# On the equivalence between P=NP and the Extended Church-Turing thesis.

*Dated: May-4-2018.*


## Abstract

Since the mere Church-Turing thesis says that Turing machines can simulate any computational device, but perhaps at the cost of an exponential blowup in time and/or space, the Extended Church-Turing Thesis (ECT for short) states that every physically realizable computational device can be efficiently simulated by Turing machines. Unlike the Church-Turing Thesis, which computer science conventional wisdom appears to think is true, the Extended Church-Turing Thesis is widely believed to be false just as it happens with the P=NP conjecture. In this paper we present an interesting extension from our seminal papers A new approach: A hardware device model solving TSP in $O(n^2)$ time and Efficiently solving Traveling Salesman Problem in a computer network. (Internet). proving the equivalence between ECT and P=NP. In section 1 we go directly to a physically feasible model and prove mathematically its validity. In section 2 we explain the basic ideas behind the physical model. In section 3 we prove the equivalence between the conjecture P=NP and the ECT thesis.


### 1. The Model.

Let be S a set of n emitter and receiver devices (e.g. computer network, cellular phones, etc., from now on *cells*) at any configuration in a 3-dimentional space. All the cells are directly linked to each other whiteout antennas or intermediate devices of any kind and every cell can perform the usual operations, receiving, processing and sending a string of bytes, at the same speed. Now, any cell, let´s say cell 1, sends in $t=t_0$ its own number (1) to the remaining cells of the set. Every time a cell receives a string, appends its own number to the string and rebroadcasts it **unless its own number is already in the string**, thus avoiding redundant loops and subtours.

*Proposition:* The first string[1] of length n to reach cell 1 in $t = t_1$ is the shortest path from 1 to itself trough all points of S-{1} and $c(t_1\text{-}t_0)$ is the optimal tour length ($c$ = speed of light), as we can state with no loss of generality that operation inside each cell takes no time.

*Lemma 1*: Every string of length n arriving to 1 is: 1,σ(S-{1}) i.e., element {1} at first position followed by a permutation of S-{1}.

*Proof*: Every string of length n reaching cell 1 started in 1 and has passed only once through each one of the n-1 remaining members of the set. Had it passed twice the string would not have been sent so the string would not have reached cell 1. Had it skipped any and the string would not have length n.

*Lemma 2*: Every possible permutation 1,σ(S-{1}) will reach 1 in finite time.

*Proof*: Every cell appends only once its number to every string and every string contains only once every cell, so given that every string begins with 1 and n is finite every 1,σ(S-{1}) string will reach 1 after n steps.

1.- A straightforward reasoning assures that in fact not one but two n-strings will arrive in first position simultaneously with the same stored sequence in reverse order.

Lemma 1 and 2 prove that every tour reaches cell 1 and every n-string reaching cell1 is a tour. Now suppose the first n-string to arrive to 1 in time t and tour distance d is not the optimal tour i.e. the tour of minimum distance. Hence the optimal tour with tour distance d´ will reach 1 in t´ > t but that is impossible given that: t´= d´/c, d´< d and c constant.
q.e.d.

**2. The idea behind.**

¿How long it takes to solve the traveling salesman problem? It takes as long as it takes to the salesman himself to travel the optimal path. However, (n-1)! salesmen are required. Let´s see it: Each salesman takes a table, a tour sequence with n boxes to be filled out with the number of the cities he passed trough (Not necessary *he*, but the table itself, or more accurately, the information stored at the table). So in the beginning n-1 salesmen leave at the same time the first city (say city one) bound for the remaining cities. In their tables the first box is filled with number one and all other boxes remain empty. Whenever a salesman reaches a city the information stored at his table is copied to the n-1 tables of n-1 new salesmen (Obviously the arriving salesman could continue his travel thus reducing number of new salesmen needed to n-2 but that is completely irrelevant to our purpose). These n-1 salesmen append the number of their city to the table, leave their city bound for n-1 cities and so on and on. There is one important exception: No salesman leaves a city when its number is already stored at the arriving table.

Is almost trivial to prove that:

1) The first salesman to reach the starting city with a full n table has traveled through the optimal path, optimal path that has been stored at his table.

2) The optimal tour length is: time from the starting city multiplied by speed.

Salesman himself has solved salesman problem and the certificate is in salesman´s hand: The table of the n cities in the very order his table passed trough.

Obviously some assumptions must be made to assure the accuracy of the result:

1) All salesmen move at the same constant speed.

2) Time taken in copying and, when required, appending data to the tables is zero. Or at least is the same for every salesman at any city. Anyway this time must be negligible in the sense that is less than the time taken to travel between the two closest cities in the set, not a challenging assumption indeed.

Now, the well known drawback, the very bottleneck of this approach is: when n grows the number of salesmen required grows exponentially until it reaches mammoth amounts of them even for humble sets of cities. Second drawback: The longer the path the longer the waiting time[2].

So all we need to improve our approach to TSP is to get very fast salesmen in industrial quantities. Do we have? Do we have an almost infinite supply of salesmen traveling at speed of light? We do. Photons. This brings us back to section 1.

2.- The problem of the tour length is almost negligible (*almost* is not absolutely, see below) given that the set can always be resized with a change of scale shrinking distances while keeping relative positions.

## 3. Equivalence between the Extended Church-Turing Thesis and P=NP.

1) Extended Church-Turing thesis implies P=NP. As shown, exists a physically realizable computation model solving TSP in $O(n^2)$ time. If ECT thesis holds, this computation model can be simulated by a TM with polynomial overhead, i.e., in $O(n^2)^k$ time, so it follows that P=NP.

2) P=NP implies Extended Church-Turing thesis. Indeed if P=NP, anything computable is solvable in polynomial time. If the mere Church-Turing thesis (widely tested and believed to be true) holds, every physically realizable model can be simulated by a Turing machine. If it is computable in $O(n^k)$ time, it follows that the ECT is true.

**$PA^3$**

*Dated: May-4-2018.*
*Óscar E. Chamizo Sánchez.*
*chasanos@telefonica.net*
*Any comment, suggestion or critic is welcome.*