Journal of Numerical Cognition
jnc.psychopen.eu | 2363-8761

PsychOpen GOLD

Theoretical Contributions

The Complexity of Mental Integer Addition

Stefan Buijsman[a], Markus Pantsar*[b]

[a] Institutet för Framtidsstudier, Stockholm, Sweden. **[b]** Department of Philosophy, History and Art, University of Helsinki, Helsinki, Finland.

Abstract

An important paradigm in modeling the complexity of mathematical tasks relies on computational complexity theory, in which complexity is measured through the resources (time, space) taken by a Turing machine to carry out the task. These complexity measures, however, are asymptotic and as such potentially a problematic fit when descriptively modeling mathematical tasks that involve small inputs. In this paper, we argue that empirical data on human arithmetical cognition implies that a more fine-grained complexity measure is needed to accurately study mental arithmetic tasks. We propose a computational model of mental integer addition that is sensitive to the relevant aspects of human arithmetical ability. We show that this model necessitates a two-part complexity measure, since the addition tasks consists of two qualitatively different stages: retrieval of addition facts and the (de)composition of multidigit numbers. Finally, we argue that the two-part complexity measure can be developed into a single response-time measure with the help of empirical study of the two stages.

Keywords: integer addition, complexity, mental arithmetic, cognitive modeling

Journal of Numerical Cognition, 2020, Vol. 6(1), 148–163, https://doi.org/10.5964/jnc.v6i1.218

Received: 2019-04-05. Accepted: 2019-07-17. Published (VoR): 2020-06-15.

*Corresponding author at: Department of Philosophy, History and Art, University of Helsinki, P.O. Box 24, Unioninkatu 40A, 00014 Helsinki, Finland. E-mail: markus.pantsar@gmail.com

Not all integer addition tasks are equally complex. Solving 3 + 4 appears to be a less complex task than solving 16 + 24, which in turn seems to be less complex than solving 635 + 242. As a first observation, it is easy to see that as the numbers become larger, arithmetical tasks become more complex. However, while this intuitive idea may work as a rough guideline for assessing the complexity of algorithms that solve addition tasks, a more exact measure of complexity is clearly needed. One approach for introducing such measures is based on the notion of computational complexity taken in theoretical computer science, in which one estimates the resources needed by a Turing machine to solve the problem via a given method. The basis of using this computational approach in cognitive science is the idea that a cognitive process can be modeled as a computation of a specific output for a particular input (e.g., Anderson, 1990; Marr, 1982; for discussion, see Szymanik & Verbrugge, 2018; van Rooij, 2008). Under this view, the complexity of cognitive processes can then be characterized in terms of the computational complexity of the functions mapping the output/input-functions (see e.g., Frixione, 2001; for critical discussion, see Fabry & Pantsar, 2019; Pantsar, 2019). In this way, the study of computational resources in complexity theory can be used in determining the complexity of cognitive processes.

In the case of the addition of integers, these computational resources are not limited to a finite domain. Therefore, the complexity measures that are used are *asymptotic*: they describe the limiting behavior of functions as the arguments approach infinity (or a particular value). The standard is to use the "Big O" notation, which measures the complexity of a problem-solving method as a function of the size of the input. The idea behind the notation is that as the arguments get closer to infinity (or the particular value), the complexity of algorithms is mostly determined by one term in the function. For example, the multiplication of integers with the standard schoolbook long multiplication method takes $n^2 + k$ resources for two n-digit numbers, where k is some constant factor. But as the numbers become larger, the influence of the constant on the resources needed to compute the function becomes smaller. Thus, in the Big O notation, the constant is dropped and the complexity of multiplying integers using the schoolbook method is said to be $O(n^2)$. Similarly, the complexity of addition using the schoolbook method (including carries) is $O(n)$.

The most common measures for the computational complexity of problem-solving methods are time and space. The latter is usually determined in terms of the memory resources needed in the computation. The former is characterized in terms of the number of elementary computational operations needed, on the assumption that the operations take a constant time to be carried out (of course this doesn't hold in practice, especially if one considers different types of operations—as one has to for mental addition). Hence the computational resources are often simply characterized by the number of computational steps required in the algorithm (see, e.g., Aaronson, 2012). Understood in this way, integer multiplication via the schoolbook method takes roughly n^2 steps, and integer addition via the schoolbook method takes roughly n computational steps.

For cognitive scientists and philosophers, the computational complexity measures have given a way to limit the class of possible functions that can model human cognitive capacities. Based on *Cobham's thesis* (Cobham, 1964), in theoretical computer science it is widely accepted that the class of tractable (or efficiently solvable) functions consists of those functions that can be solved by a deterministic Turing machine in polynomial time (forming the complexity class P). This thesis has been applied to cognitive capacities as the *tractable cognition* thesis, stating that only functions in P can plausibly model human cognition computationally (Frixione, 2001; van Rooij, 2008). This use of computational complexity measures to characterize the cognitive tasks would seem to be extendable to a general analysis of the complexity of the tasks. In case of mental arithmetical operations, for example, it initially seems plausible to make the case that multiplication via the schoolbook method is of complexity $O(n^2)$ and addition via the schoolbook method of complexity $O(n)$.

However, the fact that the Big O notation denotes asymptotic behavior makes those complexity measures imprecise characterizations of the methods we use when solving arithmetical problems without using external tools. While the Big O notation is still applicable to the algorithms used in mental arithmetic, it lacks the detail necessary to describe all the differences between addition problems that are solved in this way, primarily because mental arithmetic is limited to small inputs.[i] These details can be important, for example for empirical research on the choice of mental calculation strategies (e.g., Fagginger Auer, Hickendorff, & van Putten, 2018; Verschaffel, Luwel, Torbeyns, & Van Dooren, 2009). Such research assumes that some strategies are more optimal than others, but so far there is no precise characterization of this intuitive idea. A detailed complexity measure for different mental strategies would be valuable in this regard.

