https://doi.org/10.29013/EJEMS-22-2-16-18

Keming Miao,

PROOF THAT THERE IS NO ALGORITHM TO THE OPTIMAL WORK ASSIGNMENTIN CONSTRUCTIVE MATHEMATICS

Abstract. This paper proves that in constructive mathematical economics, there could not exist an algorithm that always does the optimal assignment of workers to the working positions. The main method is based on the fact that there does exists a computable function that does not admit an everywhere defined computable extension.

Keywords: Constructive Mathematics, Economics, Optimal Work Assignment.

Introduction

Constructive mathematics is the field which studies describable constructive numbers and constructive topological or metric spaces. It is characterized by proofs based on explicit, algorithmic solutions, and proofs by contradiction are considered to be not constructive. The idea of constructive mathematics started from A. A. Markov, who proposed the Markov's Principle, that is, for a decidable *P*, $\forall n\ (P(n) \vee \neg P(n))$, *then* $\neg\neg\exists n\ P(n)$ *implies* $\exists n\ P(n)$.

A computable number can be determined by a finite computer program. It was defined by A. M. Turing, that computable numbers are the real numbers whose binary expansions can be enumerated by a finite procedure. He similarly defined computable functions. However, his definitions turned out to be incorrect, as under his definition, the addition and multiplication of computable numbers are no longer computable.

The Optimal Work Assignment problem is significant in the field of economy. Solving the problem would help corporations to lower the cost and save money. In this paper, we will prove that there is no algorithm that can solve the general Optimal Work Assignment problem.

Algorithm to the optimal work assignment

The paper mainly proves the following theorem:

Theorem 1.1.

In constructive mathematical economics, there does not exist an algorithm that always does the optimal assignment of workers to the working positions.

To start the proof, we consider an algorithm H that is partially defined that is not extendible to all natural inputs. Note that such algorithm is valid since non-extendible programs exist (1).

First, we define two sequences $A_{n,k}$ and $B_{n,k}$, generated by algorithm H:

Definition 1.1.

$$A_{n,k} = \begin{cases} 1, & \text{if } H \text{ by step } k \text{ did not terminate yet or it terminated alrady and gave 1.} \\ 1+2^{-m}, & \text{if } H \text{ by step } k \text{ terminate and gave 0, m is the number of steps when this happens.} \end{cases}$$

Definition 1.2.

$$B_{n,k} = \begin{cases} 1, & \text{if } H \text{ by step } k \text{ did not terminate yet or it terminated alrady and gave 0.} \\ 1+2^{-m}, & \text{if } H \text{ by step } k \text{ terminate and gave 1, m is the number of steps when this happens.} \end{cases}$$

Definition 1.3.

Constructive Real Numbers: A constructive number is a pair of programs $\alpha(i)$ *and* $\beta(i)$, *where* $\alpha(i)$ *is a Cauchy sequence and* $\beta(i)$ *is the convergence regulator, such that for every i, j greater or equal to* $\beta(N)$, *we have* $I\alpha(i)-\alpha(j) < 2^{-N}$. [2; 3].

We first observe the following fact that sequences $A_{n,k}$ and $B_{n,k}$ both lead to constructive real numbers when we fix n and vary k, as it is obvious that such convergence regulator $\beta(i)$ do exist for the two sequences.

An example of the cost matrix

Now, consider the specific case when the cost matrix is:

	Job 1	*Job* 2
Worker 1	A_n	1
Worker 2	B_n	1

[1]

In this particular case, Worker 1 and Worker 2 receives a payment of A_n and B_n for doing Job 1, respectively. Both Worker 1 and Worker 2 get 1 for Job 2.

In order to determine the optimal work assignment in this situation, we need to choose the case when the total payment is the least. Therefore, a comparison between the value of A_n and B_n needs to be made.

Specifically, algorithm H will lead to the choice of the combination Worker 1→Job 1& Worker 2→Job 2 if the program prints 1, because in this case, A_n, which is equal to 1, is smaller than B_n, which is equal to $1 + 2^{-m}$.

