Non-existence of the Algorithm that can Obtain the Optimal Solution for a Few Given Options of Investment in Constructive Mathematics

Jiahong Toby Sun

March 13, 2023

Abstract

Constructive mathematics is simply a mathematical process of computational procedures. Constructive Mathematical Analysis is the contrast of classical analysis. In this paper, I apply constructive mathematical analysis to prove that there could not be an algorithm that always chooses the optimal investment with the largest profit from a few given options in constructive mathematical economics. One of the main tools is the existence of partially defined non-extendable algorithms.

1 Introduction

Any real number can be described by a finite symbol in some ways, including language. The set of all sentence of finite length is countable. Therefore, it means that the real number can be enumerated by positive integers. However the Cantor diagonal construction shows that the set of all real numbers is not countable. Hence most real numbers can not be described in any mathematical sense and thus may be considered as not been relevant for the world we live in.

In the last few decades, with the development of computer science, people want to extract algorithm functions to represent proof of existence. Tradition Mathematics can no longer satisfy this purpose because of the issues of the standard decimal representation of the computation. For example, we supposed that x = 0.3333.... which is the decimal representation of $\frac{1}{3}$ [7]. When x is multiplied by 3. There is no way for the computer to decide the first digit is 0 or 1 because the computer cannot decide the last digit. Moreover, For same x, computer cannot precisely determine if $x < \frac{1}{3}$ or $x \geqslant \frac{1}{3}$, since the computer can never get the last digit of x. Due to these shortcomings, Constructive Mathematics can be an effective tool to solve these deadlocks. We can use constructive real number to describe this kind of situation which is generated by Cauchy sequence and a computer algorithm. The definition of some concepts in Constructive Mathematics will be illustrated later.

Constructive Mathematics is the background for my whole study. It is different from Classical Mathematics. People need to construct instead of interpreting "there exists strictly. [1]" The entire project stems from the notion of a computer algorithm. The program we use in this project is a partially defined algorithm that is non-extendable to all the natural number inputs [2]. The outputs of this program can be 0, 1, or never terminate [2].

Turing [3], as the originator of computers, generated an idea of whether he could set up a method to prove mathematics. He began to manufacture physical machines through human logical instructions and thinking activities. He also confirmed that automatic calculation could not solve all mathematical problems. This concept is called a Turing machine. Finally, he used many concepts of the Turing machine and made a computer that uses algorithms or programs to accomplish clearly defined tasks.

There are two famous schools of constructive mathematics that are following the ideas of Bishop and of Markov and Shanin. The Markov-Shanin school of Constructive Mathematics accepts the so called Markov principle, saying that if a set S of positive integers can be shown to be nonempty then one can algorithimically find some element of S. This of course can be done by a complete search through all the elements of the set S, but since there is no estimate on how long it will take a computer program to find this element, the followers of Bishop's approach do not allow the use of Markov's principle in their proofs and constructions.

In Bishop, Bridges, and Douglas's book Constructive Analysis [8], the basis of structural analysis is introduced, which explains the usefulness of many standard impossible programs [4][5]. However, we will follow a somewhat different approach to this subject which is developed by Markov, Shanin, and their followers. A good reference on this approach s the book written by B. A. Kushner, "Lectures on Constructive

Mathematical Analysis [6].

On the other hand many surprising fact that are clearly false in the traditional versions of the subjects are true in the constructive world. For example Ceitin (Tzeitin) showed that every function defined on real numbers is continuous, and in particular the functions similar to the Heaviside function do not exist in the world of Constructive Mathematics. In fact he showed more: that given ϵ from the definition of the continuous function one can always algorithmic find δ. However surprisingly not all functions on the constructive interval $[0, 1]$ are uniformly continuous even though in the classical Real Analysis all continuous functions on a closed unit interval are uniformly continuous.

Mathematics plays one of the central roles in theoretical economics and some of the classical problems studied are: optimal investment, optimal work assignment and the travelling salesman problem. In this program we studied how do the solutions and the existence of the solutions to these problems change when the cost involved are constructive real numbers. It goes without saying that in the cases where the costs are ordinary or even better rational numbers, that you can easily compare, the solutions to these problems are well known and understood.

2 Concept and Theory

Definition 2.1. **Alphabet** is a finite list of primitive symbols.

Definition 2.2. **Algorithm** is a finite sequence of symbols from Alphabet.

Figure 1 shows the block diagram of a normal algorithm. Figure 2 works in the following way: if P_i occurs in P, the result will be the output 2 in the case of substitution; if p_i occurs in P, the result will be the output 3 in the case of termination; if P_i does not occur in word P that enters as the input, then P moves into the $(i + 1)^th$ block.