The problem is, the Big O notation is not sensitive enough for such research. When we say that schoolbook addition is of the complexity $O(n)$, we mean that for two n-digit numbers it takes $n + k$ computational steps, *k* being some constant factor. In studying asymptotic computational complexity, it is understandable that the constant factor is ignored, since the effect of *k* on the total computational load becomes smaller as *n* becomes larger. For large enough *n*, the effect of *k* becomes negligible. It should be noted that the Big O notation does allow for a more precise treatment, as one can specify the minimum, maximum and average complexity of, for example, addition according to the schoolbook method. The minimum complexity in this case occurs when there are zero carries, and calculating the addition then requires *n* computational steps. The maximum complexity occurs when there are *n* carries, and is therefore $2n-1$ computational steps. We can also determine that the average complexity is $3n/2$. But even with these additional measures, there is relatively little sensitivity to the differences in complexity that occur between small inputs.

Because human mental arithmetic is only used for very small inputs, the constant *k* is non-trivial, and differences between cases are more pronounced. For example, the addition 23 + 42 can be computed without carrying, thus requiring two addition steps, 3 + 2 and 2 + 4. The addition 28 + 37, although resulting in the same value, includes an extra addition step. After adding 8 + 7, the 1 must be carried to the next addition 1 + (2 + 3), thus resulting in three addition steps. In computational complexity analysis, the carrying is included in the constant *k*. But as our examples show, depending on the numbers that are being added, the constant can represent a 50% increase in the number of computational steps. Such an increase can hardly be ignored in an empirically accurate description of the complexity of the mental arithmetic task. If our aim is to characterize the complexity of mentally solving arithmetical problems accurately, then our complexity measure needs to be sensitive to such differences.

For a completely (descriptively) accurate complexity measure of mental arithmetic one needs to account for a number of cognitive effects that are central to (theoretical) debates in the empirical literature. The main body of this paper is structured around this list of effects, which we try to accommodate into our new complexity measure. First, there is a cognitive difference between additions with zero and additions that do not involve zeroes (Brysbaert, Fias, & Noël, 1998 found response times are roughly 10% higher to, for example, 21 + 4 compared to 20 + 4). Second, there is a carry effect, which means that additions that involve a carry take more time (Ashcraft & Stazyk, 1981; Nuerk, Moeller, & Willmes, 2015; Brysbaert et al., 1998 recorded a 50% increase in response times compared to additions with no carry or zeroes). Furthermore, there is a problem-size effect, where response times increase for single-number additions as the outcome of the addition increases (e.g., 3 + 4 took on average 10% longer to compute than 1 + 2 in Barrouillet & Thevenot, 2013). Finally, there are more general influences on the complexity of mental addition such as an individual's math abilities and working memory capacity (Imbo, Vandierendonck, & De Rammelaere, 2007 saw response times almost double when participants had to concurrently perform a second task that engaged the central executive, as per the model of working memory in Baddeley & Hitch, 1974). All of these effects are left unspecified in the computational complexity measures, yet they can have highly significant effects on the cognitive complexity of the task. We discuss the effects specific to addition in section 2 of the paper, including how our proposed simple measure manages to capture all but the problem-size effect. We suggest how a more involved measure could be sensitive even to this latter effect, but point to difficulties in developing it that makes us prefer to present a simpler measure in this paper. Finally, we discuss the general influences and how their effect is reflected in our model in section 3, along with general properties and limitations of our proposed complexity measure.

In what follows, then, we introduce a more fine-grained complexity measure of integer addition as performed by human subjects. This is further supported by a computational model from which the complexity of an arbitrary problem can be inferred. This allows us to present a complexity measure which is sensitive to more than just the input size of an addition task. As a result, it is more precise than the asymptotic approach, but this comes at a cost. The Big O notation gives an estimate of the complexity (possibly refined into a maximum, minimum and average complexity) irrespective of the content of the input. The complexity measure enabled by our model, on the other hand, uses the content of the input to determine the complexity of solving that specific problem using the methods of mental arithmetic. This makes the measure more descriptively accurate, primarily because it uses more information about the problems to be solved.

A More Sensitive Complexity Measure

In order to give a more detailed complexity measure for addition problems we need a different setup for the units used in the complexity measure. As mentioned above, in standard complexity theory this is often simply the number of computational steps (Aaronson, 2012). While this works well for Turing machines, it can be less suitable for modeling human problem-solving processes. The upshot of this approach is that the potential differences between different multiplication and addition tasks are ignored, with the focus being on the size of the input. Essentially, every addition task is equally complex as long as the sizes of the input are the same. However, the tasks performed in the course of human mental calculation can differ significantly, which is why a feasible complexity measure sensitive to such differences should be designed. The sizes of input may be the same, yet the cognitive tasks involved of different complexity, as we will show in the case of integer addition. In order to be able to introduce such a more fine-grained complexity measure, we need to establish groundwork on a computational model of integer addition, as actually performed by human subjects.

There is important evidence that mathematically competent human problem solvers, when confronted with an integer addition problem, always proceed in roughly the same way (see Nuerk et al., 2015 for an extensive review of the cognitive science literature). First, they decompose the numerals that are perceived. For example, 24 is decomposed into 2 and 4. Then, they perform separate additions for the individual numerals. 24 + 32 is solved by performing 2 + 3 and 4 + 2 after the decomposition step. Finally, the separate results are recomposed in order to arrive at the correct answer. If necessary, carries are also computed.[ii]

Cognitive scientists are fairly certain that numerals are decomposed in this way by arithmetically competent people, and that this happens automatically (García-Orza & Damas, 2011; Moeller, Huber, Nuerk, & Wilmes, 2011; Nuerk et al., 2015). Given that the numerals are processed in this decomposed form, addition operations are always performed with single digits. This fits with suggestions about the way in which addition facts are stored (Butterworth, Zorzi, Girelli, & Jonckheere, 2001), as well as experiments that suggest that other mental arithmetic operations such as multiplication are executed with single digits (Domahs, Delazer, & Nuerk, 2006; Domahs et al., 2007; Verguts & De Moor, 2005).