Similarly, it will lead to the choice of the combination Worker 1→Job 2 & Worker 2→Job 1 if the program prints 0, because in this case B_n, which is equal to 1, is smaller than A_n, which is equal to $1 + 2^{-m}$.

However, note that algorithm is inextendible, and therefore cannot give an answer to all natural inputs.

Now, we prove the problem by contradiction by introducing a new algorithm P.

Definition 1.4.

Algorithm P: A hypothetical algorithm that will always give an answer to the optimal work assignment problem for every cost matrix.

We state that if there is such algorithm P that can always solve the problem, there would be an extension of H to all natural numbers.

Theorem 1.2. *The existence of Algorithm P will lead to an extension of program H. Proof.* Suppose that algorithm H will never terminate at x.

According to the definition, algorithm P would be able to compare A_n and B_n, giving an answer to the Optimal Work Assignment problem when the input is s.

However, if this was true, the extension H' of algorithm H could be defined at x as follows:

$$H' = \begin{cases} 1, & \text{when } P \text{ gives the combination of Worker } 1 \to \text{Job } 1, \text{ Worker } 2 \to \text{Job } 2 \\ 0, & \text{when } P \text{ gives the combination of Worker } 1 \to \text{Job } 2, \text{ Worker } 2 \to \text{Job } 1 \end{cases}$$

This is an extension of algorithm H at x, which can be all natural numbers that is initially not defined in algorithm H, so H can be extended to all inputs.

However, if there exist such algorithm P that can solve the Optimal Work Assignment problem, the conclusion (Theorem 1.2.) will contradict with the fact that algorithm H is not extendible, which we mentioned before.

Therefore, there could not exist an algorithm P, such that it can solve the Optimal Work Assignment problem.

A generalization of the cost matrix

As mentioned previously, we constructed cost matrix when there are only two workers and working positions. Based on the construction we already had, we can also give constructions and proofs of the cost matrix when there are arbitrary numbers of workers and working positions.

We start with a simple case. First, we analyze the situation when three workers need to be assigned to three working positions.

Based on matrix [1], which there are only two workers and two jobs, consider the specific case when the cost matrix is:

	Job 1	*Job* 2	*Job* 3
Woker 1	A_n	1	1
Woker 2	B_n	1	1
Woker 3	A_n	1	1

This time, A_n, B_n, and A_n are the payment Worker 1, Worker 2, and Worker 3 get respectively for Job 1. All three workers get 1 for Job 2 as well as Job 3.

Still, to determine the optimal work assignment in this situation, we need to choose the option when the total payment is the least, so a comparison between the value of A_n and B_n needs to be made.

However, as we already proved, there is no such algorithm that can do this, or otherwise, the program can be extended to all natural inputs.

Here, we finished constructing an example of the cost matrix for three workers and working positions.

The construction is very similar when there are k workers and jobs. Consider the following cost matrix:

	Job 1	*Job* 2	*Job* 3	...	*Job k*
Woker 1	A_n	1	1	...	1
Woker 2	B_n	1	1	...	1
Woker 3	A_n	1	1	...	1
Woker 4	A_n	1	1	...	1
...	...	...	...	$\ddots$	...
Woker k	A_n	1	1	...	1

All the workers get A_n for Job 1 except for Worker 2, who gets B_n for Job 1. All the k workers get 1 for Job 2, Job 3, …, Job k.

The reason why there is no algorithm that can do the Optimal Work Assignment is the same, that is, a comparison between A_n and B_n cannot be made. If there was such algorithm, an extension of this algorithm to all natural numbers can be found, which leads to a contradiction.

Conclusion

We conclude that in constructive mathematical economics, there does not exist an algorithm that always does the optimal assignment of workers to the working positions.

References:

1. Vereshchagin N. K., Shen A. Computable Functions. Amer Mathematical Society. (2002).
2. Kushner B. A. Lectures on Constructive Mathematical Analysis. Amer Mathematical Society. (1984).
3. Bishop E., Beeson M. Foundations of Constructive Analysis. Ishi Press. (2012).

This research was conducted during the Ivy Mind Summer Program, and the project was created as a result of joint collaboration of Viktor Chernov and Vladimir Chernov