

计算思维培养之路还有多远?

——基于计算思维测评视角

郁晓华 王美玲

(华东师范大学 教育学部教育信息技术学系,上海 200062)

[摘要] 随着人工智能时代人才竞争的加剧,K-12 阶段计算思维的培养成为重要的抓手。其中,通过测评准确把握现阶段 K-12 实践开展情况可推动计算思维培养更具针对性。本文先从理论层面建构了包含培养内容、教学方式和测评方法三个维度的 K-12 计算思维培养框架;然后采用元分析法和内容分析法分析 41 个测评实践案例,揭示和讨论计算思维培养理论与实践之间的差距,展示未来可行的研究空间。分析结果表明,计算概念、计算实践、计算观念等计算思维培养内容在实践中都有体现,编程教学、跨学科课堂和独立学科三类教学形式也有一定的实践基础,题目测试法、编程测试法、作品分析法、调查法、图文分析法和观察访谈法等六种测评方式得到了使用和发展;但实践中仍存在培养内容不全面、教学形式相对单一以及多元评价意识不足等问题。文章最后提出全面掌握计算思维内容体系,关注空白研究维度,增加对非认知层面的关注,注重指标间的内部关联;尝试计算科学和编程教育范畴外的课程载体,增强个体发展的比重;研究各种测评方法的综合应用,加强对形成性评价的关注,以全面展现计算思维的发展。

[关键词] 计算思维;K-12;人工智能;编程

[中图分类号] G449.7

[文献标识码] A

[文章编号] 1007-2179(2020)01-0060-12

一、引言

人工智能的新一轮兴起推动人类社会各领域向智能化转变。为确保人工智能时代的竞争地位,在中小学阶段设置人工智能课程成为提高全民人工智能整体认知和应用水平的重要保障措施(国务院,2017)。在这一新的形势下,计算思维因能够帮助学生理解计算的世界,有效地使用和创造技术,不仅被视为人工智能时代智慧型人才的必备素质(郑旭东,2018),更被视作读、写、算之外的又一项基本素

养(Wing,2006)。

尽管目前还没有公认的定义,但计算思维已深入人心,相关理论和实践不断涌现。在理论层面,已有研究较好地揭示了计算思维的内容和本质(Wing,2006),细化计算思维发展的过程和环节(ISTE & CSTA,2011a),同时将计算思维的概念具体化,从而贴近教学实际(Brennan & Resnick,2012;CAS,2015)。伴随着理论研究的百花齐放,实践层面也取得了不少进展。首先,计算思维发展重心向基础教育阶段转移的号召得到了世界各国的响应,

[收稿日期] 2019-09-19 **[修回日期]** 2019-12-08 **[DOI 编码]** 10.13966/j.cnki.kfjyyj.2020.01.007

[基金项目] 2019 年度教育部人文社会科学研究规划基金项目“5-15 岁儿童计算思维测评框架及方法研究”(19YJA880079)。

[作者简介] 郁晓华,博士,副教授,华东师范大学教育学部教育信息技术学系,研究方向:信息化教与学、学习技术系统设计与开发、学习分析与可视化、计算思维培养(xhyu@deit.ecnu.edu.cn);王美玲,华东师范大学教育学部教育信息技术学系研究生。

英国的计算(computing)课程(Brown et al., 2014)、美国的计算科学课程(K-12 Computer Science, 2016)等都直接或间接地在 K-12 实施。计算思维测评方面,柯尔克玛兹(Korkmaz et al., 2015)、罗曼·冈萨雷斯(Román-González et al., 2017b)等开发了用于评估学生计算思维的工具;沃纳(Werner et al., 2012)、奥尔索普(Allsop, 2019)尝试通过测试和评价证明计算思维的发展。但是,计算思维理论框架中的哪些思维和能力的发展得到了实践的有效验证?哪些尚未开展?未开展的原因是什么?教学方式以及测评方法的使用与理论间的对应关系如何?存在怎样的差距?现有实践结论能够为新理论与实践的开展提供怎样的启示和引导呢?

测评既是教学活动的组成环节,也是教学成效验证的重要依据。测评具有诊断性、针对性、客观性、严格性和发展性等特点。测评内容可以帮助我们准确把握计算思维培养的内容,测评结果又可客观地反映培养实践的真实效果,揭示可能的发展方向。本研究将测评作为切入视角,梳理现有 K-12 阶段计算思维测评实践案例,分析当前计算思维培养理论与实践之间的差距。

二、理论层面:K-12 计算思维培养框架

计算思维培养框架的建构首先要明确“教什么”。计算思维常被概述为解决问题的思维方式,实际上它包含从思维习惯到编程概念等一系列要素(Corradini et al., 2017)。因此,梳理归纳、统筹分类这些要素是指导培养活动开展的首要条件;其次,合适的课程载体和教学活动可确保计算技能和观念在课程中的传递;最后是计算思维“如何测”的问题。测评作为教学活动的最后环节,是深入持续开展培养活动的依据,也是评判培养成效的证据。计算思维实践成熟有赖于标准化的测评(Buffum et al., 2015)。

(一)培养内容

探索 K-12 计算思维培养内容实质上是对概念的深入解读。早期对计算思维概念的认识主要是抽象理解其普遍特征,现在,我们已能将这一混合概念分解为一系列定义明确且可衡量的技能、概念或实践(Weintrop et al., 2016)。佩珀特(Papert, 1972)最早指出计算是教育创新的基本要素,并提出计算