The empirical evidence thus suggests that there are two steps relevant for the complexity of mental addition problems: the composition steps, in which the given numerals are decomposed and the solution is recomposed, as well as addition steps, in which addition facts are retrieved to perform the calculations. That being said, it is important to note that these are the relevant steps for arithmetically educated adults (and older children with arithmetical knowledge and skills). Young children, who in the beginning often resort to finger counting strategies (e.g., Canobi, Reeve, & Pattison, 2003), would require a different complexity measure (if only because there is little or no retrieval of addition facts from memory). In such a case, a “+ 1” step would be the basic measure of complexity, rather than the retrieval of a single-digit addition fact.[iii] It is also plausible that different numeral systems (e.g., different bases) can cause relevant differences in the mental arithmetic processes. Finally, there is some disagreement over whether the single-digit addition operations are indeed based on retrieval strategies. An alternative is that these problems are solved procedurally, but quickly enough that we are not consciously aware of the fact that we actively solve the problem, rather than remember the solution (Fayol & Thevenot, 2012; Barrouillet & Thevenot, 2013). Should this be the case, then one can fill in these sub-procedures in place of the retrieval steps in our model. In the rest of the paper we will refer to the retrieval account, as it is the most common in the literature, but our model can easily be changed if a procedural account turns out to be correct.

So, then, for adult subjects familiar with arithmetic in a decimal number system there appears to be a constant two-stage procedure for completing mental addition tasks. We therefore propose, as the foundation of our more sensitive complexity measure, a measure specified by a pair: (c, a) with c the number of composition steps and a the number of addition operations based on retrieval from memory (or sub-procedures that solve single-digit addition problems). The first part of the measure, c, can be computed in terms of the length of the input. We also make the simplifying assumption that each decomposition step is equally complex. However, the second part of the measure, a, depends on the specific numbers that one has to work with. For simplicity, we assume that each single-digit addition operation is equally complex. We will see that even under that assumption one cannot, in general, compute the number of addition operations purely in terms of the length of the input.

We think that this model is best motivated and explained on the basis of the known differences in complexity from the cognitive science literature. However, we will also discuss several objections one might have to the model in section 3. First, we turn to the peculiar difference in complexity (reaction times and error rates) between 60 + 8 and 61 + 8.

Additions That Involve 0

As Brysbaert et al. (1998) found, it is simpler for humans to compute 60 + 8 than to compute 61 + 8, even though a standard complexity measure could not establish any difference. The length of the input is the same, and the computational problem (natural number addition) is the same, so therefore standard complexity theory doesn’t distinguish between the two. Yet there is a clear difference, as there is no need to retrieve an addition fact in this case. The Arabic numeral system works in such a way that for 0 + 8 one can simply replace the 0 by 8. That is in contrast to 1 + 8, where one does need to retrieve an addition fact, namely 1 + 8 = 9. A similar effect has been found with number comparisons. Decisions which of, for example, 050 and 003 is larger are made faster and with fewer errors than between, for example, 050 and 030 (García-Orza, Estudillo, Calleja, & Rodríguez, 2017; Kallai & Tzelgov, 2012).

For this reason, 60 + 8 has, according to our measure, complexity (2, 0) with 2 composition steps—one to decompose 60 and one to recompose 68—and no addition steps. 61 + 8 has complexity (2, 1) because one needs to retrieve 1 + 8. It is, for that reason, more complex and will take longer.

Yet zeroes do add some complexity to a problem. Compare, for example, the addition tasks 200 + 300 and 2 + 3. Intuitively, the former is a more complex addition problem than the latter. Standard complexity theory agrees with that intuition, since the input for the first is much longer than that for the second. This is also reflected in our complexity measure, though only in the number of composition steps. 2 + 3 does not need to be decomposed and therefore has complexity (0, 1). 200 + 300 does need to be decomposed and recomposed. It requires 4 decomposition steps (if we count them as cutting up the numeral, for example, first we take the last 0 from 200 and then the second zero to arrive at 2, 0, 0). It also needs two steps to recompose the numeral. That gives a complexity of (6, 1). So, we predict that 200 + 300 is more complex than 2 + 3, though that the difference in response times is less than the difference between 2 + 3 and 254 + 289. We are not yet aware of studies that have tested this exact difference, but expect there to be one (which is overlooked by the standard complexity theory measure).

In this way we can, we think correctly, model many apparent differences between addition problems. 200 + 300 is (we predict) more complex than 2 + 3 but less complex than 210 + 310, the difference being that the last problem requires us to retrieve two addition facts, compared with just one for the first two. And, of course, this is (so we predict) again simpler than 214 + 312 which requires three addition steps, making its complexity (6, 3).

Differences in Length of Input

Our complexity measure is more similar to the standard asymptotic complexity measure when no zeroes or carries are involved. 61 + 18, for example, is more complex than 61 + 8. The main reason for this is that one needs to add more numbers together in the case of 61 + 18. 61 + 18 requires two addition steps (6 + 1 and 1 + 8), whereas 61 + 8 only requires one. Furthermore, because 61 + 18 is longer it requires more composition steps. Our model claims that 61 + 18 has complexity (3, 2) and 61 + 8 has complexity (2, 1).

Our account does not differ significantly from standard complexity theory in this sense. The increase in complexity with length is also linear. As long as there are no zeroes, one gets an additional addition step whenever the length of both addends has increased. However, addition steps increase only in that case: there are no extra addition steps when only the length of one of the two addends increases. For example, 6431232351 + 8 has equally many addition steps as 1 + 8.

The only thing that truly increases with length of input in the way codified by standard complexity theory is the number of composition steps. The longer the input numerals, the more effort it takes to separate them out into component parts. However, for both addition steps and composition steps there is an important special case: carrying.

