International Journal of u- and e- Service, Science and Technology
Vol.8, No. 7 (2015), pp.387-396
http://dx.doi.org/10.14257/ijunesst.2015.8.7.39

Research on Primary Arithmetic Fact Mastery for Elementary School Children Education in China: a Computerized Programming Model of Problem Building

Tianxiang Zheng

Shenzhen Tourism College of Jinan University
zheng_tx@jnu.edu.cn

Abstract

Young children often show uneven patterns of competencies in mathematics and some of them cannot even master basic arithmetic facts. This study concerns the primary arithmetic fact mastery for elementary school children education in China by developing an auto problem building model. To do this, a well-defined programming algorithm was first proposed to construct three-term arithmetic operation facts including addition, subtraction, multiplication and division, followed by construction of four-term hybrid combinations. The proposed algorithm was then implemented accordingly by Excel VBA to get a computerized system, which supports randomized operands, one-click building and customized settings. The results show that this model can be applied to the parents or experimenters to give their children routine education in a variety of exercise situations, which helps the school children become more expert at solving arithmetic calculations and perform better on standardized achievement tests.

Keywords: *Elementary School Children Education, Primary Arithmetic Fact Mastery, Arithmetic Problem Solving, Computerized Programming, Excel VBA*

1. Introduction

For Chinese children from five or six years on, with an age, which means to attend academic grades for elementary school study, learning arithmetic is challenging. Not only must they learn to add, subtract, multiply and divide, they must also understand how these operations are related. It is well-recognized that children with poor fact mastery show weaknesses in nonverbal reasoning [1]. Previous studies have even shown that some children cannot master basic arithmetic facts despite relatively strong problem-solving skills [2-4]. Unlike sophisticated problem solving in mathematics which makes use of inferences, analogies, and other conceptually based techniques to give a whole outline to the solution, arithmetic facts requires elaborate calculation to give a more specific and detailed answer to the problem. To accomplish arithmetic fact tasks, participants need to master some skills such as encoding or adding multiple digits and temporarily holding partial results. They also need to know other useful strategies such as finding direct retrieval of the correct solution in long-term memory or using finger-counting strategy, and different transformation (*e.g.*, to solve 45 + 39, one can do 45 + 40 – 1, 40 + 40 + 5 – 1, or 50 + 34) and decomposition (*e.g.,* to solve 45 + 39, one can do 40 + 30 + 5 + 9, 45 + 30 + 9, 40 + 39 + 5, or 5 + 9 + 30 + 40) techniques that allow they to transform the problem into another one to suit their basic knowledge. As a result, arithmetic problem is hard for all graders, no matter what the operation is [5]. As to addition, for instance, it has been revealed that participants' performance decreased as the number and value of carries increased in multidigit addition problem solving [6]. As to subtraction, it has been found that the borrow effect in subtraction is so hard to execute that the children notably lack both in their knowledge that the minuend is conserved and in their knowledge of the

ISSN: 2005-4246 IJUNESST

values exchanged during regrouping [7, 8]. As to multiplication and division, it has been suggested that skill in these two situations depends on cognitive processes that are different from those needed for addition and subtraction, where the children most likely used computational strategies to reach solutions rather than rote retrieval [1]. Therefore, it is a great help for the children to perform plenty of arithmetical exercises on different types of calculation during the study period in order to become more procedurally proficient and serve for better understanding of school learning.

For children's parents, education on arithmetic fact solving for their kids is also a challenge. Some of them may want to find out how children spot the need for specific types of calculation when an operation fact is presented to them, and how they actively reconstruct the information given to them so that it fits the mathematical solution which they wish to apply. Others may wonder whether or not their kids can make particular kinds of calculations and whether or not they possess particular kinds of conceptual knowledge. Large majority, perhaps not all, of these parents may need to understand the logical structures that are constructed by their children as they learn from typical school instruction [9]. However, they suffered from a deficit in energy due to the busy life of everyday's routine jobs and housework, which prevents them from guiding their children procedural skills, not even personally building up a paper for the exercise. Therefore, it is a great help for the parents to have a chance of giving some education to their children, if there exists such an auto-building system.

Because the main goal in arithmetic teaching and learning is to develop the ability to resolve a variety of complex step by step organized tasks [10], primary arithmetic fact solving has special importance in children's study so it is crucial to reexamine this issue. Past researches have focused mainly on (1) the age-related strategies that the participants used, including inversion problems solving skills [11, 5, 12], computational estimation skills [13-15] and other mastery skills [1, 16-18], (2) performance level or learning abilities [19-23], (3) behaviors with learning disabilities [24-27] and (4) neuronal functionalities [28-32]. These researches have made great contributions to the understanding of primary arithmetic fact solving and mastery in the literature. Nevertheless, the problem itself, *i.e.*, the operation facts building on a test paper that contains addition, subtraction, multiplication, division and their combinations, has been ignored and left untreated. This paper seeks to contribute to leveraging this by developing a problem building model, which can automatically build up exercises with arithmetic combination problems made of operands with two-digit or three-digit numbers, to allow the children to examine their conceptual understanding and computational skill in arithmetic problem solving.

2. Model Architecture

2.1. Three-term Arithmetic Fact Building

2.1.1. Addition Operation

Denote the fact as "a + b = □" and suppose that either of the two addends is at the limit of a given number (denoted by "p"), the construction algorithm is listed as follows.

