills assessment for cooperative learning," *Educ. Res. Eval.*, vol. 17, no. 4, pp. 233–251, 2011.
- [49] D. E. Sondakh, K. Osman, and S. Zainudin, "A Proposal for Holistic Assessment of Computational Thinking for Undergraduate: Content Validity," *Eur. J. Educ. Res.*, vol. 9, no. 1, pp. 33–50, 2020.
- [50] M. Norris and L. Lecavalier, "Evaluating the use of exploratory factor analysis in developmental disability psychological research," *J. Autism Dev. Disord.*, vol. 40, no. 1, pp. 8–20, 2010.
- [51] J. F. Hair, G. T. M. Hult, C. M. Ringle, and M. Sarstedt, *A primer on partial least squares structural equation modeling (PLS-SEM)*, 2nd ed. Sage Publications, Inc., 2017.
- [52] N. K. Avkiran, "An in-depth discussion and illustration of partial least squares structural equation modeling in health care," *Health Care Manag. Sci.*, vol. 21, no. 3, pp. 401–408, 2018.