Carrying

Whenever the addition of two single digits gives a result above 9, one needs to carry part of the result to the left. This makes mental addition more complex, as we also know from empirical studies (Ashcraft & Stazyk, 1981; Nuerk et al., 2015). While this is implicitly captured by standard complexity theory, there is no way to explicitly represent the added complexity of carrying in that framework. Our complexity measure, however, registers it explicitly in terms of more addition steps and/or more composition steps.

This touches on an ongoing debate about the reason carries are more complex. By formalizing it as an extra addition step our measure suggests that this is a categorical difference: 18 + 13 is more complex than 18 + 11 not just because the numbers involved are larger (known as the problem-size effect) but because the carry operation itself requires more work. There are, however, studies that found a more gradual increase in complexity (based on response times) dependent on the sum of the single digits being added (e.g., 3 + 4 is more complex than 1 + 4 but less complex than 3 + 6). This could mean that the carry effect is rather an easily observable case of this gradual increase, since the outcomes of single-digit problems with carry are always higher than those without carry—or that it is a combination of this gradual effect and the more abrupt need for additional work in the case of carries (Artemenko, Soltanlou, Ehlis, Nuerk, & Dresler, 2018; Klein et al., 2010). The model presented here, however, does not include this gradual increase, mainly in order to keep the complexity measure as simple as possible (and the debate has not yet ruled out the categorical view of the carry effect). So, we leave aside the gradual increase for now and will return to it in section 3, where we discuss limitations of our model.

Consider first the difference between 65 + 24 and 65 + 26. The latter is more complex, because one needs to carry the result from 5 + 6 to the addition of 6 + 2. What we suspect happens is that the carry is calculated after the two separate additions have already been computed. That is, we compute 6 + 2 and 5 + 6 first, and then perform the carry to 8 + 1. The alternative hypothesis is that we compute the 5 + 6 first, notice the need to carry part of the result, and then compute 6 + 2 + 1. While that is how we often work on paper, mental addition may not follow this second scenario. Eye-tracking studies suggest that the carry is computed after the initial two additions (Moeller et al., 2011).

The carry is, then, a separate addition step. Therefore, 65 + 24 has complexity of (3, 2) whereas 65 + 26 has a complexity of (3, 3). Even if the second scenario is correct, where we compute 6 + 2 + 1 instead of 6 + 2 and 8 + 1, we still think the extra addition step is warranted. In that case, too, we need to retrieve two addition facts, assuming that we only store addition facts involving two addends (as suggested by Butterworth et al., 2001).

This is the scenario where the carry is not in the leftmost digits. If, instead, one compares 65 + 24 with 65 + 44, there is another complexity difference. There is no need for an extra addition step. Instead, the resulting numeral is longer, requiring an additional composition step. 65 + 24 has complexity (3, 2) and 65 + 44 has complexity (4, 2). Of course, those effects could combine. 65 + 36 has complexity (4, 3) because it requires both the extra addition 6 + 3 + 1 and an extra composition step.

A Computational Toy Model of Integer Addition as Performed by Human Subjects

We believe that a computational toy model of the addition of two *i*-digit integers that takes into account the above aspects of integer addition as performed by human subjects can help clarify our proposed measure. For the sake of simplicity we present here only the case when the integers consist of equally many digits.

Let n be an *i*-digit number with the following composition of digits $n = n_i \dots n_2 n_1$.

Let m be an *i*-digit number with the following composition of digits $m = m_i \dots m_2 m_1$.

In the beginning, carry memory is empty. The model computes the sum $n + m = s = s_j s_i \dots s_2 s_1$ in the following way with the input (n, m):

```
1    Decompose $n$ into $n_i, \ldots, n_2, n_1$
2    Decompose $m$ into $m_i, \ldots, m_2, m_1$
3    For $k = 1$ to $i$:
4        Take the pair of digits $(n_k, m_k)$:
5        If $n_k = 0$, then assign $s_k = m_k$
6        If $m_k = 0$, then assign $s_k = n_k$
7        Otherwise:
8            Retrieve addition fact $n_k + m_k$
9            If $n_k + m_k > 9$ then assign rightmost digit to $s_k$
10           Otherwise $s_k = n_k + m_k$
11       If carry memory is 1:
12           Retrieve addition fact $s_k + 1$
13           If $s_k + 1 > 9$ then assign rightmost digit to $s_k$
14           Otherwise $s_k = n_k + m_k$
15       If $n_k + m_k > 9$, or carry memory is 1 and $n_k + m_k + 1 > 9$, then store 1 to carry
         memory.
16       Otherwise, set carry memory to 0
17   End for-loop
18   If carry memory is 1, assign $s_j = 1$
19   Otherwise, assign $s_j = 0$
20   Compose and output the integer $s = s_j s_i \ldots s_2 s_1$
```

Roughly put, the model takes the input and runs the lines 4 to 16 until it gets to the final digits of *n* and *m*. At this point it checks to see if there is a final carry before composing the output (line 20). We can see that this computational model is sensitive to the difference between adding zero and adding two non-zero integers. It is also sensitive to the extra addition steps needed when carrying. Furthermore, it is sensitive to the extra composition step if the sum has more digits than the addend numbers. Due to these features, the model captures aspects of mental integer addition that are not included in the asymptotic computational complexity measures. With our model, one can extract the complexity of a specific addition problem (according to our measure) by counting the number of (de)composition steps and addition fact retrievals one encounters while solving the problem in this computational model. In other words, this offers a model that can be used to automatically produce the complexity of a given addition problem. Consequently, one can also use this to generate predictions about which additions are more complex and which are simpler, that can then be tested empirically.

Basic Properties of the Complexity Measure

As we mentioned in the introduction we do not think that our complexity measure conflicts with the Big O notation. Rather, it gives a more detailed picture for situations in which only small inputs occur. While this does not undermine the validity of standard asymptotic complexity measures in any way, it does show some differences even in terms of the best-case, worst-case and average-case complexity. Recall that these three are, for solving addition problems with the schoolbook method, n, $2n-1$ and $3n/2$, respectively.