思维术语(Papert, 1980)。2006年,周以真规范化地提出计算思维的基本定义,认为计算思维是指运用计算机科学的基础概念进行问题求解、系统设计和人类行为理解等一系列思维活动(Wing, 2006)。但这一定义将计算思维融入课程以及如何观察学生的能力发展方面显得笼统而抽象(Deng et al., 2009),之后更多细化、可操作的定义相继被提出。2011年,美国教育技术协会与计算机科学教师协会指出,计算思维的特征包括:对问题的建构能用计算机及相关工具加以解决;能用逻辑的方式组织和分析数据;能以抽象(如模型和模拟)的方式表示数据;能用算法思维(即一系列有序步骤)自动化解决方案;能识别和分析可行的解决方案,实现步骤和资源的有效组合;能概括问题解决过程并迁移到其他各类问题解决中。他们还指出计算思维技能的提升依靠解决复杂问题的信心,处理棘手问题的恒心,对多种意见的包容心,处理开放式问题的能力及与他人沟通和一起工作以实现共同目标或解决问题的能力(ISTE & CSTA, 2011a)。此外,这两家协会还联合提出计算思维九大词汇:数据收集、数据分析、数据表示、问题分解、抽象、算法和程序、自动化、模拟、并行化(ISTE & CSTA, 2011b)。2012年,布伦南等(Brennan & Resnick, 2012)提出计算思维三维框架包括三个关键部分:计算概念(顺序、循环、事件、并行、条件、运算、数据)、计算实践(增加与迭代、测试与调试、再利用与再混合、抽象化与模块化)和计算观念(表达、连接、质疑)。2013年,塞尔比等(Selby & Woollard, 2013)提出计算思维应关注问题的解决过程,包含抽象、分解、算法设计、评价和概念化五个环节。2015年,英国学校计算工作小组(Computing at School, 简称 CAS)指出,计算思维涉及概念层面(抽象、评价、算法思维、分解、概念化)与操作层面(反应、编码、设计、分析、应用),是识别我们周边世界计算的过程,也是应用计算工具和技术理解、推演自然和人工系统的过程(CAS, 2015)。谷歌探索计算思维研究组(Exploring Computational Thinking, 简称 ECT)列出并重新提出了一系列计算思维概念,指出计算思维概念是与解决计算问题相关的心理过程和有形结果,包含抽象、算法设计、自动化、数据分析、数据收集、数据表示、分解、并行化、模式泛化、模式识别和模拟(ECT, 2017)。我国普通高中信息技

术课程标准(2017年版)指出,具备计算思维的学生,在信息活动中可采用计算机处理问题的方式界定问题、抽象建模、组织数据;通过整合资源并运用合理算法构建问题解决方案;总结计算机解决问题的过程方法并迁移到相关问题解决(教育部,2018)。

综上所述,数十年来,计算思维的内涵不断扩充完善,除了普遍认可的计算思维是解决问题的思维方式外,所有版本的定义都罗列了构成要素。因此,建立对计算思维的系统化认知,需要充分考虑已有研究,提炼出计算思维的核心与相关要素,还要梳理这些元素间的逻辑结构,构建层级关系。

(二)教学方式

计算思维在 K-12 阶段的培养活动表现为三种形式。一是编程教学,如美国发起的编程一小时活动(Code.Org,2013),旨在通过编程教学实现培养计算思维的核心目标。K-12 学校大多以计算科学/信息技术课堂为载体开展编程教学活动。二是将计算思维嵌入数学、科学、历史等跨学科课堂活动,实现学科能力和计算思维的双重发展。例如,英国学校计算课程工作小组(Brown et al.,2014)、马来西亚国家教育部(Ismail,2016)等倡导将计算思维融入更广泛的学科,以提高学生解决问题的能力。三是专门设立计算思维学科,如澳大利亚(Falkner et al.,2014)和我国香港地区(CoolThink@JC,2016)打造全新课程框架教授计算思维。计算思维与编程教学相辅相成,一方面编程教学是培养计算思维的有效手段,另一方面培养学生的计算思维是编程教学的高阶目标,帮助他们从技能操作层面转向创新创造层面。计算思维是一项通用技能,不仅关于计算,更是跨越了课程边界,为理解和改变世界提供了强有力的途径(Teaching London Computing, 2017)。因此,只有在丰富的跨学科环境中培养计算思维,才能打破认知和应用的壁垒,使其真正为学生提供思考世界和解决问题的新视角。但在与其他学科结合时,计算思维的培养目标要与课程目标协调,要结合学科特色开展,这往往有难度。将计算思维列为独立学科,最大的优势是能全面地兼顾计算思维多方面的培养内容,更加灵活。

(三)测评方法

教育工作者需要有适当的措施验证学生所学知

识,如果不注重评估,计算思维就无法成功进入 K-12 课程(Grover & Pea,2013)。现阶段常用的 K-12 学生计算思维测评方式主要有六种:1) 题目测试法,如计算思维技能转移测试题(Bebras,2004);2) 编程测试法,如程序故障的测试调试(Zhong et al.,2016);3) 作品分析法,如编程成品自动化分析工具 Dr. Scratch(Moreno-León et al.,2015);4) 调查法,如计算思维水平量表(Korkmaz et al.,2015);5) 图文分析法,如流程图评价法(Berland & Wilensky,2015);6) 观察访谈法,如基于项目的访谈法(Brennan & Resnick,2012)。这些方法各有侧重点,题目测试法和编程测试法倾向于考察基本概念和局部操作;作品分析法和图文分析法常作为对问题解决与作品开发整体的综合考量;调查法和观察访谈法是评价计算思维非认知因素的主要手段。由于计算思维由多元能力组成,其复杂的内容结构决定了不可能用一种单一的方法评估,因此在实际操作时组合多种方法才能提供关于学生对计算概念理解的更多更深入的信息(Allsop,2019)。此外,计算思维培养强调关注学生经历的思维发展过程而不仅仅是编程制品的结果,因此多元评估还需体现在评价类型的多元化,形成性与总结性评价缺一不可。

总体而言,计算思维这一复杂概念目前仍然是定义不清的心理结构(Román-González et al.,2017a),其名词网络尚未完全建立起来。在这种情况下,要建立计算思维培养的框架,只靠单一版本的定义或做法尚不足以完全支撑起对计算思维培养的立体化理解。在诸多定义中,计算思维三维框架(Brennan & Resnick,2012)基于可视化思维和过程思维,在教育实践中有较强的可操作性(Zhong et al.,2016),而且计算概念、计算实践和计算观念也能很好地对照教学的三维目标(王旭卿,2014)。因此,本研究在计算思维三维框架的基础上,界定了计算概念(编程过程中形成的可迁移至不同情境的技能或知识)、计算实践(在计算学习或编程练习中形成的解决问题的策略)和计算观念(具备以计算的视角洞察与理解自身和周围世界的良好品质)的范畴,然后根据其他权威定义的文本,依次补充每个维度下的二级、三级指标;同时,由于计算思维作为非独立存在的概念,容纳了不同思维模型,并且在实践测评中常与逻辑、算法、创造性思维等交汇融合

(Hu, 2011), 因此本研究还添加了计算思维的第四维度, 即与计算思维相关的思维模型。最终整合形成的计算思维四维内容体系(见表一)全面考虑了计算思维的内涵与外延以及各要素的内部结构, 所搭建的计算思维培养内容体系使得 K-12 培养活动不论从哪些能力组合的角度出发都能有据可循。表一还呈现了内容指标与教学方式、测评方法间的对照关系, 可有效指导教师针对不同培养目标开展教学与测评实践。