1) Generate a random number between 1 to "p" for the first addend "a". If any digit of the number is 0, the generation must be repeated. Since this rule will be used repeatedly in the following building, it is convenient to introduce a simpler notation named nonzero-rule.

2) Generate a random number between 1 to "p" for the second addend "b" that satisfies the nonzero-rule.

If a limited sum (also denoted by "p") is preferred to represent the result of "a + b" is not greater than "p", the algorithm should be changed to the following:

1) Generate a random number between 1 to “p- 1” for the first addend “a” that satisfies the nonzero-rule.

2) Generate a random number between “a + 1” to p for the sum result (suppose it is denoted by “s”). To ensure that the second addend “b” to be generated in the next procedure satisfies the nonzero-rule, digits must not be repeated in the same unit, decade, or hundred positions (except the highest position) across “a” and “s”. If any repeat happens, the generation must be done again.

3) Solve the expression of “s - a” to get the second addend “b”.

2.1.2. Subtraction Operation

Denote the fact as “a - b = □” and suppose that either the minuend or subtrahend is at the limit of a given number (denoted by “q”), the construction algorithm is listed as follows.

1) Generate a random number between 1 to “q” for the minuend “a”.

2) Generate a random number between 1 to “a – 1” for the subtrahend “b” that satisfies the nonzero-rule.

2.1.3. Multiplication Operation

Denote the fact as “a × b = □” and suppose that the multiplicand “a” is at the limit of a given number (denoted by “u”) while the multiplicator “b” is at the limit of another given number (denoted by “v”), the construction algorithm is listed as follows. By the way, without loss of generality, suppose that “u > v”.

1) Generate a random number between 2 to “u” for the multiplicand “a”. If “a” is the power of ten, like 1, 10, or 100, the generation must be repeated. Since this rule will be used repeatedly in the following building, it is convenient to introduce a simpler notation named non-ten-power-rule.

2) Generate a random number between 2 to “v” for the multiplicator “b” that satisfies the non-ten-power-rule.

2.1.4. Division Operation

Denote the fact as “a ÷ b = □” and suppose that the dividend “a” is at the limit of “u” while the divisor “b” is at the limit of “v”, the construction algorithm is listed as follows.

1) Generate a random number between 1 to “u” for the dividend “a”.

2) Generate a random number between 1 to “v” for the divisor “b” that satisfies the non-ten-power-rule.

2.2. Four-term Arithmetic Fact Building with Hybrid Combination

Apparently, the precedence order (*i.e.*, priority of operations) is crucial for the construction. Due to the page length limit, only several cases, which need special attentions, are illustrated.

2.2.1. “(a + b) ÷ c = s ÷ c = □”

1) Generate the first and second operands (“a” and “b”) by use of the above 2.1.1 strategy similarly, except for the requirement on the sum result “s”. Since “s” serves for the dividend of the next operation, it is required to be at the limit of “u”. In other words, the generation must be repeated until “s” is not greater than “u”. To avoid infinite iteration, a maximum loop is introduced to fulfill this requirement. If the iteration process reaches the maximum loop but no such candidate for “a” or “b” can be found, use the following strategy instead. First generate a random number between 1 to “u/2” for the

first addend "a", and then generate a random number between 1 to "u/2" for the second addend "b".

2) If aliquot result is required, find out all candidates from 2 to the sum result "s" and then randomly select one candidate as the divisor "c"

3) If no such candidate can be found, figure out the operation fact reversely. That is,

a) Generate a random number between 1 to "v" as one possible alternative of "c".

b) Generate a random number between 1 to <u ÷ c> (the notation "<a>" refers to potential floor effect of "a", similarly hereinafter) and then multiply this number by "c" to get "s".

c) Generate a random number between 1 to "p" for the first addend "a".

d) Solve the expression of "s - a" to get the second addend "b".

e) Randomly select one divisible candidate from 2 to "s" for the divisor "c".

2.2.2. "(a - b) ÷ c = s ÷ c = □"

1) Generate a random number between 1 to "q" for the subtrahend "b" that satisfies the nonzero-rule.

2) Generate a random number between 1 to "q – b" for the subtraction result "s". Since s serves for the dividend of the next operation, it is required to be at the limit of "u". In other words, the generation must be repeated until "s" is not greater than "u". To avoid infinite iteration, a maximum loop is introduced to fulfill this requirement. If the iteration process reaches the maximum loop but no such candidate for "b" or "s" can be found, use the following strategy instead. First generate a random number between 1 to "u" for the subtraction result "s", and then generate a random number between 1 to "q – s" for the subtrahend "b".

3) Solve the expression of "s + b" to get the minuend "a".

4) If aliquot result is required, find out all candidates from 2 to "s" and then randomly select one candidate as the divisor "c".

5) If no such candidate can be found, figure out the operation fact reversely. That is,

a) Generate a random number between 1 to "v" as one possible alternative of "c".

b) Generate a random number between 1 to <u ÷ c> and then multiply by "c" to get "s".

c) Generate a random number between 1 to "q – s" for the subtrahend "b".

d) Solve the expression of "s + b" to get the minuend "a".

e) Randomly select one divisible candidate from 2 to "s" for the divisor "c".

