THE QUARTERLY JOURNAL OF AUSTRIAN ECONOMICS

VOLUME 26 | NO. 3 | XX–XX | FALL 2023 WWW.QJAE.ORG

REVISITING THE COMPUTATION PROBLEM

PAUL F. CWIK AND LUCAS M. ENGELHARDT*

JEL CLASSIFICATION: B51, B53, C63, P20

ABSTRACT: In 2013, Engelhardt (2013) calculated that the combined power of the top five hundred supercomputers would take approximately 10.5 quintillion years to compute the distribution of eighty thousand heterogeneous goods among six billion consumers, posing a serious practical challenge to the implementation of computerized central planning. Allin Cottrell (2021) calls into question Engelhardt's assertion, noting that the algorithm used by Engelhardt not only scales poorly but is not even valid for the problem Engelhardt posed. Cottrell offers an iterative algorithm that a one-petaflop machine could use to solve the distribution problem in about five minutes. We correct errors in both Engelhardt and Cottrell, and we offer a way to incorporate production into the problem. The result: modern supercomputers are still not powerful enough to solve central planning's computation problem.

One of the wonders of the modern market system is its ability to gather and process large amounts of data that allow for better decision-making. Retailers use technology to create targeted advertisements, which allow them to reduce their marketing costs and match consumers with the products they most likely want to purchase. These improvements have been made possible by significant advances in information technology combined with entrepreneurial insight.

* Paul F. Cwik (pcwik@umo.edu) is the BB&T Professor of Economics and Finance at the University of Mount Olive.

Lucas M. Engelhardt (lengelha@kent.edu) is an associate professor of economics at Kent State University's Stark Campus.

Quart J Austrian Econ (2023) 26.3:XX–XX
https://qjae.scholasticahq.com/
doi:

But improvements in computer technology also embolden "technosocialists," who advocate replacing markets with computers to better meet human needs and desires.[1] But is such computerized central planning actually possible?

In his 2013 study, Engelhardt (2013) looks at technosocialism from a new perspective: is it possible for an advanced computer system to solve, in a reasonable amount of time, all the equations necessary to centrally plan an economy? He estimates the time it would take for the TOP500 supercomputers[2] to solve the system of equations required to maximize social utility for the world population, assuming heterogeneous preferences for eighty thousand heterogeneous goods. Engelhardt's arithmetic shows that to calculate a moment's equilibrium would take several times longer than the estimated current age of the universe. This conclusion suggests that, in addition to other problems, socialism faces a "computation problem"—an inability to process the required information in a useful amount of time. A recent study by Cottrell (2021) calls into question those results, suggesting a more time-efficient algorithm and pointing out that supercomputers have increased in speed since Engelhardt's study—the top supercomputer today is about four times faster than the combined TOP500 in 2013. While noting that maximizing social utility is an unfair standard that capitalism itself does not satisfy, Cottrell contends that a more efficient algorithm and faster computation means that the problem proposed by Engelhardt could be solved in just a few minutes. This paper answers Cottrell and updates Engelhardt to consider whether computing power is a barrier to the implementation of technosocialism.

Since the computation problem is technological, its size varies as the world changes. As populations grow and new products are introduced, the problem gets larger; however, as computer processing technology and algorithms improve, the problem gets smaller. In this paper, we show that (1) Engelhardt significantly understates the size of the problem and (2) Cottrell correctly criticizes Engelhardt's algorithm as inefficient, but that (3) Cottrell's algorithm still does not solve the problem posed by Engelhardt

[1] See, for example, the classic work of Cockshott and Cottrell (1993), as well as more recent work by Dapprich (2023), Nieto and Mateo (2020), and Nieto (2022).

[2] TOP500 is the name of the project that tracks the top five hundred supercomputers.

and drastically underestimates the time required. In the end, a more efficient algorithm does reduce the time required to solve the problem posed, but two barriers remain. The computation problem continues to be an issue in the world today because (1) the size of the problem is drastically larger than Engelhardt assumed and (2) electricity requirements limit computation. Implementing technosocialism not only requires faster computers but also enough electrical energy to perform the computation. The computation problem remains a significant barrier to the implementation of large-scale central planning.

PROBLEMS WITH CENTRAL PLANNING

The idea of centrally planning a community shows up in ancient Greek thought, debated by both Plato and Aristotle, but, throughout human history, its implementation has failed on any scale larger than a family or small clan. The main problems with central planning can be summarized in four words: incentives, information, calculation, and computation. Before exploring the arguments against central planning, we want to establish the best conditions for its possibility by assuming that the hypothetical planners are angelic—working in the best interest of society rather than their own.