In the case of mental arithmetic, the best-case scenario involves zeroes. If the inputs are allowed to be of unequal length, then this occurs when one number consists of 1 digit and the other of *n* digits where the last digit is a zero. The complexity of mental addition in this case is ((*n*−1)*2, 0). With inputs of equal length, the best-case is, instead, when the left-most digits do not lead to a carry and all the other digits are zeroes (e.g., 200 + 300). In such a scenario mentally adding the numbers has complexity ((*n*−1)*3, 1). This is, when one only looks at the number of addition steps, significantly lower than the best-case scenario as described by the asymptotic measure. In the situation without zeroes and with equally-sized inputs, the best-case complexity on our measure is ((*n*−1)*3, *n*)—yielding the same number of addition steps as predicted by the asymptotic complexity measure.

The worst-case scenario is basically the same on either complexity measure. When there are *n* carries, the complexity of a problem when solving it using mental arithmetic is 2*n*–1 according to the standard asymptotic measure and ((*n*−1)*3+1, 2*n*−1) according to our measure. But with the average complexity there is again a difference between the two measures. The average is lower in our complexity measure because any computation involving a zero is easier to solve using mental arithmetic than the asymptotic measure assumes.

Finally, note that both our complexity measure and the standard asymptotic complexity measure are sensitive to the base of the numeral system. By changing the base one changes the length of the inputs, thereby also changing the number of addition steps needed to solve a specific problem. Specifically, if one uses a higher base then there will be fewer addition and composition steps, on balance. On the other hand, it does become necessary to remember more arithmetical facts (which isn't captured by a complexity measure). For example, with base 16 it is necessary to remember that 10 + 2 = 12 (written here in base 10). In other words, on either measure the complexity of problems decreases when the base is increased at the cost of needing to remember more addition facts, though this doesn't affect the general complexity of the method.

Possible Issues With the Model

While we think that our complexity measure captures the more fine-grained distinctions in complexity of addition tasks much better than standard complexity theory, it has several features that require further research. To begin with, the complexity measure is given in terms of a pair of numbers, rather than a single number. This can make it seem impossible to compare two complexity measures when both values are different. Is (6, 1) more complex than (3, 2)? The complexity measure itself doesn't seem to give a clear answer.

However, it is certainly possible to find an answer. The empirical results we have referred to understand complexity primarily in terms of reaction times. A mental addition problem is more complex if the reaction time of participants is longer. If we learn, from more experiments, how long a single composition step takes (on average, or for a particular person) and how long an addition step takes, then every pair can be evaluated to a reaction time. Those reaction times can be compared without any problems. So, even though the complexity measures cannot easily be compared as pairs, with the help of empirical data they can be converted to a format in which they can be compared generally.

In fact, we think this is a highly useful feature of the model rather than a problem. One thing to keep in mind is that the composition steps we need to take are heavily influenced by language (Nuerk et al. 2015; Nuerk, Moeller, Klein, Willmes, & Fischer et al., 2011). If we take into account spoken numerals as well, then languages where those are inverted (such as German, where twenty-four is *vierundzwanzig*, "four-and-twenty") complicate the composition step. Reaction times are longer for languages with inversion and error rates are higher (Barrouillet, Camos, Perruchet, & Seron, 2004; Camos, 2008; Göbel, Moeller, Pixner, Kauffmann, & Nuerk, 2014; Power & Dal Martello, 1990, 1997). On the other hand, languages such as Korean where place values are explicitly mentioned in all cases (e.g., 222 is spoken as two hundred two ten two) also lead to better arithmetic performance, presumably by simplifying the composition step (Fuson & Kwon, 1992). By separating the composition and addition steps we can account for such cultural differences in complexity more easily, especially since they seem to be more pronounced for composition steps than for addition steps.

Another upside of this feature of our model is that it allows an implicit role for the more general influences on the complexity of mental addition mentioned in the introduction: an individual's mathematics skills and working memory capacity. As shown by Hitch (1978), the effect of working memory on mental arithmetic is not a simple phenomenon. Working storage decay in cognitive tasks is in general difficult to model and our model of mental integer addition cannot capture the phenomenon in its full complexity. For example, Hitch (1978, pp. 320–321) presents data that carrying errors do not occur with the same probability in different places (ones, tens, etc.) of adding multi-digit numbers. However, by studying solution times our model can still capture important effects of domain-general factors for integer addition tasks. For a calculation of the response times of a specific individual, these factors will be important influences on the amount of time any one step takes. For example, lower working memory capacity increases the response times on mental addition, especially in the case where carries are present (Imbo et al., 2007). Better math skills can lower response times by reducing the time needed to retrieve addition facts. So, the domain-general features that influence the complexity of mental addition are eventually reflected in the computation of response times of an individual in relation to the number of composition and retrieval steps.

Still, we cannot capture all of the differences between addition problems with our complexity measure. For one thing, there is a difference between 48 + 25 and 45 + 28 in terms of strategy execution (Guillaume, Nys, & Content, 2012). For the first, 4 > 2 and 8 > 5, but for the second 4 > 2 while 5 < 8. The incongruence in the second addition problem has an effect on the order in which participants add the different digits (known as the unit-decade compatibility effect in tasks where participants have to choose the larger of two multi-digit numbers). Since it doesn't seem to affect reaction times in mental addition (Guillaume et al., 2012), we haven't included it.

Perhaps more importantly, there is a suggestion that addition facts are always stored in the format "max + min = …," that is, as 6 + 2 = 8 and never as 2 + 6 = 8 (Butterworth et al., 2001). If so, then 6 + 2 would be less complex than 2 + 6, in the sense that 6 + 2 takes less time to retrieve. These kinds of subtle differences between different addition steps, as well as any possible differences between composition steps, are not currently included in our complexity measure. One could, presumably, build more details into the composition and addition parts of the complexity measure. However, since these effects seem to be relatively small, compared to the difference made by having to carry or being able to make use of a zero, we have not done so.

