Mapping Integers to the Smallest Integer Possible

John Seibert
Advisor: William Calhoun
Bloomsburg University of Pennsylvania

Background and Motivation

- If we are given an arbitrary computer program and an input, **will a program always give an output?**
- Cannot say "yes" to above due to real-world limitations:
  - Limited size of memory (e.g., disk drive)
  - Replicability on all inputs
  - Program execution time
  - Programs that take itself as input
- **Project Goal:**
  - Construct a mathematical function in Python mapping integers to integers based on a set of algorithms

Original Cost Function

- **Attempt at computability**: let m be an input and C(m) its output
- Consider the original definition of the cost of an integer, m:
  - Given two operands, a and b, such that if a + b = m, or a * b = m, the goal is to **minimize C(a) + C(b)**
  - The current minimum starts with the input and follows the chart
  - Easily computable using smaller values of the function

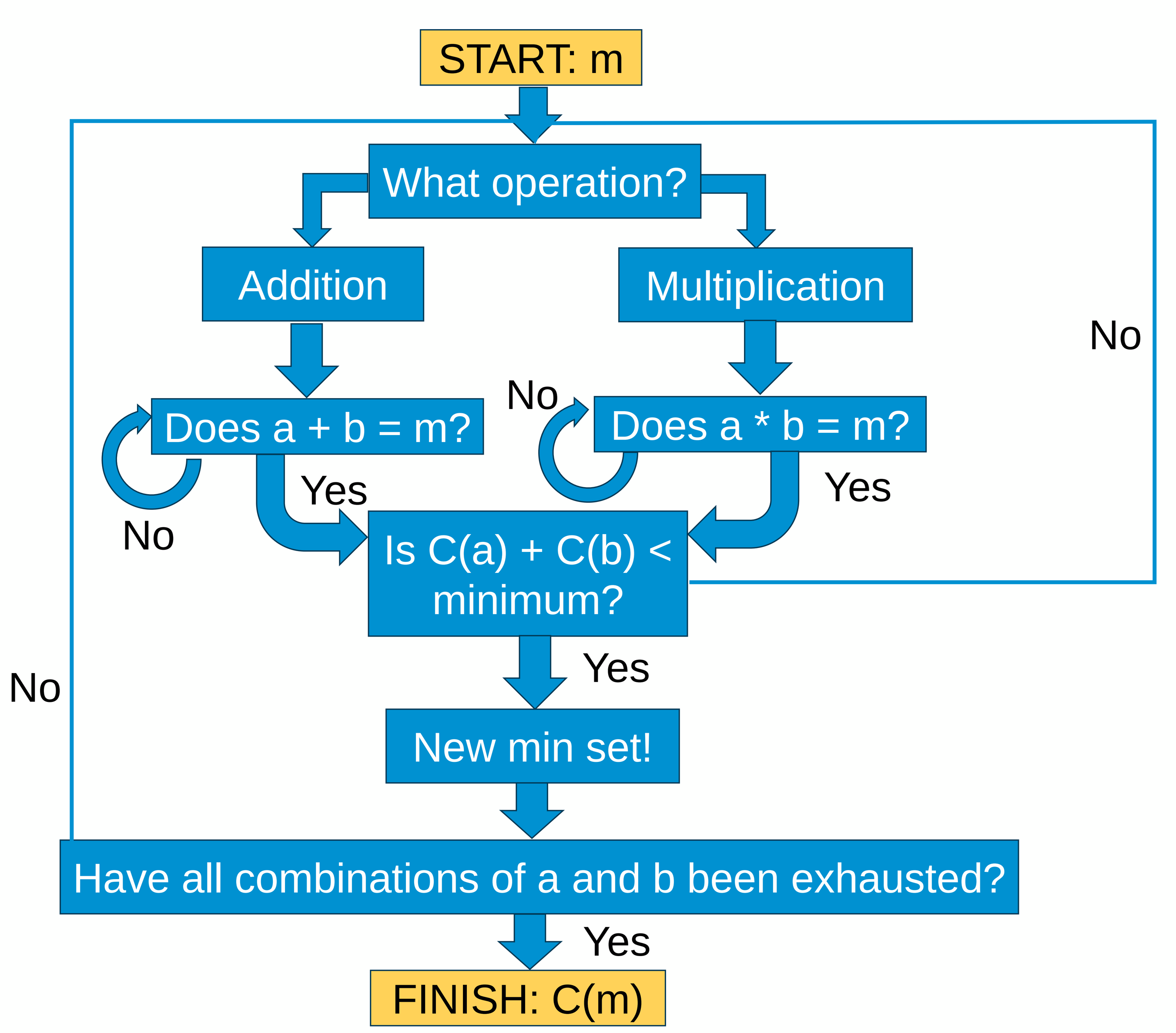

Figure 1. Flowchart of determining the cost of an integer using addition and multiplication.

Cost Function Implementation

Cost Function Framework

- Norfolk's work proposes loose principles for the cost function:
  - Only a finite number of functions can be considered
  - One operation can be evaluated at a time
  - The function gives a charge to the operation itself
  - Any number of operands can be in an operation
  - The minimum can decrease according to the driver algorithm
- **Any iteration (function-wide) has a fixed upper bound!**

Driver Algorithm

- The cost function contains several sub-algorithms
- If the cost, C(m), using just one algorithm is smaller than the minimum assigned, the cost becomes the minimum

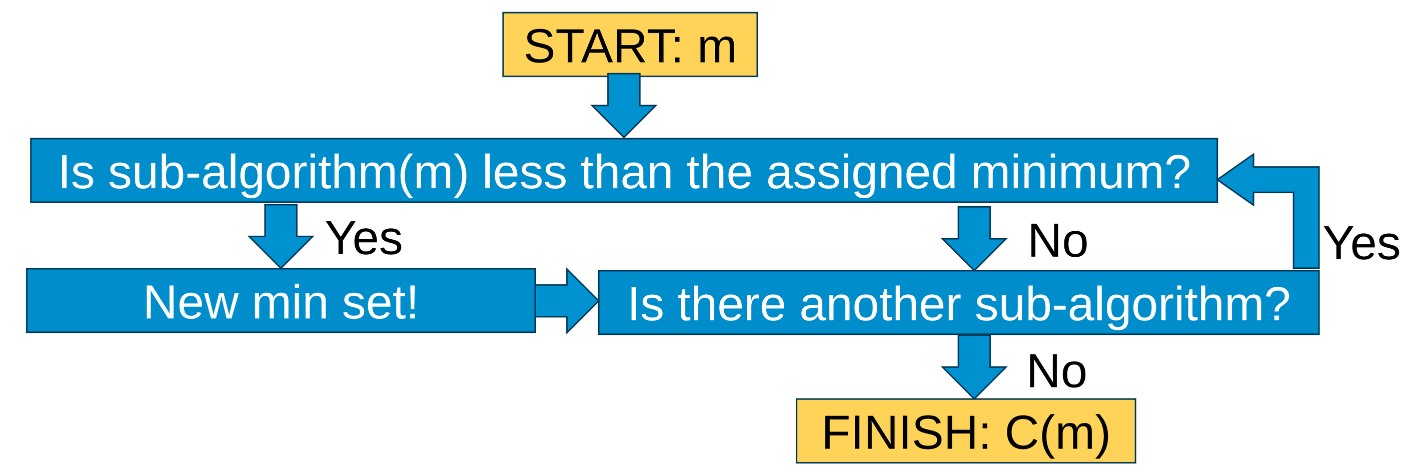

Figure 2. Flow chart explaining the driver algorithm for cost function.