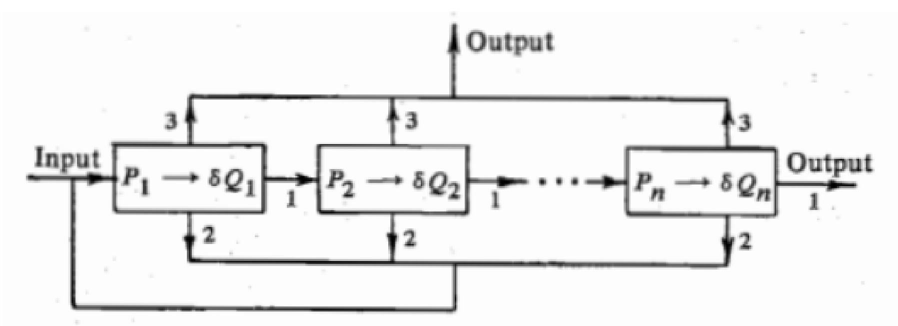

Figure 1: Block Diagram of a Normal Algorithm

Definition 2.3. Let α be a normal algorithm in the alphabet A_1, and α' be a normal algorithm in the alphabet A_2. If α' with exactly same scheme as α, the algorithm α' will be called the **extension** of α.

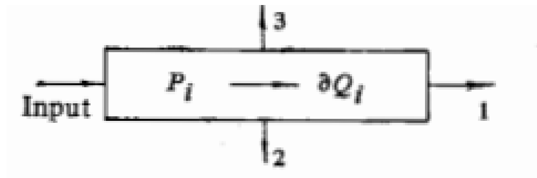

Figure 2: Each Step of a Normal Algorithm

Definition 2.4. Constructive Sequence of Natural Number(SNN)is an algorithm transforming every natural number into a natural number.

Definition 2.5. Constructive Sequence of Rational Number(SRN) is an algorithm transforming every natural number into a rational number.

Definition 2.6. A Constructive Real Number(CRN) is a pair of programs α and program β. An integer n is taken as the input. The program gives the Cauchy convergent sequence $\alpha(n)$. Given integer n, $\beta(n)$ is the constructive sequence of natural number such that for all i, $j \geq \beta(n)$, we have $|\alpha(i)-\alpha(j)| \leq 2^{-n}$. In this situation, We defined that program α is fundamental, and program β is the regulator of the fundamentality of program α.

Definition 2.7. A function is **computable** if there exists an algorithm that can do the same thing as the function.

Definition 2.8. For $n, x \in \mathbb{N}$, we defined that $U(n, x)$ is a **universal function** for a class of computable function of one variable: if for every n, the function $U_n(x)$ is a computable function of one variable, and all computable functions of one variable are one of the U_n. $U_n(x)$ is defined as $U(n, x)$.

Lemma 2.1. *There exists a computable function U of two variables that is universal for the class of computable function of one variable.*

Proof. Let us write all programs' computing functions of one variable into a computable sequence $p_0, p_1, p_2, ...$ based on the increasing length of the program. We defined $U(i, x)$ as the result of the work of program p_i on input x. This is the desired universal function . The section U_i is a computable function via the program P_i. □

Lemma 2.2. *There does not exist everywhere defined function of two variables that is universal for the class of computable everywhere defined function of one variable.*

Proof. Let U be such function of two variables. We define that $U(n) = U(n, n)$ and U(n) $= U_n(n)$. Moreover, We define that $d(n) = u(n) + 1$ is different from U_n. Thus, the computable function everywhere defined function $d(n)$ is different from every section U_n. Hence U is not a universal function, which leads to the contradiction. □

Lemma 2.3. *There exists a computable function d' that does not admit on everywhere defined computable extension.*

Proof. We defined that $d'(n) = d(n) + 1$, where d is the function from the previous proof. Every extension of d' that is everywhere defined is different from d. Therefore, it does exist a computable function that doesn't admit on everywhere defined computable extension. □

3 Proof

3.1 Assumption

The entire project stems from the notion of a computer algorithm, The program we use in this project is a partially defined algorithm that is non-extendable to all the natural numbers input. The outputs of this program can be 0, 1, or never terminate. Lemma 2.1, Lemma 2.2, and Lemma 2.3 can prove that this algorithm is valid.

Let's suppose we can invest in two different companies that are called X and Y. The profit that I can get in company X is the constructive real number a_n. The profit that I can get in company Y is the constructive real number b_n. Let H be the non-extendable partially-defined program that transforms positive integers to 0 and 1. Definition 2.6 can define constructive real numbers a_n and b_n that would lead to the desired example.

Definition 3.1. $a_{(n,k)} = 1$, if the computer program H finished working on input n by the k^{th} step or if the program H has already finished working on n by the k^{th} step produced 1.

$a_{(n,k)} = 1 + 2^{-m}$, if the computer program H has finished on input n by the k^{th} step and produced 0. Variable m is the step number when it finished working on n.

Definition 3.2. $b_{(n,k)} = 1$, if the computer program H has not finished working on input n by the k^{th} step or if the program H has already finished working on n by the k^{th} step produced 0.