Furthermore, as already mentioned in section 2.3, our model does not account for the gradual increase in response times that is seen with single-digit additions as the outcome increases. We think that a more complex model could account for this problem-size effect, by more explicitly producing expectations for response times that are influenced by the specific digits involved. However, it will be harder to specify this in the kind of computational model we gave in section 2.4. Yet, if one wants a truly accurate model of how difficult it will be to solve a problem by mental addition, the gradual effect has to be incorporated, probably as a linear function from specific single-digit additions to response times. This could supersede our treatment of the carry effect, although as mentioned earlier, the debate is ongoing whether the extra effort involved in carries can wholly be explained by the gradual increase of the outcome.

Another aspect not included in our computational model and the complexity measure are heuristic rules such as 99 + 32 = (100 – 1) + 32 = 100 + (32 − 1) = 100 + 31 = 131. In practice, many such rules may be used and they can change the amount of addition (and subtraction) steps, composition steps, additions with zero, and carries. A fully accurate computational model of integer addition as performed by human subjects would need to take into account also such rules, which may differ greatly across cultures as well as across individuals.

Finally, our computational model and the relevant complexity measure are not sensitive to some special cases of addition where the result is familiar due to some reason. 25 + 25 = 50 could potentially be an example of a case which does not require explicit carrying, especially for people living in cultures where currency is divided into quarters. Similarly, 64 + 64 = 128 and similar sums of powers of two can be retrievable without calculation for people working with computers.[iv] Perhaps for golfers, 18 + 18 = 36 requires no calculation. The list of such examples could be continued, and in particular cases, the computational model could be adjusted accordingly. What we have wanted to do in this paper is providing a more general computational model and a relevant complexity measure for it. One of the great strengths of the model is that it can be adjusted for further more fine-grained aspects of mental integer addition.

As such, our model is line with the more pluralist conceptual frameworks that have been proposed for the cognition of numbers and mental arithmetic. In the “abstract code” model of McCloskey and others (McCloskey, 1992; McCloskey & Macaruso, 1995), mental arithmetic is thought to be possible by first forming abstract semantic representations of numerals and number symbols, which are then used for calculations. In that model, the differences between numeral systems can be present in the abstraction process, but the calculations are carried on the abstract representations. In contrast, in the “triple-code” model of Dehaene and Cohen (1995), there are thought to be different representations of numerosities (visual symbolic, auditory verbal, and analog magnitude estimations), which have different roles in numerical cognition tasks. This model has been further refined by Campbell and colleagues (Campbell, 1994; Campbell & Epp, 2004), who have proposed the “encoding complex” model to include interactions between the different representations. Campbell (1995) argues that empirical data on common errors in arithmetic, for example, can be explained by the interaction of the different subsystems of numerical cognition.

In this paper, we do not want to commit to any particular theory of cognitive architecture when it comes to number representations. However, it should be noted that compared to standard computational complexity measures, our complexity measure is a better fit with a pluralist model like that of Campbell and colleagues. Our measure is sensitive to differences in number symbol systems and it can be adjusted to account for differences in verbal numeral systems. Visual number fact memory, familiar multi-digit addition facts, and other similar phenomena can be included in the model by adjusting the number of retrieved addition facts and (de)composition steps accordingly.

Conclusion

It is obvious that not all arithmetical problems are equally difficult. Standard complexity theory gives some idea of how hard individual problems are, though it is fairly coarse-grained. We have presented a computational model that enables formulating a more fine-grained complexity measure that closely tracks how difficult addition problems are for human problem solvers. Based on our understanding of the way arithmetically competent humans tackle addition problems, we split up their complexity into two kinds of steps: composition steps and addition (retrieval) steps.

Not only is this model more fine-grained, it also shows that the approach of standard complexity theory is not sufficient to model the complexity of mental addition problems. They can be much simpler than one would expect, for example, because we don’t need to retrieve any addition facts when zeroes are involved. Our more fine-grained measure thus shows that standard complexity theory is not helpful if one wants to know which addition problem is more difficult for a human problem solver.[v]

Finally, while we have only looked at addition, we think that the model can be extended to other types of arithmetical problems. There are similar effects for multiplication (e.g., Domahs et al., 2006, 2007). Our model might also be a good start for modeling the complexity of number comparisons, though there are many more notation-dependent effects for number comparisons than for addition and multiplication (for a review, see Nuerk et al., 2015). Our model has also ignored the use of external tools such as finger counting or pen and paper, which might change the relative complexity of problems.

There is much more work to be done before we understand the complexity of problems for human problem solvers, rather than for a Turing machine. We hope that this complexity measure for addition is a step in the right direction, if only because it shows that there are some cases where one gets a surprising divergence from standard complexity theory. Generally, we believe that more sensitive complexity measures can be used to make empirical theories on mathematical problem solving more accurate. While in many cases undoubtedly fruitful, the use of asymptotic computational complexity measures can lead us astray by disregarding significant factors. We have shown that computational models of arithmetical problem solving tasks such as addition and multiplication, for example, cannot rely only on input sizes. This is a potential problem for models and explanations in the cognitive sciences, but also for (in particular early) mathematics education, where finding a suitable complexity level for problems is a key concern. In the case of integer addition, but also generally in arithmetic, using input size as the exclusive measure of complexity can be misleading for just the kind of small inputs that school children deal with.