三、实践层面: K-12 测评实践案例分析

(一) 案例收集

为了解现阶段 K-12 计算思维培养实践的开展情况, 本研究收集了相关测评实践案例。2019 年 2 月, 研究人员以“计算思维”(Computational Thinking)为关键词在 SAGE、ERIC、Springer Link、ACM、Science Direct、IEEE 以及中国知网等数据库中检索相关文献, 然后结合文章摘要、关键词和各层级标题剔除综述与理论类文献、非 K-12 阶段的文献以及不包含计算思维测评环节的文献, 再从检索到文献的参考文献中追踪相关文献, 最终获得有效文献 41 篇(见表二)。

(二) 案例编码

研究人员依据 K-12 计算思维培养框架(见表一)对检索到的 41 篇论文进行编码, 重点关注这些

表二 案例检索结果统计(截至 2019 年 2 月)

数据库	有效文献(篇)
SAGE	2
ERIC	2
Springer Link	6
ACM	8
Science Direct	2
IEEE	14
中国知网	2
其他	5
总计	41

研究的基本信息(开展时间、国家和实验对象等)、培养内容、教学方式、测评方法与工具、研究结论等内容。本研究对需要编码的维度均进行两次编码, 最终编码一致性达到 100%。综合来看, 一篇文章可能围绕多个维度针对多个阶段的学生展开培养, 并采用多种方式进行测评, 编码的基本原则是各指标出现一次就计数一次, 因此最终的编码结果往往会出现一对多的情况。

(三) 案例分析

1. 基本情况

相对于计算思维理论和教学方面的研究, K-12 计算思维测评实践起步较晚, 2010 年起有迹可循。近年来随着对 K-12 计算思维培养活动的普遍重视, 越来越多的国家开始关注测评研究, 研究数量呈上

表一 K-12 计算思维培养框架

培养内容(四维体系)			教学方式	主要测评方法
一级	二级	三级		
计算概念 (编码: CC)	数据(CD)	数据收集(CD-1)、数据存储(CD-2)、数据分析(CD-3)、数据表示(CD-4)	编程教学、独立学科	题目测试法
	控制(CK)	顺序(CK-1)、条件(CK-2)、循环(CK-3)、并行(CK-4)		
	编程(CS)	事件(CS-1)、运算符(CS-2)、用户交互性(CS-3)		
	计算术语(CY): 参数、索引、效率等需要了解的计算机科学基础概念			
系统(CX): 在掌握基本计算概念的基础上, 理解系统开发是一个重复的设计过程, 涉及框架构造、开发和维护等多种复杂决策的权衡协调				
计算实践(CP)	问题(PW)	问题识别(PW-1)、问题分解(PW-2)、抽象建模(PW-3)、问题解决(自动化)(PW-4)、问题迁移(泛化)(PW-5)	编程教学、跨学科课堂、独立学科	题目测试法 编程测试法 作品分析法 图文分析法
	作品(PZ)	规划设计(PZ-1)、递增与迭代(PZ-2)、模块化与重用(PZ-3)、测试与调试(PZ-4)、评估与提升(PZ-5)		
计算观念(CA)	计算品质(恒心毅力等)(AQ)、计算表达(AE)、计算协作(AC)、计算意识(质疑等)(AA)		编程教学、跨学科课堂、独立学科	调查法 观察访谈法
相关思维(RT)	算法思维(TS)、逻辑思维(TL)、批判思维(TP)、创造性思维(TC)、分析思维(TO)		跨学科课堂、独立学科	题目测试法 调查法

升趋势(见图1)。从地区看,美国走在发展的前列,研究数量占总数量的51%,其他国家和地区尚处于初步探索阶段。从学段看,现有实践已经覆盖K-12全学段,但主要集中在6-8年级(见图2)。结合皮亚杰的认知发展理论,这一阶段学生的抽象逻辑能力逐渐形成。抽象逻辑是计算思维的重要组成部分,因此在这一阶段开展测评,可操作性强,测评效果也更明显。其他学段研究的缺乏表明测评的实践和研究还存在较多空白。

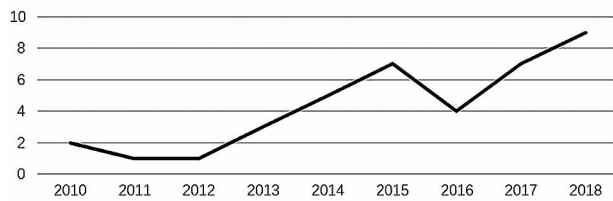


图1 K-12 计算思维测评实践案例文献发表时间分布
注:因为文献调研截至2019年2月,故图1未呈现2019年数据。

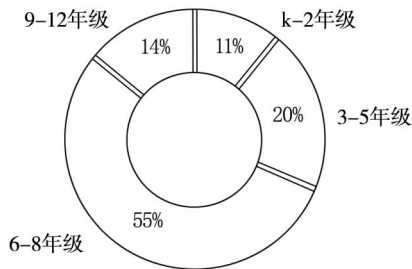


图2 K-12 计算思维测评实践研究学段分布

2. 培养内容

实践案例在计算概念维度主要集中于对控制的测评,聚焦在循环和条件两方面。计算实践维度对抽象建模、测试与调试的关注较多。相关思维维度较关注对算法思维的测评,这体现了实践与理论的契合。计算思维在计算科学领域的最初原型就是算法思维(Denning,2009),算法思维属于计算思维的核心要素之一,备受关注。整体看来,有关计算观念的测评最为欠缺,一方面在于课程培养中对非认知要素关注不足,另一方面在于可选的测评手段有限,且受实操可行性限制。实践案例还包含理论框架未涉及的培养内容CS-4,即计算思维模式分析(Computational Thinking Pattern Analysis,简称CTPA)(Ioannidou et al.,2011),这是为了评估学生提交的A-

gent Sheets 游戏作品代码意义而衍生的项目专用评价模式。综合来看,表一中大部分的理论维度都已落实于实践,但只有较少的维度受到广泛关注,非认知层面的实践还较少,数据收集、数据分析、系统和问题识别等还存在空白(见图3)。