$b_{(n,k)} = 1 + 2^{-m}$, if the computer program H has finished on input n by the k^{th} step and produced 1. Variable m was the step number when it finished working on n.

3.2 Proof of Two Investment Options

Theorem 3.1. *There does not exists an algorithm can obtain the optimal way for investment with larger profits from two given investment options.*

Proof. Based on the definition of constructive real number that I provided before, we can easily find the regulator of a_n and b_n. So, they are two constructive real numbers.

According to the definition of a_n and b_n, we can compare profit based on the output of program H.

When the output of computer program H is 1, the profit of X company a_n equals one, the yield of Y company b_n equals $1 + 2^{-m}$. Since 2^{-m} is always a positive number no matter what the value of m is. So, the profit I can get from the Y company is higher than the profit that I can obtain from the X company. On the contrary, when the output of computer program H is 0, the profit of X company a_n equals $1 + 2^{-m}$, and the profit of Y company b_n equals 1. As a result, the X company's profit is higher than the Y company's profit.

Note that, since the algorithm H is partially defined and non-extendable, it cannot give an output to all natural inputs. Thus, program H cannot always tell us the optimal way for investment.

Now, we use the contradiction to prove this theorem by introducing a new hypothetical program P. We suppose that there exists an hypothetical program P that can always determine the optimal way of investment. Now, we can prove that hypothetical program P will lead to an extension of program H. We suppose that program H will never terminate when the input is x. According to our definition, algorithm P can obtain the optimal way of investment when the input is x. Therefore, it can always compare the value of a_n and b_n. However, in this case, there will be an extension of algorithm H at x, which we called H'. Algorithm H' is defined as follows:

Definition 3.3. $H' = 1$, when program P can gives that $b_n > a_n$, company Y is profitable.

$H' = 0$, when program P can gives that $a_n > b_n$, company x is profitable.

Therefore, there is an extension of algorithm H at x, which can be all natural numbers that is initially not defined in algorithm H. Thus, H can be extended to all the natural number inputs. This extension lead to the contradiction with the non-extendable program H.

As a result, there does not exist such hypothetical program P that can always give the optimal way for investment with the larger profits from two options. □

3.3 A Generalization of Proof

As mentioned previously, we prove that there is no algorithm that can always choose the optimal solution from two investment options. Now we generalize two investment options to several investment options.

Theorem 3.2. *There is no algorithm that can obtain the optimal way for investment from several investment options with Constructive Real Number profits.*

Proof. Now, let us consider the condition that we have n investment options. Thus, there are n constructive real numbers. When we compare two of the constructive real numbers, as we have already proved, there is no such algorithm that can choose the optimal investment option from these two choices. Otherwise, it will be extended to all-natural number inputs, which will lead to a contradiction. Therefore, we cannot compare each two investment options because of Theorem 3.1.

Thus, we cannot find an algorithm to choose the optimal solution for a few given options of investment. □

4 Discussion

Overall, we conclude that there does not exist an algorithm to always obtain optimal solutions for a few given options in Constructive Mathematics. We have shown the non-existence of optimal solutions for a few given options of constructive real number profits. Still, our proof doesn't indicate whether we can acquire an optimal solution for the possibility of real number profits and constructive real number profits. Furthermore, since the application of constructive real numbers is abundant, I will extend my research to discuss more problems such as employees' salaries and stock markets. The introduction of Constructive Mathematics in economic problem can have promising future.

5 Acknowledgment

This project was created by Viktor Chernov and Vladimir Chernov. I wish to show my deep appreciation to Dr. Chernov. I would also like to extend my thanks to the teacher's assistant Daniel Xiang for academic support. Studies were completed through the Ivy Mind Academy Program in the summer of 2021.

References

[1] Constructive Mathematics (Stanford Encyclopedia of Philosophy). https://plato.stanford.edu/entries/mathematics-constructive/.

[2] Shen, Alexander, and Nikolai Konstantinovich Vereshchagin. Computable Functions. American Mathematical Society, 2003.

[3] Turing: Computer Pioneer, Code-Breaker, Gay Icon — Live Science. https://www.livescience.com/29483-alan-turing.html.

[4] Bishop, Errett, and Douglas S. Bridges. Constructive Analysis. Springer, 1985.

[5] Constructive Analysis - Encyclopedia of Mathematics. https://encyclopediaofmath.org/wiki/Constr uctive_analysis.

[6] Kushner, B. A. Lectures on Constructive Mathematical Analysis. American Mathematical Society, 1985.

[7] Geuvers, Herman, Milad Niqui, Bas Spitters, and Freek Wiedijk. n.d. "Constructive Analysis, Types and Exact Real Numbers."

[8] Bishop, E., amp; Bridges, D. S. (1985). Constructive analysis. Springer-Verlag.