Notes

i) The input sizes can of course be increased by using cognitive tools like pen and paper. Much of the account of integer addition in this paper is feasible to use also for pen-and-paper calculations, but this is not necessarily generally the case.

ii) It should be noted that although we speak of human problem solvers as a single group, these results are in fact based on studies almost exclusively on Western test subjects. For future study, the possible cultural differences in mathematical problem solving is an important topic. Indeed, we will see that the model we propose is sensitive to such cultural differences (also within Western cultures, e.g., based on language.).

iii) As discussed below, it might be that the retrieval of addition facts is in fact a very quick, subconscious, adding procedure not unlike finger-counting. If so, then the complexity measures would be more similar: the main measure would be a “+1” step in each case, but the time it takes to execute the step is much lower for adults. Furthermore, very young children would probably not engage in composition steps, since all they do is count on from the lowest/leftmost number. Such problem solving strategies can then be tackled in the same way as we approach adult arithmetic, though it is out of scope to analyze each such strategy here.

iv) We thank an anonymous reviewer for making these points.

v) Our model can be processed by a Turing machine, so we are not suggesting that standard complexity theory could not ultimately include the more fine-grained complexity measure proposed here. As it is now, however, complexity measures for arithmetical operations do not include such considerations.

Funding

Stefan Buijsman's work was partially funded by Vetenskapsrådet grant nr. 2018-01163. Markus Pantsar's work was part of the Academy of Finland project "Dependence and Independence in Logic," led by Professor Gabriel Sandu, decision nr. 286991.

Competing Interests

The authors have declared that no competing interests exist.

Acknowledgments

We would like to thank Regina Fabry for providing valuable comments to an earlier version of this manuscript. We would also like to thank the editor and the two anonymous reviewers for their help in finalizing the paper.

Author Contributions

Both authors have made substantial and direct intellectual contributions to this work in equal terms.

References

Aaronson, S. (2012). Why philosophers should care about computational complexity. In B. J. Copeland, C. J. Posy, & O. Shagrir (Eds.), *Computability: Turing, Gödel, Church, and beyond* (pp. 261-328). London, United Kingdom: MIT Press.

Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates.

Artemenko, C., Soltanlou, M., Ehlis, A.-C., Nuerk, H.-C., & Dresler, T. (2018). The neural correlates of mental arithmetic in adolescents: A longitudinal fNIRS study. *Behavioural and Brain Functions, 14*(1), Article 5. https://doi.org/10.1186/s12993-018-0137-8

Ashcraft, M., & Stazyk, E. (1981). Mental addition: A test of three verification models. *Memory & Cognition, 9*, 185-1196. https://doi.org/10.3758/BF03202334

Baddeley, A. D., & Hitch, G. (1974). Working memory. In G. Bower (Ed.), *The psychology of learning and motivation* (Vol. 8, pp. 47–89). New York, NY, USA: Academic Press.

Barrouillet, P., Camos, V., Perruchet, P., & Seron, X. (2004). ADAPT: A developmental, asemantic, and procedural model for transcoding from verbal to Arabic numerals. *Psychological Review, 111*, 368-394. https://doi.org/10.1037/0033-295X.111.2.368

Barrouillet, P., & Thevenot, C. (2013). On the problem-size effect in small additions: Can we really discard any counting-based account? *Cognition, 128*, 35-44. https://doi.org/10.1016/j.cognition.2013.02.018

Brysbaert, M., Fias, W., & Noël, M. (1998). The Whorfian hypothesis and numerical cognition: Is “twenty-four” processed in the same way as “four-and-twenty”? *Cognition, 66*(1), 51-77. https://doi.org/10.1016/S0010-0277(98)00006-7

Butterworth, B., Zorzi, M., Girelli, L., & Jonckheere, A. R. (2001). Storage and retrieval of addition facts: The role of number comparison. *Quarterly Journal of Experimental Psychology, 54A*, 1005-1029. https://doi.org/10.1080/713756007

Camos, V. (2008). Low working memory capacity impedes both efficiency and learning of number transcoding in children. *Journal of Experimental Child Psychology, 99*(1), 37-57. https://doi.org/10.1016/j.jecp.2007.06.006

Campbell, J. (1994). Architectures for numerical cognition. *Cognition, 53*, 1-44. https://doi.org/10.1016/0010-0277(94)90075-2

Campbell, J. (1995). Mechanisms of simple addition and multiplication: A modified network-interference theory and simulation. *Mathematical Cognition, 1*, 121-164.

Campbell, J., & Epp, L. (2004). An encoding-complex approach to numerical cognition in English-Chinese bilinguals. *Canadian Journal of Experimental Psychology, 58*, 229-224. https://doi.org/10.1037/h0087447

Canobi, K., Reeve, R., & Pattison, P. (2003). Patterns of knowledge in children’s addition. *Developmental Psychology, 39*(3), 521-534. https://doi.org/10.1037/0012-1649.39.3.521

Cobham, A. (1964). The intrinsic computational difficulty of functions. In Y. Bar-Hillel (Ed.), *Proceedings of the 1964 Congress on logic, mathematics and the methodol-ogy of science* (pp. 24–30). Amsterdam, The Netherlands: North Holland.

Dehaene, S., & Cohen, L. (1995). Towards an anatomical and functional model of number processing. *Mathematical Cognition, 1*, 83-120.

Domahs, F., Delazer, M., & Nuerk, H.-C. (2006). What makes multiplication facts difficult. *Experimental Psychology, 53*, 275-282. https://doi.org/10.1027/1618-3169.53.4.275

Domahs, F., Domahs, U., Schlesewsky, M., Ratinckx, E., Verguts, T., Willmes, K., & Nuerk, H.-C. (2007). Neighborhood consistency in mental arithmetic: Behavioral and ERP evidence. *Behavioral and Brain Functions, 3*, Article 66. https://doi.org/10.1186/1744-9081-3-66

Fabry, R. E., & Pantsar, M. (2019). A fresh look at research strategies in computational cognitive science: The case of enculturated mathematical problem solving. *Synthese,* 1-43. https://doi.org/10.1007/s11229-019-02276-9