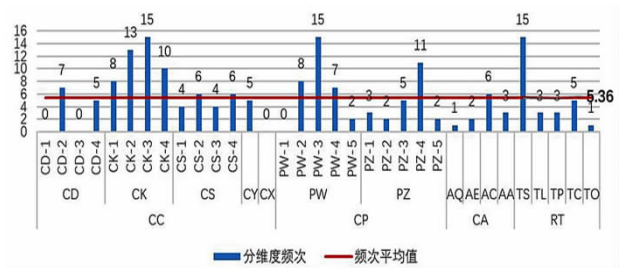


图3 K-12 计算思维实践测评维度频次统计

3. 教学方式

41项K-12计算思维实践案例研究中,71%的测评与教学直接相关,这其中约72%属于编程教学范畴,24%涉及跨学科课堂计算思维的培养。在STEM等教学中,计算思维适用于其他领域和一般化问题的解决得到了有效证实(Arraki et al.,2014)。仅4%的案例发生在培养计算思维的专门学科中。目前看来,虽然跨学科的意识有所增强,但编程教育仍是K-12计算思维培养活动的主要形式,这也说明了为什么现阶段针对算法思维和控制维度的测评较多,这是受整个培养实践的影响。

具体到教学细节,这些教学大多在Scratch等丰富的计算环境下开展。这种环境可以检查、操作和定制底层抽象和机制,辅助用户发展计算思维并实现从消费者向创造者的身份转换(Lee et al.,2011)。与面向结果、注重技能培养的传统程序教学不同,关注过程、构建计算解决方案是计算思维教学的核心目标,课堂中教师的角色相对被淡化,学生多通过合作完成问题解决或项目开发,整体上留给学生发挥的弹性空间较大。教学活动通常是开发一个游戏或者对科学现象进行计算建模。游戏开发不仅具有激励因素,还可以帮助学生独立思考,体现建构主义思想(Munoz et al.,2018)。综合来看,计算思维与成熟课程相整合的形式,能更快地促进计算思维的普及教育。在与编程教育的结合过程中,虽然受课程性质和工具使用的影响,实施测评更为自然,但也造成了对计算思维不同要素的关注分配不

均。此外,课堂教学常用的合作学习虽有助于实现课堂主体向学生的转移、增强学生计算协作能力,但如何关注小组成员个体能力的发展仍是问题。

4. 测评方法

从表三可以看出,六种测评方法均有一定的实践基础,并在实践中衍生更多的方法或工具。具体来看,题目测试法可操作性强,适用年龄段广,是目前使用频率最高的计算思维测评方式。题目测试法以客观题为主,集中考察学习者计算概念、计算实践以及相关思维的发展情况。编程教学是目前计算思维培养的主要手段,因此编程测试法也得到了广泛使用,测试的主要形式是指定开发任务,以编程为媒介考察学生计算实践的掌握情况。作品分析法是指结合编程成品或开发过程对计算实践进行评价。Dr. Scratch 和 Agrade 是最常见的两种自动化作品分析工具,能有效节省人工分析的时间成本且达到较高的准确度(Moreno-León et al., 2015),也是当前常用的总结性评价手段。过程实时评价则通过即时捕获每个学习者与系统交互的底层细节,通过对这些数据的分析帮助教师监控学生学习情况,尤其是那些需要帮助但又保持沉默的学习者的实时发展情况。这一工具的出现实现了对计算思维进行过程性评价的重大突破(Srinivas et al., 2018),但目前较少被使用。调查法常使用李克特量表衡量学生计算观念和相关思维发展情况,这种形式的测评结构简单易设置,实操性强,但由于评价主体变成了学生,若要深入了解学生的发展,还应结合其他方式收集基于知识的评价数据(Weese & Feldhausen, 2017)。图文分析法主要基于学生的反思报告、创新设计报告中的伪代码和流程图等内容,考察学生形成关系机制、建立结构模型的能力,即计算思维的抽象建模。实践证明,图文形式可帮助教师发现学生的思维障碍,特别是在计算实践方面(Zhong et al., 2016)。观察访谈法常作为其它测评方式的补充,辅助说明测评结果,适合于计算品质等难以用其他方式测评的计算思维。这种方法使用半结构化访谈或课堂观察记录、田野笔记得出结论。

单一的评价方式无法展现学生理解的全部过程。然而,在 41 个测评案例中,仅 37% 的案例采用两种及以上的多元评价方式,其余只采用单一的测评手段;仅 20% 明确整合了形成性评价与总结性

评价。

5. 案例结论

本研究通过查看上述案例的研究结论,得到以下发现:1)同一学段学生不同维度计算思维的发展不均衡,例如,有研究指出七八年级学生的抽象和数据表示较弱,但同步和流控制表现较好(Lawanto et al., 2017)。因此,计算思维培养要结合学生的思维发展特点,有侧重、针对性地开展教学,无需追求全部维度的发展。2)计算思维被认为是读、写、算之外的一项基本素养,但计算思维的发展依赖于读、算等基础能力。研究表明,计算技能需要阅读和数学技能的支撑,阅读和数学技能低于标准的学习者往往无法完成计算思维的评估任务(Seiter, 2015)。这一发现进一步解释了为什么在低年龄段,较少培养计算思维培养,因为较低的认知水平阻碍了培养活动的深入开展。3)在教学中,编程教育作为主要课程形式被广泛使用,但要把握编程工具和机器人等硬件设备的使用度。在繁忙的环境中教授严谨的计算概念会使学生分心,影响教学效果(Weese & Feldhausen, 2017)。4)学生的计算思维技能水平发展不受互联网、移动设备使用能力的影响(Korucu et al., 2017)。5)课程编排要考虑性别差异,特别要关注女生的参与程度和兴趣。研究表明,男女能力发展不存在明显差异,但女生不善于评估自我能力,导致其自我效能感比男生低,需要一定的干预措施帮助女生提高自我效能感(Duncan & Bell, 2015);对课程的感兴趣程度是影响学生自我效能的重要因素,在传统编程课上,女生的兴趣度往往不如男生(Kong et al., 2018)。

四、实践与理论的距离:可行的研究空间

综上,计算思维培养内容的实践与理论差距较小,但教学方式和测评方法还存有较大的发展空间(见图 4)。两者之间的差距也为今后的发展提供了研究空间。

(一)增强对计算思维内容体系的系统化认知及对非认知层面的关注

人工智能时代人们对计算思维寄予了很高的期望,因而从理论层面赋予其丰富的内容体系。但在实践层面仍存在培养和测评不均衡以及不合理的操作。首先,尚未有研究对数据收集、数据分析、系统