- Sub-algorithms can be used in any order leading to the same result

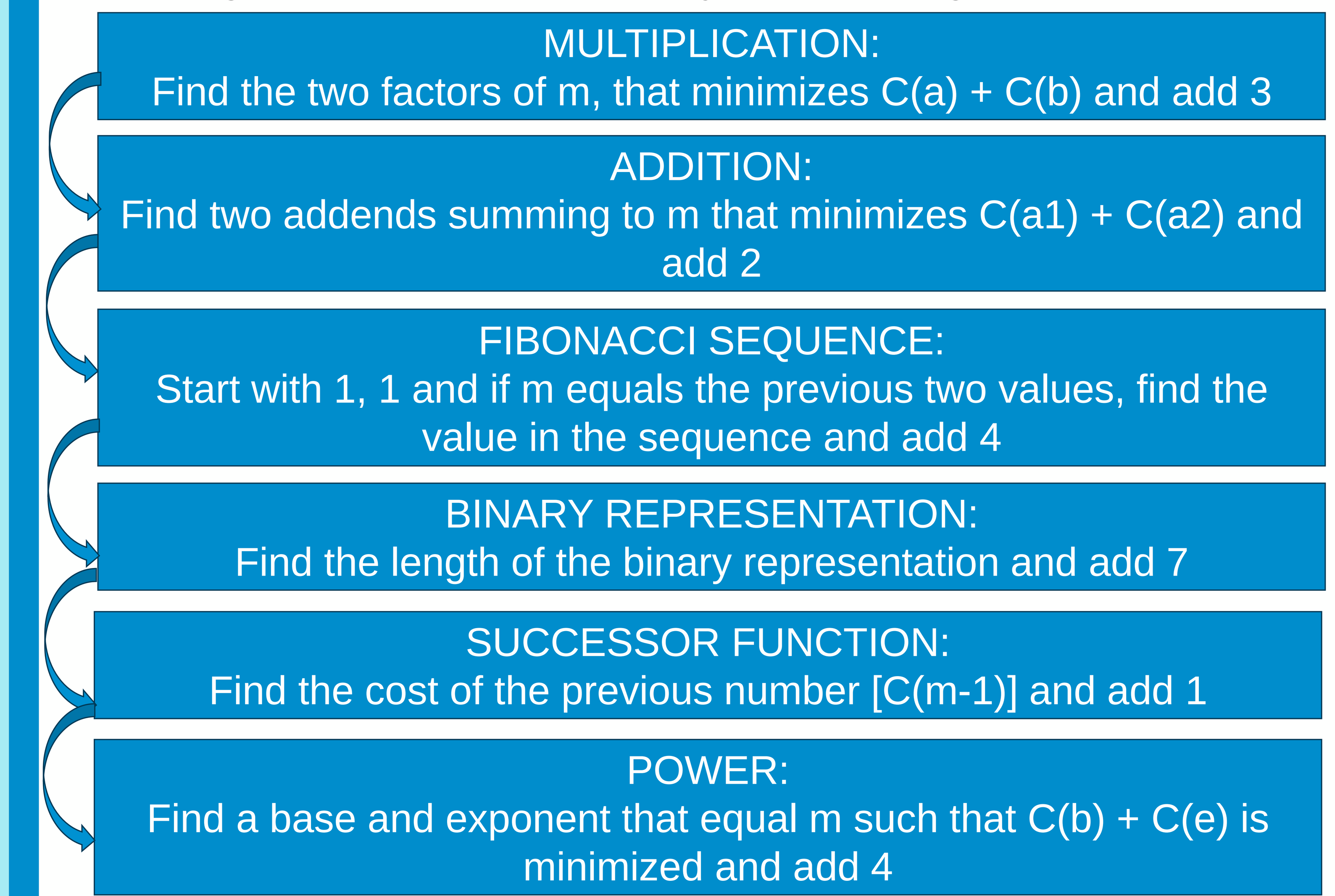

Figure 3. Diagram outlining the cost procedures for each sub-algorithm

Results

- We consider the lowest cost attained across all sub-algorithms
- Both models start increasing quickly and slow down for higher m
- Both show positive correlation with higher R^2 present in newer model