Fagginger Auer, M., Hickendorff, M., & van Putten, C. (2018). Training can increase students’ choices for written solution strategies and performance in solving multi-digit division problems. *Frontiers in Psychology, 9*, Article 1644. https://doi.org/10.3389/fpsyg.2018.01644

Fayol, M., & Thevenot, C. (2012). The use of procedural knowledge in simple addition and subtraction problems. *Cognition, 123*, 392-403. https://doi.org/10.1016/j.cognition.2012.02.008

Frixione, M. (2001). Tractable competence. *Minds and Machines, 11*(3), 379-397. https://doi.org/10.1023/A:1017503201702

Fuson, K., & Kwon, Y. (1992). Korean children’s understanding of multidigit addition and subtraction. *Child Development, 63*, 491-506. https://doi.org/10.2307/1131494

García-Orza, J., & Damas, J. (2011). Sequential processing of two-digit numbers: Evidence of decomposition from a perceptual number matching task. *Journal of Psychology, 219*(1), 23-29. https://doi.org/10.1027/2151-2604/a000042

García-Orza, J., Estudillo, A., Calleja, M., & Rodríguez, J. (2017). Is place-value processing in four-digit numbers fully automatic? Yes, but not always. *Psychonomic Bulletin Review, 24*, 1906-1914. https://doi.org/10.3758/s13423-017-1231-x

Göbel, S., Moeller, K., Pixner, S., Kauffmann, L., & Nuerk, H.-C. (2014). Language affects symbolic arithmetic in children: The case of number word inversion. *Journal of Experimental Child Psychology, 119*, 17-25. https://doi.org/10.1016/j.jecp.2013.10.001

Guillaume, M., Nys, J., & Content, A. (2012). Is adding 48 + 25 and 45 + 28 the same? How addend compatibility influences the strategy execution in mental addition. *Journal of Cognitive Psychology, 24*, 836-843. https://doi.org/10.1080/20445911.2012.702750

Hitch, G. J. (1978). The role of short-term working memory in mental arithmetic. *Cognitive Psychology, 10*(3), 302-323. https://doi.org/10.1016/0010-0285(78)90002-6

Imbo, I., Vandierendonck, A., & De Rammelaere, S. (2007). The role of working memory in the carry operation of mental arithmetic: Number and value of the carry. *The Quarterly Journal of Experimental Psychology, 60*(5), 708-731. https://doi.org/10.1080/17470210600762447

Kallai, A., & Tzelgov, J. (2012). The place-value of a digit in multi-digit numbers is processed automatically. *Journal of Experimental Psychology: Learning, Memory and Cognition, 38*(5), 1221-1233. https://doi.org/10.1037/a0027635

Klein, E., Moeller, K., Dressel, K., Domahs, F., Wood, G., Willmes, K., & Nuerk, H.-C. (2010). To carry or not to carry—Is this the question? Disentangling the carry effect in multi-digit addition. *Acta Psychologica, 135*, 67-76. https://doi.org/10.1016/j.actpsy.2010.06.002

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. San Francisco, CA, USA: W.H. Freeman and Company.

McCloskey, M. (1992). Cognitive mechanisms in numerical processing: Evidence from acquired dyscalculia. *Cognition, 44*(1–2), 107-157. https://doi.org/10.1016/0010-0277(92)90052-J

McCloskey, M., & Macaruso, P. (1995). Representing and using numerical information. *American Psychologist, 50*(5), 351-363. https://doi.org/10.1037/0003-066X.50.5.351

Moeller, K., Huber, S., Nuerk, H.-C., & Wilmes, K. (2011). Two-digit number processing: Holistic, decomposed or hybrid? A computational modelling approach. *Psychological Research, 75*(4), 290-306. https://doi.org/10.1007/s00426-010-0307-2

Nuerk, H.-C., Moeller, K., Klein, E., Willmes, K., & Fischer, M. (2011). Extending the mental number line—A review of multi-digit number processing. *Journal of Psychology, 219*, 3-22. https://doi.org/10.1027/2151-2604/a000041

Nuerk, H.-C., Moeller, K., & Willmes, K. (2015). Multi-digit number processing: Overview, conceptual clarifications, and language influences. In R. Cohen Kadosh & A. Dowker (Eds.), *The Oxford handbook of numerical cognition* (pp. 106–139). Oxford, United Kingdom: Oxford University Press.

Pantsar, M. (2019). Cognitive and computational complexity: Considerations from mathematical problem solving. *Erkenntnis*. Advance online publication. https://doi.org/10.1007/s10670-019-00140-3

Power, R., & Dal Martello, M. (1990). The dictation of Italian numerals. *Language and Cognitive Processes, 5*, 237-254. https://doi.org/10.1080/01690969008402106

Power, R., & Dal Martello, M. (1997). From 834 to eighty thirty four: The reading of Arabic numerals by seven year old children. *Mathematical Cognition, 3*, 63-85. https://doi.org/10.1080/135467997387489

Szymanik, J., & Verbrugge, R. (2018). Tractability and the computational mind. In M. Spreval, & M. Colombo (Eds.), *The Routledge handbook of the computational mind* (pp. 339-354). Oxford, United Kingdom: Routledge.

van Rooij, I. (2008). The tractable cognition thesis. *Cognitive Science, 32*, 939-984. https://doi.org/10.1080/03640210801897856

Verguts, T., & De Moor, W. (2005). Two-digit comparison decomposed, holistic or hybrid? *Experimental Psychology, 52*, 195-200. https://doi.org/10.1027/1618-3169.52.3.195

Verschaffel, L., Luwel, K., Torbeyns, J., & Van Dooren, W. (2009). Conceptualizing, investigating, and enhancing adaptive expertise in elementary mathematics education. *European Journal of Psychology of Education, 24*, 335-359. https://doi.org/10.1007/BF03174765

PsychOpen GOLD is a publishing service by
Leibniz Institute for Psychology Information (ZPID),
Trier, Germany. www.leibniz-psychology.org

PsychOpen GOLD