表三 计算思维测评方法使用情况

测评方式	具体手段	方法简介	使用频次	主要适用评价类型	文献编号
题目测试法	计算思维术语测试	考查学生对计算科学相关概念的认识和解读	5	总结性评价	[2]、[10]、[27]、[49]、[53]
	计算思维技能测试	评估学生在计算情境下解决问题的能力	9		[13]、[21]、[26]、[27]、[48]、[56]、[62]、[69]、[77]
	计算思维技能迁移评估测试	评估学生将计算思维技能迁移到不同类型情境中的能力	5		[21]、[37]、[49]、[54]、[73]
编程测试法	程序代码阅读理解	测试学生阅读和理解基于文本的代码片段的能力	2	总结性评价	[26]、[70]
	纠错和故障排除类任务	要求学生修正程序中的错误,使其正常运行	3		[58]、[64]、[76]
	指定式开发任务	向学生提供所需的功能列表,然后要求他们编写和操作代码块,并让这些功能正确运行	7	形成性评价、总结性评价	[21]、[26]、[27]、[58]、[67]、[70]、[76]
	开放式开发任务	学生需完成具有开放结果和开放过程的项目开发任务	3		[21]、[26]、[76]
作品分析法	作品成品评价	对最终的编程制品代码或功能等进行评价	11	总结性评价	[1]、[3]、[5]、[12]、[17]、[32]、[39]、[44]、[47]、[48]、[57]
	开发过程实时评价	记录学生编程创作每一步的数据流,掌握学生实时发展情况	2	形成性评价	[40]、[60]
调查法	量表法	李克特五点量表	7	总结性评价	[10]、[22]、[41]、[43]、[50]、[65]、[72]
	问卷调查法	调查学生的兴趣和态度以及对一些开放性问题的看法	2		[21]、[26]
图文分析法	文本分析法	对项目开发过程中填写的脚手架、书写的伪代码等进行分析	4	形成性评价、总结性评价	[1]、[71]、[73]、[76]
	图示法	分析基于流程图的算法设计	5		[3]、[6]、[71]、[73]、[76]
观察访谈法	访谈法	针对课程学习成果及校外相关活动进行提问	3	总结性评价	[1]、[26]、[62]
	(课堂)观察法	建立量规,结合该准则对学生进行课堂观察评估	4	形成性评价	[1]、[2]、[21]、[73]

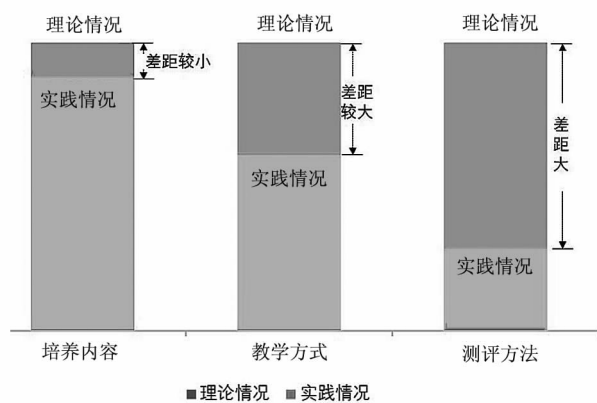


图4 实践与理论差距对比

以及问题识别开展测评,前两者是大数据的关键技术,是引领人工智能发展的必要条件;系统是考查学生将基本计算概念融会贯通后对系统架构的整体认知,是培养学生在计算实践中解决问题和动手设计

开发作品的基础;问题识别则是问题解决的首要环节,这些指标在计算思维中占重要位置。其次,41个实践案例更多地是围绕计算概念、计算实践等认知层面的计算思维进行测评,而对计算观念等非认知评价较少。虽然计算思维主要是一种解决问题能力的认知心理学构造,但它同时也存在互补的非认知因素。此外,计算思维包含问题识别、分解、抽象建模、问题解决与迁移等一系列步骤,这些步骤在实践中以线性的方式相互跟随,构成闭环,循环往复。然而,现阶段研究往往只聚焦几个环节,这就造成理论与测评实践之间的断层,未能全面呈现学生问题解决能力的发展情况,同理作品维度也是如此。造成上述问题的主要原因在于对计算思维概念的认识不全面,遗漏了对部分重要维度的培养以及忽略了培养内容指标内部的联系,只关注局部信息,不能体系化。因此,后续研究首先应注重对理论框架的全面解读,在此基础上补充对空白内容指标的培养

和测评;提升对非认知因素的培养和测评,实现对学生计算思维的全方位发展;在教与评的过程中注重内容指标间的内部关联,促进测评的系统化,扩大测评的涉及面。

(二)探索在更多学科嵌入式培养计算思维

计算思维源于计算科学领域,与编程教学密不可分,理论上二者相辅相成,互相促进,编程教学承载着发展计算思维的重任。在课程载体的选择方面,K-12 计算思维培养活动大多与编程教学结合,并取得了较好的阶段性成果。然而,受编程课程特点的影响,仅算法思维和控制等维度受到较多关注,各分量间发展不均衡(见图3)。为了促使计算思维测评全面发展,将计算思维列为单独学科是最佳选择,但在当前中小学的课程设置情况下,大多数国家或地区并没有时间或资源将计算思维列为一门独立学科,跨学科整合不仅解决了培养阵地的的问题,使擅长不同学科的学生都能从中收益,而且能够帮助学生意识到计算思维可以应用于许多领域,体验计算思维的应用前景。虽然目前对计算思维与非计算科学学科整合的呼声和资源支持在不断增强,但从数据反馈看,跨学科培养计算思维的意识仍然不强,应探索不同学科的计算思维培养。

在具体教学实施过程中,以小组合作为主的学习方式虽然实现了将课堂主动权交给学生的重要进步,但同时也存在个别成员“一言堂”把控全局、部分成员合作过程中“划水”现象严重、学生个性化发展受阻等隐患。为了避免这种情况,教师可尝试增强个人设计解决方案的比重,让每位学生都有充分的发挥空间;在必要的小组合作环节,教师可以通过给学生分配特定角色、独立评估每位学生的表现等方式避免消极影响。

(三)践行多元评价方式,加强形成性评价

针对计算思维的复杂构成,理论研究提出了多种测评方式,并指出要将这些方式以多元组合的形式全面了解学生计算思维的发展情况。但实践表明,仅37%的案例研究采用多元评价方式,20%的研究同时关注形成性与总结性评价。现阶段研究更偏重总结性评价,造成这一问题的主要原因有:1)受测评方式可操作性的影响,题目测试法、作品成品评价法简单易操作,不占课堂过多时间,测评结果定量直观,较受青睐,但测评结果仅能反映学生发展的