2.2.3. "a × b ÷ c = s ÷ c = □"

1) Generate the first and second operands ("a" and "b") by use of the above 2.1.3 strategy similarly, except for the requirement on the multiplication result "s". Since "s" serves for the dividend of the next operation, it is required to be at the limit of "u". In other words, the generation must be repeated until "s" is not greater than "u". To avoid infinite iteration, a maximum loop is introduced to fulfill this requirement. If the iteration process reaches the maximum loop but no such candidate for "a" or "b" can be found, use the following strategy instead. First generate a random number between 1 to square root of "u" for the multiplicand "a", and then generate a random number between 1 to square root of "v" for the multiplicator "b".

2) If aliquot result is required, find out all candidates from 2 to "s" and then randomly select one candidate as the divisor "c".

3) If no such candidate can be found, figure out the operation fact reversely. That is,

a) Generate a random number between 1 to "v" as one possible alternative of "c".

b) Generate a random number between 1 to <u ÷ c> and then multiply by "c" to get "s".

c) Find out a number between 1 to “s” that can be divisible by “s” for the multiplicand “a”.
d) Solve the expression of “s ÷ a” to get the multiplicator “b”.
e) Randomly select one divisible candidate from 2 to “s” for the divisor “c”.

2.2.4. “c ÷ (a ÷ b) = c ÷ s = □”

1) Generate a random number between 1 to “v” for the third operand “b” that satisfies the non-ten-power-rule.
2) Generate a random number between 1 to <u ÷ b> for “s”. Since “s” serves for the divisor of the next operation, it is required to be at the limit of “v”. In other words, the generation must be repeated until “s” is not greater than “v”. To avoid infinite iteration, a maximum loop is introduced to fulfill this requirement. If the iteration process reaches the maximum loop but no such candidate for “b” or “s” can be found, use the following strategy instead. First generate a random number between 1 to “v” for the division result “s”, and then generate a random number between 1 to “v” for the second addend “b”. Check whether “b × s” is greater than “u”, and if so, iterate the generation of “b” until “b × s < u” holds.
3) Solve the expression of “b × s” to get the second operand “a”.
4) If aliquot result is required, generate a random number between 1 to <u ÷ s> and then multiply this number by “s” to get the first operand “c”. Otherwise, generate a random number between “s” to “u” to get the first operand “c”.

2.2.5. “a ÷ b ÷ c = s ÷ c = □”

1) Generate a random number between 1 to “v” for the second operand “b” that satisfies the non-ten-power-rule.
2) Generate a random number between 1 to <u ÷ b> for “s”.
3) Solve the expression of “b × s” to get the first operand “a”.
4) If aliquot result is required, find out all candidates from 2 to “s” and then randomly select one candidate as the divisor “c”.
5) If no such candidate can be found, figure out the operation fact reversely. That is,
a) Generate a random number between 1 to “v” as one possible alternative of “c”.
b) Generate a random number between 1 to <u ÷ c> and then multiply by “c” to get “s”.
c) If <u ÷ s> is greater than “v”, then generate a random number between 1 to “v” for the operand “b”; Otherwise, generate a random number between 1 to <u ÷ s> for “b”.
d) Solve the expression of “s × b” to get the first operand “a”.
e) Randomly select one divisible candidate from 2 to “s” for the third operand “c”.

3. Implementation of Model and Auto Evaluation of the Problems

The proposed model was implemented to a system with Excel VBA and run on any computer that installed Microsoft Office, which is amenable to use. There are two issues about the model that still need to be addressed. For one thing, in order to cater for a variety of different-aged children, the system is designed to be versatile at the limit of, or just beyond, what could be solved by a particular age group. In other words, settings (also known as preferences) like the amount of problems, the range of operands and whether aliquot result is required or not, should be involved in and customized by the user. The homepage of the system is illustrated in Figure 1.

For the other, in order to further alleviate the experimenter’s burden on going over the paper, it appears to be more prevailing to automatically solve the problems (operation facts) generated by the aforementioned algorithm. To this end, “eval” function of

VBScript would be helpful. Before an operation fact was transferred to the "eval" function, some preprocessing procedures should be applied.

1) The notation "=" must first be eliminated from the original formula.

2) For the case of multiplication, the notation "×" must be replaced by "*" to suite the language of Visual Basic.

3) For the case of division, if it is a three-term arithmetic problem, or it is a four-term arithmetic problem but is not continued division, the notation "÷" must be replaced by "/" to suite the language. Otherwise, it is a four-term arithmetic problem with continued division. For such a formula, if aliquot result is imposed, the notation "÷" must be replaced by "/" to suite the language. Otherwise (*i.e.*, aliquot result is not required), the second operation (in the sense of priority of operations) "÷" must be replaced by "\" and "mod" in twice to calculate the quotient and residue respectively.

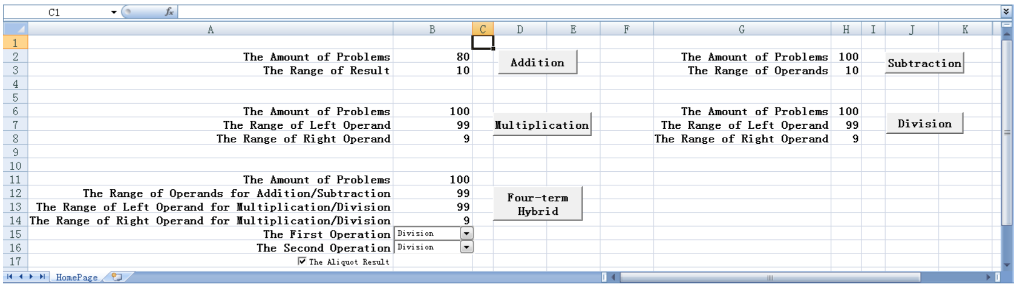


Figure 1. Homepage of the System (Settings)