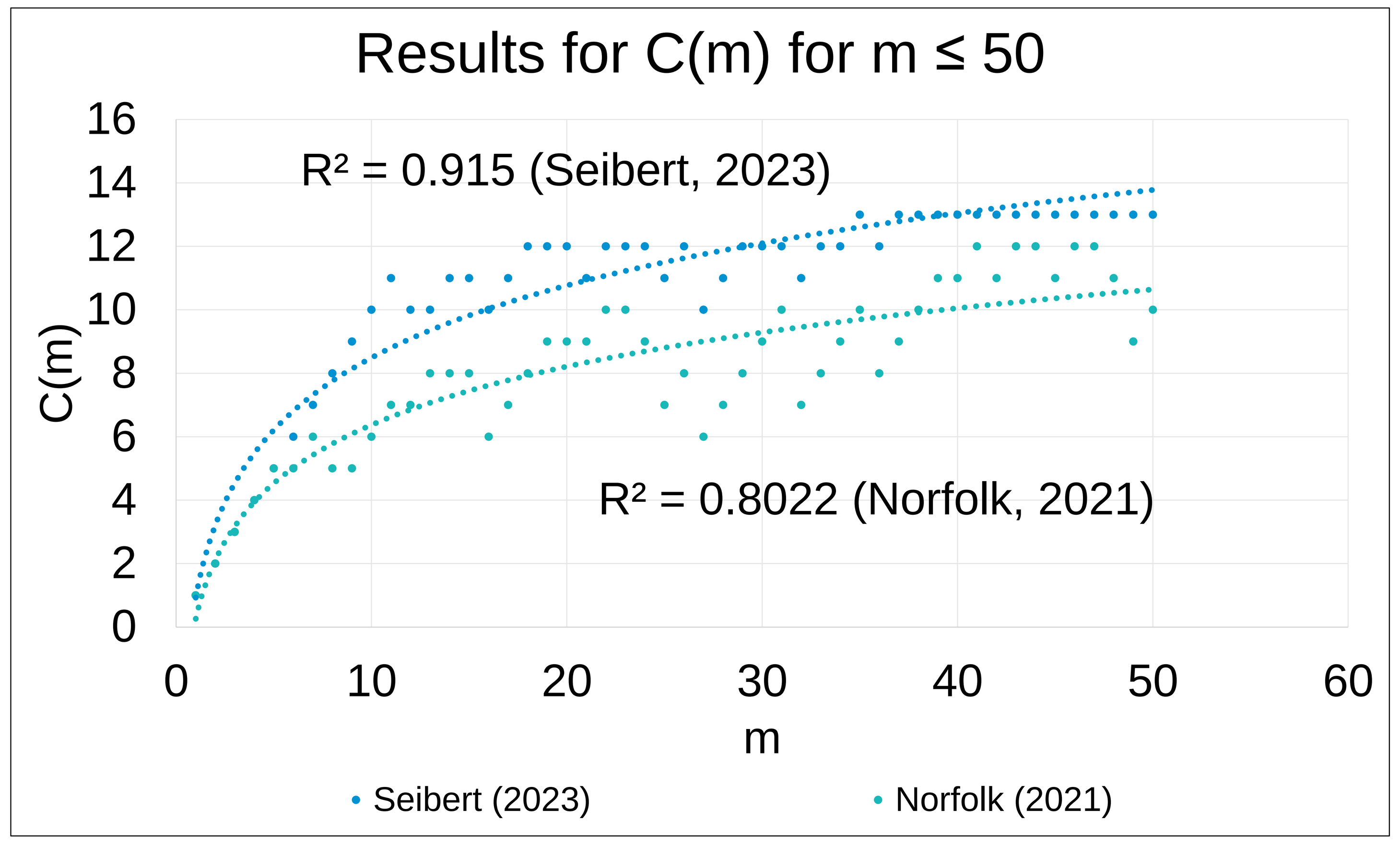

Figure 4. Associated graphs of C(m) vs. m for m < 50

Conclusions

- Outputs to C(m) mostly level out (except for powers of 2) in the newer model due to binary function being added
- Subtraction relies on cost values higher than numbers passed into the sub-algorithms, rendering it impossible for use
- More functions can be used when considering ones that use iteration on the output, but it requires changing the driver algorithm

Reference

- Norfolk, Maxwell (2021) "The Cost of a Positive Integer," Rose-Hulman Undergraduate Mathematics Journal: Vol. 22: Iss. 1, Article 9.

Source Code