片面情况;2)对学生解决问题的思考过程关注不足,评价依据更多来自静态的程序代码、测试结果,导致测评不能体现学生思维发展的细节变化。后续研究应:1)优化测评方式,细化每种方式的实施步骤并提高操作的可行性;2)结合培养内容,组合适当的评价方式,践行多元评价模式;3)同时关注过程与结果,让评价贯穿教学始终,减轻学生负担的同时使测评结果更加全面可靠。

五、结论与展望

计算思维测评作为计算思维培养的重要组成部分,能对实践效果和理论价值起到重要的诊断作用,因此本研究将其作为了解计算思维培养实践开展情况的重要立足点。在梳理国内外文献的基础上,本研究整合形成了计算思维培养框架,尤其在内容体系上比较全面地考虑了计算思维的内涵与外延以及要素间的内部结构;然后,基于41个测评案例,从培养内容、教学方式和测评方法三方面分析理论与实践开展的差距,探讨可能的原因,提出了未来的研究方向。

总的看来,目前实践与理论相呼应的地方有:绝大部分培养内容在实践中有所体现、三类教学形式均有一定的实践基础、六种测评方法在实践中得到了使用和细化发展。存在的主要问题包括:培养内容不全面、教学形式使用相对单一以及多元评价意识不足。未来研究可从以下三方面着手:结合我国的国情和发展目标,构建对计算思维的全面认识,以信息技术教师为向导,通过更丰富的课程设计与协同开启算法思维、控制之外的培养维度,尤其是数据分析等研究维度,加强对问题维度的全面把握,加强对计算思维非认知要素的关注;关注形成性评价,将测评纳入教学各个环节,使得教有所依,实现从一刀切、笼统化的片面评价迈向客观全面的评价;综合应用各种测评方法,全面展现计算思维,如通过徽章化的形式展现学生计算思维的个性化发展过程和差异。

在现今全球互联的时代,要想有效理解、参与甚至改造周边基于计算的世界,计算思维已然成为一项备受重视的关键素质。随着人工智能人才竞争的加剧,从低年龄阶段就开始培养学生的计算思维能力显得尤为重要(Haseski et al., 2018),如何在K-12

培养、推广和评估计算思维仍是一项艰巨的挑战 (Freeman et al., 2017), 这也将是今后较长一段时间的研究重点 (Djurdjevic-Pahl et al., 2016)。

[参考文献]

- [1] Allsop, Y. (2019). Assessing computational thinking process using a multiple evaluation approach [J]. *International Journal of Child-computer Interaction*, 19: 30-55.
- [2] Arraki, K., Blair, K., Bürgert, T., Greenling, J., Haebe, J., Lee, G., Peel, A., Szczepanski, V., Pontelli, E., & Hug, S. (2014). DISSECT: An experiment in infusing computational thinking in K-12 science curricula [A]. 2014 IEEE Frontiers in Education Conference (FIE) Proceedings [C]. IEEE: 1-9.
- [3] Basu, S., Kinnebrew, J. S., & Biswas, G. (2014). Assessing student performance in a computational-thinking based science learning environment [A]. In *International conference on intelligent tutoring systems* [C]. Springer, Cham: 476-481.
- [4] Bebras (2004). Task examples [EB/OL]. [2019-5-30]. <https://www.bebas.org/?q=examples>.
- [5] Bennett, V. E., Koh, K., & Repenning, A. (2013). Computing creativity: Divergence in computational thinking [A]. *Proceeding of the 44th ACM technical symposium on Computer science education* [C]. ACM: 359-364.
- [6] Berland, M., & Wilensky, U. (2015). Comparing virtual and physical robotics environments for supporting complex systems and computational thinking [J]. *Journal of Science Education and Technology*, 24(5): 628-647.
- [7] Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking [EB/OL]. [2019-5-30]. <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>.
- [8] Brown, N. C., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart: The resurgence of computer science in UK schools [J]. *ACM Transactions on Computing Education (TOCE)*, 14(2): 1-22.
- [9] Buffum, P. S., Lobene, E. V., Frankosky, M. H., Boyer, K. E., Wiebe, E. N., & Lester, J. C. (2015). A practical guide to developing and validating computer science knowledge assessments with application to middle school [A]. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* [C]. ACM: 622-627.
- [10] Burgett, T., Folk, R., Fulton, J., Peel, A., Pontelli, E., & Szczepanski, V. (2015). DISSECT: Analysis of pedagogical techniques to integrate computational thinking into K-12 curricula [A]. 2015 IEEE Frontiers in Education Conference (FIE) [C]. IEEE: 1-9.
- [11] CAS (2015). Computational thinking: A guide for teachers [EB/OL]. [2019-5-30]. <https://community.computingatschool.org.uk/resources/2324/single>.
- [12] Chang, Z., Sun, Y., Wu, T. Y., & Guizani, M. (2018). Scratch analysis Tool (SAT): A modern scratch project analysis tool based on ANTLR to assess computational thinking skills [A]. 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC) [C]. IEEE: 950-955.
- [13] Chaudhary, V., Agrawal, V., Sureka, P., & Sureka, A. (2016). An experience report on teaching programming and computational thinking to elementary level children using lego robotics education kit [A]. 2016 IEEE Eighth International Conference on Technology for Education (T4E) [C]. IEEE: 38-41.
- [14] Code.Org (2013). Hour of Code [EB/OL]. [2019-5-30]. <https://hourofcode.com/cn/zh/learn>. [15] CoolThink@JC (2016). Our Approach [EB/OL]. [2019-5-30]. <https://www.coolthink.hk/en/ct/>.
- [16] Corradini, I., Lodi, M., & Nardelli, E. (2017). Conceptions and misconceptions about computational thinking among Italian primary school teachers [A]. *Proceedings of the 2017 ACM Conference on International Computing Education Research* [C]. ACM: 136-144.
- [17] Dasgupta, S., Hale, W., Monroy-Hernández, A., & Hill, B. M. (2016). Remixing as a pathway to computational thinking [A]. *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing* [C]. ACM: 1438-1449.
- [18] Deng, Z., Huang, W., & Dong, R. (2009). Discussion of ability cultivation of computational thinking in course teaching [A]. 2009 International Conference on Education Technology and Computer [C]. IEEE: 197-200.
- [19] Denning, P., J. (2009). The profession of IT beyond computational thinking [EB/OL]. [2019-5-30]. <http://denninginstitute.com/pjd/PUBS/CACMcols/cacmJun09.pdf>.
- [20] Djurdjevic-Pahl, A., Pahl, C., Fronza, I., & El Ioini, N. (2016). A pathway into computational thinking in primary schools [A]. *International Symposium on Emerging Technologies for Education* [C]. Springer, Cham: 165-175.
- [21] Duncan, C., & Bell, T. (2015). A pilot computer science and programming course for primary school students [A]. *Proceedings of the Workshop in Primary and Secondary Computing Education* [C]. ACM: 39-48.
- [22] Durak, H. Y. (2018). The effects of using different tools in programming teaching of secondary school students on engagement, computational thinking and reflective thinking skills for problem solving [DB/OL]. [2019-5-30]. <https://doi.org/10.1007/s10758-018-9391-y>.
- [23] ECT (2017). CT Overview [EB/OL]. [2019-05-30]. <https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-overview>.
- [24] Falkner, K., Vivian, R., & Falkner, N. (2014). The Australian digital technologies curriculum: Challenge and opportunity [A]. *Proceedings of the Sixteenth Australasian Computing Education Conference-Volume 148* [C]. Australian Computer Society, Inc.: 3-12.
- [25] Freeman, A., Adams Becker, S., Cummins, M., Davis, A., & Hall Giesinger, C. (2017). *The NMC/CoSN Horizon Report*:

- 2017 K-12 Edition [EB/OL]. [2019-5-30]. <https://cdn.nmc.org/media/2017-nmc-cosn-horizon-report-k12-EN.pdf>.
- [26] Grover, S. (2017). Assessing algorithmic and computational thinking in K-12: Lessons from a middle school classroom [M]. In Emerging research, practice, and policy on computational thinking (pp. 269-288). Springer, Cham.
- [27] Grover, S., Cooper, S., & Pea, R. (2014). Assessing computational learning in K-12 [A]. Proceedings of the 2014 conference on Innovation & technology in computer science education [C]. ACM: 57-62.
- [28] Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field [J]. Educational researcher, 42(1): 38-43.
- [29] 国务院(2017). 国务院印发《新一代人工智能发展规划》[EB/OL]. [2019-5-30]. http://www.gov.cn/xinwen/2017-07/20/content_5212064.htm.
- [30] Haseski, H. I., Ilic, U., & Tugtekin, U. (2018). Defining a new 21st Century skill-computational thinking: Concepts and trends [J]. International Education Studies, 11(4): 29-42.
- [31] Hu, C. (2011). Computational thinking: What it might mean and what we might do about it [A]. Proceedings of the 16th annual joint conference on Innovation and technology in computer science education [C]. ACM: 223-227.
- [32] Ioannidou, A., Bennett, V., Repenning, A., Koh, K. H., & Basawapatna, A. (2011). Computational thinking patterns [DB/OL]. [2019-05-30]. <https://files.eric.ed.gov/fulltext/ED520742.pdf>.
- [33] Ismail, A. R. (2016). Computational thinking to be integrated into primary and secondary curriculum [EB/OL]. [2019-05-30]. <https://www.hardwarezone.com.my/tech-news-computational-thinking-be-integrated-primary-and-secondary-curriculum>.
- [34] ISTE & CSTA(2011a). Operational Definition of Computational Thinking for K-12 Education [EB/OL]. [2019-05-30]. <http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf>.
- [35] ISTE & CSTA(2011b). Computational Thinking teacher resources [EB/OL]. [2019-05-30]. https://id.iste.org/docs/ct-documents/ct-teacher-resources_2ed-pdf.pdf?sfvrsn=2.
- [36] 教育部(2018). 普通高中信息技术课程标准(2017年版)[M]. 北京:人民教育出版社:5-6.
- [37] Jun, S., Han, S., Kim, H., & Lee, W. (2014). Assessing the computational literacy of elementary students on a national level in Korea [J]. Educational Assessment, Evaluation and Accountability, 26(4): 319-332.
- [38] K-12 Computer Science (2016). K-12 Computer Science Framework [EB/OL]. [2019-05-30]. <https://k12cs.org/>.
- [39] Koh, K. H., Basawapatna, A., Bennett, V., & Repenning, A. (2010). Towards the automatic recognition of computational thinking for adaptive visual language learning [A]. 2010 IEEE Symposium on Visual Languages and Human-Centric Computing [C]. IEEE: 59-66.
- [40] Koh, K. H., Basawapatna, A., Nickerson, H., & Repenning, A. (2014). Real time assessment of computational thinking [A]. 2014 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) [C]. IEEE: 49-52.
- [41] Kong, S. C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education [J]. Computers & Education, 127: 178-189.
- [42] Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2015). Computational thinking levels scale (ctls) adaptation for secondary school level [J]. Gazi Journal of Educational Science, 1(2): 143-162.
- [43] Korucu, A., Gencturk, A., & Gundogdu, M. (2017). Examination of the computational thinking skills of students [J]. Journal of Learning and Teaching in Digital Age, 2(1): 11-19.
- [44] Lawanto, K., Close, K., Ames, C., & Brasiel, S. (2017). Exploring strengths and weaknesses in middle school students' computational thinking in scratch [M]. In Emerging Research, Practice, and Policy on Computational Thinking (pp. 307-326). Springer, Cham.
- [45] Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice [J]. Acm Inroads, 2(1): 32-37.
- [46] Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking [J]. RED. Revista de Educación a Distancia, (46): 1-23.
- [47] Munoz, R., Barcelos, T. S., Villarroel, R., & Silveira, I. F. (2016). Game design workshop to develop computational thinking skills in teenagers with Autism Spectrum Disorders [A]. 2016 11th Iberian Conference on Information Systems and Technologies (CISTI) [C]. IEEE: 1-4.
- [48] Munoz, R., Villarroel, R., Barcelos, T. S., Riquelme, F., Quezada, A., & Bustos-Valenzuela, P. (2018). Developing computational thinking skills in adolescents with autism spectrum disorder through digital game programming [DB/OL]. [2019-05-30]. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8502019>.
- [49] Nesiba, N., Pontelli, E., & Staley, T. (2015). DISSECT: Exploring the relationship between computational thinking and English literature in K-12 curricula [A]. 2015 IEEE Frontiers in Education Conference (FIE) [C]. IEEE: 1-8.
- [50] 宁可为, 杨晓霞(2018). 基于 App Inventor 的初中计算思维培养实证研究 [J]. 课程·教材·教法, 38(2): 110-115.
- [51] Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas [EB/OL]. [2019-05-30]. <http://kvannti.kapsi.fi/Documents/LCL/mindstorms-chap1.pdf>.
- [52] Papert, S. (1972). Teaching children thinking [J]. Programmed Learning and Educational Technology, 9(5): 245-255.
- [53] Peel, A., Fulton, J., & Pontelli, E. (2015). DISSECT:

- An experiment in infusing computational thinking in a sixth grade classroom [A]. 2015 IEEE Frontiers in Education Conference (FIE) [C]. IEEE: 1-8.
- [54] Rojas-López, A., & García-Pe? alvo, F. J. (2018). Learning scenarios for the subject Methodology of Programming from evaluating the Computational Thinking of new students [J]. IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, 13(1): 30-36.
- [55] Román-González, M., Moreno-León, J., & Robles, G. (2017a). Complementary tools for computational thinking assessment [A]. Proceedings of International Conference on Computational Thinking Education (CTE 2017), S. C Kong, J Sheldon, and K. Y Li (Eds.) [C]. The Education University of Hong Kong: 154-159.
- [56] Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017b). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test [J]. Computers in Human Behavior, 72: 678-691.
- [57] Seiter, L., & Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students [A]. Proceedings of the ninth annual international ACM conference on International computing education research [C]. ACM: 59-66.
- [58] Seiter, L. (2015). Using SOLO to classify the programming responses of primary grade students [A]. Proceedings of the 46th ACM Technical Symposium on Computer Science Education [C]. ACM: 540-545.
- [59] Selby, C., & Woollard, J. (2013). Computational thinking: the developing definition [EB/OL]. [2019-5-30]. https://eprints.soton.ac.uk/356481/1/Selby_Woollard_bg_soton_eprints.pdf.
- [60] Srinivas, M. J., Roy, M. M., Sagri, J. N., & Kumar, V. (2018). Assessing scratch programmers' development of computational thinking with transaction-level data [M]. In Towards Extensible and Adaptable Methods in Computing (pp. 399-407). Springer, Singapore.
- [61] Teaching London Computing (2017). Interdisciplinary Computational Thinking [EB/OL]. [2019-5-30]. <https://teachinglondon-computing.org/interdisciplinary-computational-thinking/>.
- [62] Tran, Y. (2018). Computational thinking equity in elementary classrooms: What third-grade students know and can do [DB/OL]. [2019-5-30]. <https://doi.org/10.1177/0735633117743918>.
- [63] 王旭卿(2014). 面向三维目标的国外中小学计算思维培养与评价研究 [J]. 电化教育研究, 35(7):48-53.
- [64] Webb, D. C. (2010). Troubleshooting assessment: An authentic problem solving activity for it education [J]. Procedia-Social and Behavioral Sciences, 9: 903-907.
- [65] Weese, J. L., & Feldhausen, R. (2017). STEM outreach: Assessing computational thinking and problem solving [A]. Conference Proceedings of ASEE Annual Conference and Exposition [C].
- [66] Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms [J]. Journal of Science Education and Technology, 25(1): 127-147.
- [67] Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The fairy performance assessment: Measuring computational thinking in middle school [A]. Proceedings of the 43rd ACM technical symposium on Computer Science Education [C]. ACM: 215-220.
- [68] Wing, J. M. (2006). Computational thinking [J]. Communications of the ACM, 49(3): 33-35.
- [69] Wong, G. K., & Jiang, S. (2018). Computational thinking education for children: Algorithmic thinking and debugging [A]. 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE) [C]. IEEE: 328-334.
- [70] Worrell, B., Brand, C., & Repenning, A. (2015). Collaboration and computational thinking: A classroom structure [A]. 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) [C]. IEEE: 183-187.
- [71] Xiao, M., & Yu, X. (2017). A model of cultivating computational thinking based on visual programming [A]. 2017 International Conference of Educational Innovation through Technology (EITT) [C]. IEEE: 75-80.
- [72] Yagci, M. (2019). A valid and reliable tool for examining computational thinking skills [J]. Education and Information Technologies, 24(1): 929-951.
- [73] 郁晓华,肖敏,王美玲,陈妍(2017). 基于可视化编程的计算思维培养模式研究:兼论信息技术课堂中计算思维的培养 [J]. 远程教育杂志, (6): 12-20.
- [74] 郁晓华,肖敏,王美玲(2018). 计算思维培养进行时:在K-12阶段的实践方法与评价 [J]. 远程教育杂志, 36(2):18-28.
- [75] 郑旭东(2018). 智慧教育2.0:教育信息化2.0视域下的教育新生态:《教育信息化2.0行动计划》解读之二 [J]. 远程教育杂志,36(4):1-19.
- [76] Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking [J]. Journal of Educational Computing Research, 53(4): 562-590.
- [77] Zur-Bargury, I., Pärvi, B., & Lanzberg, D. (2013). A nationwide exam as a tool for improving a new curriculum [A]. Proceedings of the 18th ACM conference on Innovation and technology in computer science education [C]. ACM: 267-272.

(编辑:魏志慧)

How Far is the Cultivation of Computational Thinking in K-12: From the Perspective of Computational Thinking Assessment

YU Xiaohua & WANG Meiling

(Department of Education Information Technology, East China Normal University, Shanghai 200062, China)

Abstract: *In the era of artificial intelligence, because of the competition for talents, the cultivation of computational thinking in K-12 has become an important focus. To cultivate computational thinking and make it more targeted, educators need to grasp the current development of K-12 practice accurately through an analytical perspective. Firstly, from the theoretical level, the K-12 computational thinking cultivation framework, consisting of three dimensions of cultivation content, teaching form, and assessment method, is constructed. Then, based on 41 cases of assessment practice, we use the methods of meta-analysis and content analysis. The gap between theoretical study and practice development in computational thinking cultivation is revealed and discussed, and some feasible research spaces for later research are demonstrated. The analysis results show that most of the theoretical cultivation content of computational thinking has been reflected in practice. All three types of teaching forms have realized a certain practical basis. Six kinds of assessment methods have been used and further refined in practice. However, there are also some shortcomings, such as incomplete cultivation content, relatively monotonous using of teaching form, and lack of multi-assessment consciousness. Therefore, a comprehensive understanding of the computational thinking content system should be established to fill the blank research dimension and increase the focus on non-cognitive aspects and internal relationships between indicators. Besides, more attempts should be taken into other curriculum forms, except for computer science and programming education, and the proportion of individual development should be improved. Finally, various assessment methods should be applied to demonstrate the development of computational thinking fully. These conclusions can provide some enlightenment and guidance for the future development of computational thinking.*

Key words: *computational thinking; K-12; artificial intelligence; programming*