4. Performance Illustration

If an experimenter wants to test the arithmetic fact solving ability of the child, what he/she needs to do is to open the excel file and enable macros to activate the VBA function. The addition, subtraction, multiplication and division button on the homepage, as shown in Figure 1, allows the experimenter to test the child's three-term arithmetic fact mastery, whereas the hybrid button and drop-down combo boxes are used for four-term arithmetic fact building. Apart from these buttons and combo boxes, the amount of problems, the difficulty of the paper and whether the result is required to be aliquot for the case that the second operation (in the sense of priority of operations) is division can also be specified. Once these preferences are set, a full page of problems which are matched across these settings is generated in the next worksheet. Figure 2 is a special case that the results of all the problems of addition operations are under 20 (more precisely, smaller or equal to 20). The experimenter can print this worksheet out as a test paper for the subsequent use.

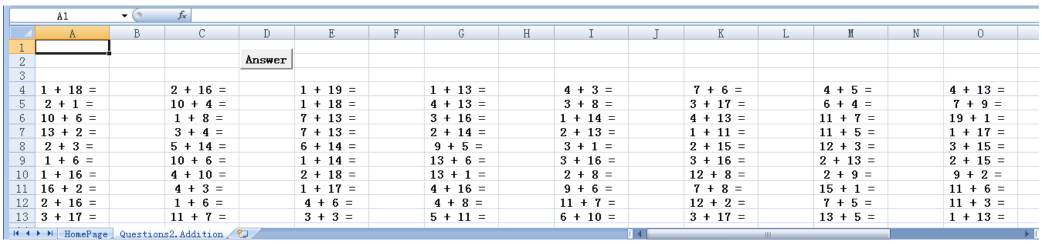
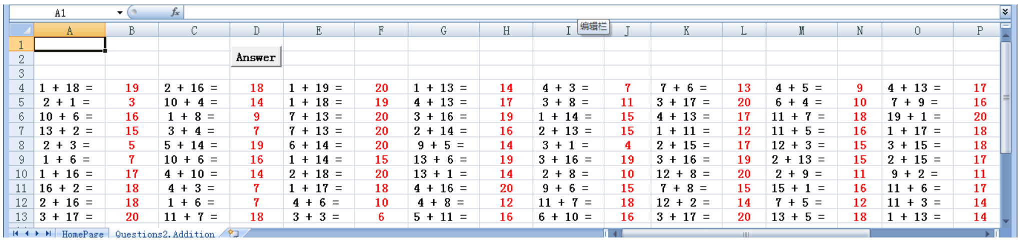


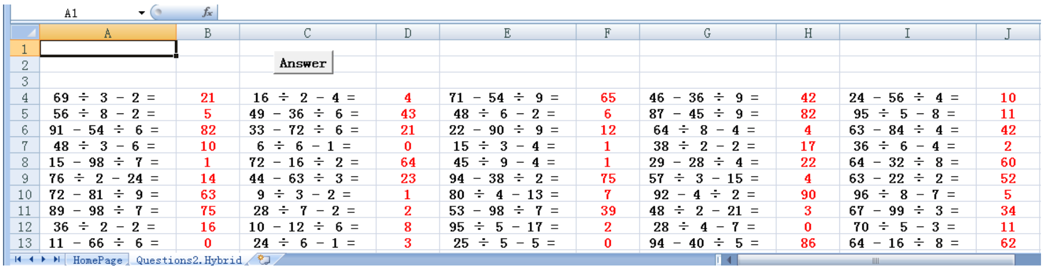
Figure 2. Built Paper after Clicking the "Addition" Button on the Homepage

The child can be tested individually in his/her school or at home with the printed version paper or in front of the computer screen. The child can use any method he/she wants to figure out the answer and is given as much time as they needed or in a certain minute session. Of course a limited duration of time is much better, for such a similarity to standardized achievement tests helps the child to become proficient with the paper-and-pencil situation, rather than performing procedures in a rote and meaningless fashion. After he/she completes all the arithmetic problems, the experimenter can execute the answers by a simple click on the page where the problems locate to allow the child to check them out himself/herself, as depicted in Figure 3. More results can be found in Figure 4, Figure 5 and Figure 6, where the hybrid combinations of four-term arithmetic problems are illustrated.



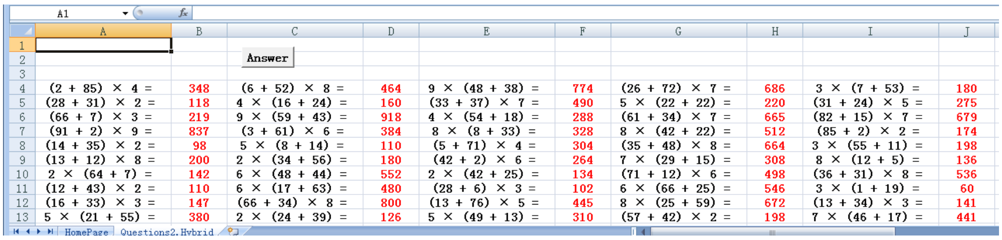
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
4	1 + 18 =	19	2 + 16 =	18	1 + 19 =	20	1 + 13 =	14	4 + 3 =	7	7 + 6 =	13	4 + 5 =	9	4 + 13 =	17
5	2 + 1 =	3	10 + 4 =	14	1 + 18 =	19	4 + 13 =	17	3 + 8 =	11	3 + 17 =	20	6 + 4 =	10	7 + 9 =	16
6	10 + 6 =	16	1 + 8 =	9	7 + 13 =	20	3 + 16 =	19	1 + 14 =	15	4 + 13 =	17	11 + 7 =	18	19 + 1 =	20
7	13 + 2 =	15	3 + 4 =	7	7 + 13 =	20	2 + 14 =	16	2 + 13 =	15	1 + 11 =	12	11 + 5 =	16	1 + 17 =	18
8	2 + 3 =	5	5 + 14 =	19	6 + 14 =	20	9 + 5 =	14	3 + 1 =	4	2 + 15 =	17	12 + 3 =	15	3 + 15 =	18
9	1 + 6 =	7	10 + 6 =	16	1 + 14 =	15	13 + 6 =	19	3 + 16 =	19	3 + 16 =	19	2 + 13 =	15	2 + 15 =	17
10	1 + 16 =	17	4 + 10 =	14	2 + 18 =	20	13 + 1 =	14	2 + 8 =	10	12 + 8 =	20	2 + 9 =	11	9 + 2 =	11
11	16 + 2 =	18	4 + 3 =	7	1 + 17 =	18	4 + 16 =	20	9 + 6 =	15	7 + 8 =	15	15 + 1 =	16	11 + 6 =	17
12	2 + 16 =	18	1 + 6 =	7	4 + 6 =	10	4 + 8 =	12	11 + 7 =	18	12 + 2 =	14	7 + 5 =	12	11 + 3 =	14
13	3 + 17 =	20	11 + 7 =	18	3 + 3 =	6	5 + 11 =	16	6 + 10 =	16	3 + 17 =	20	13 + 5 =	18	1 + 13 =	14

Figure 3. Auto Solving after Clicking the "Answer" Button for the Problems in Figure 2



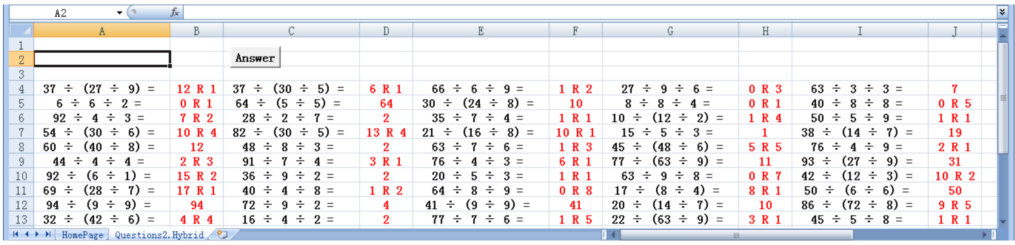
	A	B	C	D	E	F	G	H	I	J
4	69 ÷ 3 − 2 =	21	16 ÷ 2 − 4 =	4	71 − 54 ÷ 9 =	65	46 − 36 ÷ 9 =	42	24 − 56 ÷ 4 =	10
5	56 ÷ 8 − 2 =	5	49 − 36 ÷ 6 =	43	48 ÷ 6 − 2 =	6	87 − 45 ÷ 9 =	82	95 ÷ 5 − 8 =	11
6	91 − 54 ÷ 6 =	82	33 − 72 ÷ 6 =	21	22 − 90 ÷ 9 =	12	64 ÷ 8 − 4 =	4	63 − 84 ÷ 4 =	42
7	48 ÷ 3 − 6 =	10	6 ÷ 6 − 1 =	0	15 ÷ 3 − 4 =	1	38 ÷ 2 − 2 =	17	36 ÷ 6 − 4 =	2
8	15 − 98 ÷ 7 =	1	72 − 16 ÷ 2 =	64	45 ÷ 9 − 4 =	1	29 − 28 ÷ 4 =	22	64 − 32 ÷ 8 =	60
9	76 ÷ 2 − 24 =	14	44 − 63 ÷ 3 =	23	94 − 38 ÷ 2 =	75	57 ÷ 3 − 15 =	4	63 − 22 ÷ 2 =	52
10	72 − 81 ÷ 9 =	63	9 ÷ 3 − 2 =	1	80 ÷ 4 − 13 =	7	92 − 4 ÷ 2 =	90	96 ÷ 8 − 7 =	5
11	89 − 98 ÷ 7 =	75	28 ÷ 7 − 2 =	2	53 − 98 ÷ 7 =	39	48 ÷ 2 − 21 =	3	67 − 99 ÷ 3 =	34
12	36 ÷ 2 − 2 =	16	10 − 12 ÷ 6 =	8	95 ÷ 5 − 17 =	2	28 ÷ 4 − 7 =	0	70 ÷ 5 − 3 =	11
13	11 − 66 ÷ 6 =	0	24 ÷ 6 − 1 =	3	25 ÷ 5 − 5 =	0	94 − 40 ÷ 5 =	86	64 − 16 ÷ 8 =	62

Figure 4. Paper Building and Auto Solving for the Problems of Four-Term Hybrid Operations (Division First, and Subtraction Next, with Aliquot Result Required)



	A	B	C	D	E	F	G	H	I	J
4	(2 + 85) × 4 =	348	(6 + 52) × 8 =	464	9 × (48 + 38) =	774	(26 + 72) × 7 =	686	3 × (7 + 53) =	180
5	(28 + 31) × 2 =	118	4 × (16 + 24) =	160	(33 + 37) × 7 =	490	5 × (22 + 22) =	220	(31 + 24) × 5 =	275
6	(66 + 7) × 3 =	219	9 × (59 + 43) =	918	4 × (54 + 18) =	288	(61 + 34) × 7 =	665	(82 + 15) × 7 =	679
7	(91 + 2) × 9 =	837	(3 + 61) × 6 =	384	8 × (8 + 33) =	328	8 × (42 + 22) =	512	(85 + 2) × 2 =	174
8	(14 + 35) × 2 =	98	5 × (8 + 14) =	110	(5 + 71) × 4 =	304	(35 + 48) × 8 =	664	3 × (55 + 11) =	198
9	(13 + 12) × 8 =	200	2 × (34 + 56) =	180	(42 + 2) × 6 =	264	7 × (29 + 15) =	308	8 × (12 + 5) =	136
10	2 × (64 + 7) =	142	6 × (48 + 44) =	552	2 × (42 + 25) =	134	(71 + 12) × 6 =	498	(36 + 31) × 8 =	536
11	(12 + 43) × 2 =	110	6 × (17 + 63) =	480	(28 + 6) × 3 =	102	6 × (66 + 25) =	546	3 × (1 + 19) =	60
12	(16 + 33) × 3 =	147	(66 + 34) × 8 =	800	(13 + 76) × 5 =	445	8 × (25 + 59) =	672	(13 + 34) × 3 =	141
13	5 × (21 + 55) =	380	2 × (24 + 39) =	126	5 × (49 + 13) =	310	(57 + 42) × 2 =	198	7 × (46 + 17) =	441

Figure 5. Paper Building and Auto Solving for the Problems of Four-Term Hybrid Operations (Addition First, and Multiplication Next)



	A	B	C	D	E	F	G	H	I	J
4	37 ÷ (27 ÷ 9) =	12 R 1	37 ÷ (30 ÷ 5) =	6 R 1	66 ÷ 6 ÷ 9 =	1 R 2	27 ÷ 9 ÷ 6 =	0 R 3	63 ÷ 3 ÷ 3 =	7
5	6 ÷ 6 ÷ 2 =	0 R 1	64 ÷ (5 ÷ 5) =	64	30 ÷ (24 ÷ 8) =	10	8 ÷ 8 ÷ 4 =	0 R 1	40 ÷ 8 ÷ 8 =	0 R 5
6	92 ÷ 4 ÷ 3 =	7 R 2	28 ÷ 2 ÷ 7 =	2	35 ÷ 7 ÷ 4 =	1 R 1	10 ÷ (12 ÷ 2) =	1 R 4	50 ÷ 5 ÷ 9 =	1 R 1
7	54 ÷ (30 ÷ 6) =	10 R 4	82 ÷ (30 ÷ 5) =	13 R 4	21 ÷ (16 ÷ 8) =	10 R 1	15 ÷ 5 ÷ 3 =	1	38 ÷ (14 ÷ 7) =	19
8	60 ÷ (40 ÷ 8) =	12	48 ÷ 8 ÷ 3 =	2	63 ÷ 7 ÷ 6 =	1 R 3	45 ÷ (48 ÷ 6) =	5 R 5	76 ÷ 4 ÷ 9 =	2 R 1
9	44 ÷ 4 ÷ 4 =	2 R 3	91 ÷ 7 ÷ 4 =	3 R 1	76 ÷ 4 ÷ 3 =	6 R 1	77 ÷ (63 ÷ 9) =	11	93 ÷ (27 ÷ 9) =	31
10	92 ÷ (6 ÷ 1) =	15 R 2	36 ÷ 9 ÷ 2 =	2	20 ÷ 5 ÷ 3 =	1 R 1	63 ÷ 9 ÷ 8 =	0 R 7	42 ÷ (12 ÷ 3) =	10 R 2
11	69 ÷ (28 ÷ 7) =	17 R 1	40 ÷ 4 ÷ 8 =	1 R 2	64 ÷ 8 ÷ 9 =	0 R 8	17 ÷ (8 ÷ 4) =	8 R 1	50 ÷ (6 ÷ 6) =	50
12	94 ÷ (9 ÷ 9) =	94	72 ÷ 9 ÷ 2 =	4	41 ÷ (9 ÷ 9) =	41	20 ÷ (14 ÷ 7) =	10	86 ÷ (72 ÷ 8) =	9 R 5
13	32 ÷ (42 ÷ 6) =	4 R 4	16 ÷ 4 ÷ 2 =	2	77 ÷ 7 ÷ 6 =	1 R 5	22 ÷ (63 ÷ 9) =	3 R 1	45 ÷ 5 ÷ 8 =	1 R 1

Figure 6. Paper Building and Auto Solving for the Problems of Four-Term Hybrid Operations (Continued Division, and Aliquot Result was not Required)

5. Conclusion

This study examines the school-aged children education of primary arithmetic fact solving competence by exploring a state-of-the-art auto problem building and evaluating model. A well-defined programming algorithm, which involved, from simple to complex, calculation of three-term arithmetic operation facts including addition, subtraction, multiplication and division associated with four-term hybrid combinations, was first proposed. An auto operation facts building system on computer was then developed by using Excel VBA. The results show that the underlying model makes the building highly effective and practically sound.

The following are some of the significant benefits of the proposed model: 1) it is of practical ease due to its highly integrated and encapsulated by using Excel VBA; 2) it is efficient for the experimenters due to its one-click building; 3) it is effective for the participants due to its auto evaluation to the problem; 4) it is particularly convenient for the experimenters to build or for the participants to perform plenty of (refers to non-repeated fact problems between two consecutive button clicks) arithmetical exercises on different types of calculation due to its randomized operands and customized settings; and 5) it is practically sound since the fact problems are specially and elaborately built, which includes: (a) no operand in either addition or subtraction problem has 0 as digits since it needs no computation; (b) specific numbers like 1, 10 or 100 are not be used (or frequently used) as operand in neither multiplication nor division problem (as divisor) since it is too much simple, expect for the situation in hybrid combination that no other operand can be found (*e.g.*, for the case of “a ÷ b ÷ c” with aliquot result required, if “a” is 5 and “b” is 5, then “c” has to be 1 since there is no other choice).

There are several issues that still need to be further emphasized. First, although the model has been built to account for simple arithmetic, it can be generalized to complex arithmetic or five-term (or more) hybrid combinations research. Second, it appears that the underlying model makes the building highly effective and practically sound, there are, however, areas needing future attention, which include the following directions: (a) digits are not repeated in the same unit, decade, or hundred positions across operands (like in 346 +149 or 346 + 376); (b) no digits are repeated within operands (*i.e.*, like in 344 +659); (c) no reverse orders of operands are used in one paper (*i.e.*, if 383 + 458 is used, 458 +383 can not be used); (d) it can be customized whether carry-over operation involved with one position can be also involved with another contiguous position (*e.g.*, if decade position involves carry-over operation, unit or hundred position also involves, such like 645 + 187); (e) since adding and subtracting the same quantity (*e.g.*, 17 + 11 - 11 = □) leaves the initial quantity unchanged and so no computation is necessary, such operation facts should be discarded; (f) the missing element of problems concerned in this study was on the right-hand side, whereas problems with the a-term missing (*e.g.*, □ + 11 = 43) should be considered in future’s work; and (g) the addend and subtrahend (or sum of subtrahends in four or more term problems) should differ by at least 3, to prevent children from decomposing the problem and using inversion, or using finger-counting strategies. All these will be part of our future work.

Acknowledgements

This work was partially supported by the Special Funds of Undergraduate Course Center of Educational Reform Research Project of Jinan University in 2014.

References

[1] N. C. Jordan, L. B. Hanich and D. Kaplan, "Arithmetic fact mastery in young children: A longitudinal investigation", Journal of Experimental Child Psychology, vol. 85, no. 2, **(2003)**, pp. 103-119.
[2] L. B. Hanich, N. C. Jordan, D. Kaplan and J. Dick, "Performance across different areas of mathematical cognition in children with learning difficulties", Journal of Educational Psychology, vol. 93, no. 3, **(2001)**, pp. 615-626.
[3] N. C. Jordan and L. B. Hanich, "Mathematical thinking in second-grade children with different forms of ld", Journal of Learning Disabilities, vol. 33, no. 6, **(2000)**, pp. 567-578.
[4] N. C. Jordan and T. O. Montani, "Cognitive arithmetic and problem solving a comparison of children with specific and general mathematics difficulties", Journal of Learning Disabilities, vol. 30, no. 6, **(1997)**, pp. 624-634.
[5] K. M. Robinson, J. E. Ninowski and M. L. Gray, "Children's understanding of the arithmetic concepts of inversion and associativity", Journal of Experimental Child Psychology, vol. 94, no. 4, **(2006)**, pp. 349-362.
[6] I. Imbo, A. Vandierendonck and S. De Rammelaere, "The role of working memory in the carry operation of mental arithmetic: Number and value of the carry", The Quarterly Journal of Experimental Psychology, vol. 60, no. 5, **(2007)**, pp. 708-731.
[7] K. M. Cauley, "Children's construction of the logic of arithmetic", Journal of Applied Developmental Psychology, vol. 13, no. 1, **(1992)**, pp. 65-74.
[8] K. M. Cauley, "Construction of logical knowledge: Study of borrowing in substraction", Journal of Educational Psychology, vol. 80, no. 2, **(1988)**, pp. 202-205.
[9] H. H. Aksu, "The views of primary school mathematics teachers over alternative methods of measurement and evaluation", Energy Education Science and Technology Part B: Social and Educational Studies, vol. 4, no. 4, **(2012)**, pp. 2047-2056.
[10] R. Yesil, "Primary education i And ii Level senior students' levels of fulfilling their learning responsibilities", Energy Education Science and Technology Part B: Social and Educational Studies, vol. 5, no. 2, **(2013)**, pp. 915-924.
[11] C. K. Gilmore and P. Bryant, "Can children construct inverse relations in arithmetic? Evidence for individual differences in the development of conceptual understanding and computational skill", British Journal of Developmental Psychology, vol. 26, no. 3, **(2008)**, pp. 301-316.
[12] B. Vilette, "Do young children grasp the inverse relationship between addition and subtraction? Evidence against early arithmetic", Cognitive Development, vol. 17, no. 3, **(2002)**, pp. 1365-1383.
[13] P. Lemaire and M. Lecacheur, "Children's strategies in computational estimation", Journal of Experimental Child Psychology, vol. 82, no. 4, **(2002)**, pp. 281-304.
[14] P. Lemaire, M. Lecacheur and F. Farioli, "Children's strategy use in computational estimation", Canadian Journal of Experimental Psychology, vol. 54, no. 2, **(2000)**, pp. 141-148.
[15] S. Dehaene, E. Spelke, P. Pinel, R. Stanescu and S. Tsivkin, "Sources of mathematical thinking", Behavioral and brain-imaging evidence, Science, vol. 284, no. 5416, **(1999)**, pp. 970-974.
[16] P. Lemaire and S. Callies, "Children's strategies in complex arithmetic", Journal of Experimental Child Psychology, vol. 103, no. 1, **(2009)**, pp. 49-65.
[17] M. Laupa, "Children's understanding of logical and conventional rules in arithmetic algorithms", The Journal of Mathematical Behavior, vol. 19, no. 3, **(2000)**, pp. 291-305.
[18] L. Farrington-Flint, S. Vanuxem-Cotterill and J. Stiller, "Patterns of problem-solving in children's literacy and arithmetic", British Journal of Developmental Psychology, vol. 27, no. 4, **(2009)**, pp. 815-834.
[19] M. Fabbri, V. Natale and A. Adan, "Effect of time of day on arithmetic fact retrieval in a number-matching task", Acta Psychologica, vol. 127, no. 2, **(2008)**, pp. 485-490.
[20] J. I. Campbell and V. A. Thompson, "Retrieval-induced forgetting of arithmetic facts", Journal of Experimental Psychology - Learning Memory and Cognition, vol. 38, no. 1, **(2012)**, pp. 118-129.
[21] M. Fabbri, "Spatial congruency between stimulus presentation and response key arrangements in arithmetic fact retrieval", American Journal of Psychology, vol. 124, no. 3, **(2011)**, pp. 325-340.
[22] M. S. van Galen and P. Reitsma, "Transfer effects in children's recall of arithmetic facts", Cognition and Instruction, vol. 29, no. 3, **(2011)**, pp. 330-348.
[23] A. A. Pyke and J.-A. LeFevre, "Calculator use need not undermine direct-access ability: The roles of retrieval, calculation, and calculator use in the acquisition of arithmetic facts", Journal of Educational Psychology, vol. 103, no. 3, **(2011)**, pp. 607-616.
[24] M. A. Barnes, M. Wilkinson, E. Khemani, A. Boudesquie, M. Dennis and J. M. Fletcher, "Arithmetic processing in children with spina bifida calculation accuracy, strategy use, and fact retrieval fluency", Journal of Learning Disabilities, vol. 39, no. 2, **(2006)**, pp. 174-187.
[25] M. M. Mazzocco, K. T. Devlin and S. J. McKenney, "Is it a fact? Timed arithmetic performance of children with mathematical learning disabilities (mld) varies as a function of how mld is defined", Developmental Neuropsychology, vol. 33, no. 3, **(2008)**, pp. 318-344.
[26] A. De Visscher and M.-P. Noël, "A case study of arithmetic facts dyscalculia caused by a hypersensitivity-to-interference in memory", Cortex, vol. 49, no. 1, **(2013)**, pp. 50-70.

[27] L. Zaunmüller, F. Domahs, K. Dressel, J. Lonnemann, E. Klein, A. Ischebeck and K. Willmes, "Rehabilitation of arithmetic fact retrieval via extensive practice: A combined fmri and behavioural case-study", Neuropsychological Rehabilitation, vol. 19, no. 3, **(2009)**, pp. 422-443.
[28] R. H. Grabner, D. Ansari, K. Koschutnig, G. Reishofer, F. Ebner and C. Neuper, "To retrieve or to calculate? Left angular gyrus mediates the retrieval of arithmetic facts during problem solving", Neuropsychologia, vol. 47, no. 2, **(2009)**, pp. 604-608.
[29] R. H. Grabner, A. Ischebeck, G. Reishofer, K. Koschutnig, M. Delazer, F. Ebner and C. Neuper, "Fact learning in complex arithmetic and figural-spatial tasks: The role of the angular gyrus and its relation to mathematical competence", Human Brain Mapping, vol. 30, no. 9, **(2009)**, pp. 2936-2952.
[30] B. Prieto-Corona, M. Rodríguez-Camacho, J. Silva-Pereyra, E. Marosi, T. Fernández and V. Guerrero, "Event-related potentials findings differ between children and adults during arithmetic-fact retrieval", Neuroscience Letters, vol. 468, no. 3, **(2010)**, pp. 220-224.
[31] B. De Smedt and B. Boets, "Phonological processing and arithmetic fact retrieval: Evidence from developmental dyslexia", Neuropsychologia, vol. 48, no. 14, **(2010)**, pp. 3973-3981.
[32] K. Jost, P. H. Khader, M. Burke, S. Bien and F. Rösler, "Frontal and parietal contributions to arithmetic fact retrieval: A parametric analysis of the problem-size effect", Human Brain Mapping, vol. 32, no. 1, **(2011)**, pp. 51-59.

Author

Tianxiang Zheng, born in 1979, Ph.D, Associate Professor. His research interests focus on tourism e-commerce, information science in tourism applications. E-mail: zheng_tx@jnu.edu.cn.