With this assumption in place, there are four arguments that show why centrally planning a complex economy cannot succeed. The first and oldest is the incentive argument, which simply asks, "What is the incentive to work hard, if everything that one works for is taken away?" In every economy, there are unpleasant and dangerous jobs that must be performed for a society to function. Who will clean the toilets, haul away the trash, go into the sewers, go into the mines, or climb the high steel? Who gets the jobs that are inside the climate-controlled offices? Market economies use positive incentives—primarily, money—to attract workers to unpleasant jobs. Centrally planned economies, on the other hand, have tried to use negative incentives, such as the gulags and secret police, to motivate people to do those jobs. Karl Marx was familiar with this argument and had a solution. He argued that under communism, human nature could be changed through education. A new "soviet" man would emerge and selflessly do all that was necessary for society to thrive. While fundamentally changing human nature has not been successful, we will assume for the purpose of this paper that all humans are already completely selfless and perform assigned tasks and duties to the best of their abilities.

The second argument for the impossibility of central planning is the knowledge argument, most notably advanced by Hayek's 1945 paper "The Use of Knowledge in Society." Hayek argues that the information that prices convey throughout the economy is not reducible to data that can be communicated to a central planning board. Beyond "the knowledge of the particular circumstances of time and place," prices communicate the subjective tastes and preferences of consumers as they buy and abstain from buying in the marketplace. Without markets, these subjective preferences remain unrevealed to central planners. Lange (1936) argues that adjusting for shortages and surpluses after production could compensate for the lack of information about preference, but, even if feasible, this approach does not advance technosocialism, which involves the computation of distribution prior to production and therefore needs information about tastes and preferences to plug into the planning algorithm. Nevertheless, we will put aside the knowledge argument by assuming that our subjective tastes and preferences (our demand schedules) can be gathered and fed into a computer algorithm. Advances in wearable technology—which can read brain patterns, create fuzzy images of thoughts, and even extract a person's PIN—may yet realize the possibility.[3]

The calculation argument offers a third demonstration that central planning is impossible. It was first developed by Ludwig von Mises in 1920 and expanded in his book *Socialism: An Economic and Sociological Analysis* (1951). Contrary to many schools of economics, Austrian economists demonstrate that supply curves are derived from the law of diminishing marginal utility and not from "objective factors" (such as the monetary cost of production). In a centrally planned economy, the planners exercise control over all the factors of production, either de facto or de jure. Since the central planners, in the absence of any private owners, dictate how resources are to be used, no true measure of opportunity cost can emerge. Supply curves are

[3] Boettke and Candela (2023) provide a response to the technosocialists' arguments, emphasizing the contextual, fleeting nature of knowledge. For this paper, though, we set aside this issue.

based upon the subjective valuations of the owners and cannot exist without them. Without supply curves, there can be no markets or market prices, and without access to market prices at various stages of production, the efficient use of resources cannot be calculated. Lambert and Fegley (2023) apply Mises's argument to the world of big data and artificial intelligence, showing the continued relevance of the calculation problem in this context. In this paper, however, we assume away this argument by collapsing all production processes into a single-step production function. As we will see below, only the volume of transactions to be calculated, and not the structure of production, is relevant to this paper's argument.

The fourth argument is the computation argument, most recently presented by Engelhardt. The computation problem goes back to at least 1908, when Enrico Barone (2012) characterized the system of equations necessary to describe an economy in equilibrium. Barone, however, does not argue for the feasibility of central planning: his system of equations only describes what Misesians call an evenly rotating economy (or, in mainstream parlance, a static general equilibrium), in which nothing ever changes. The next section explores the history of this problem in more detail.

HISTORY OF THE COMPUTATION PROBLEM

In his paper "The Ministry of Production in the Collectivist State," Enrico Barone describes the number of equations necessary to solve for equilibrium in an economy. Barone intends his description to lay out exactly how difficult such a problem is to solve. His paper is filled with two types of comments that call into question the feasibility of widespread central planning. First, he emphasizes that, while the form of the equations can be known a priori, the parameters in each equation call for information that the planner would have to acquire. Anticipating Hayek's information problem, he is doubtful about the possibility of acquiring this information and keeping it sufficiently up to date to be useful in making economic decisions. Second, he emphasizes that his system consists of an enormous number of equations. Solving them would require a significant amount of time, labor, and resources. He concludes that while such computations are possible in principle, they are not feasible in practice.

The rise of the electronic computer eliminated the need for people to solve by hand the massive system of equations that Barone describes. Certain advocates believed that the ability of computers to perform computations quickly meant that the time and resources required were no longer insurmountable. Oskar Lange (1967), for example, declared nearly sixty years ago that an electronic computer could solve the necessary equations in "less than a minute." Ceding ground on computational speed, those opposed to central planning tended to emphasize the Hayekian and Misesian arguments—that the necessary data is not available under central planning—instead of debating whether the technology was capable of solving the necessary system of equations in a reasonable amount of time.

Engelhardt calls these assumptions about technology into question. While Lange makes a specific claim about the time required, he gives no argument to defend it. Engelhardt simplifies Barone's problem into one of how to distribute goods, ignoring production, to consumers on the basis of known quadratic utility functions that reflect their preferences. The incentive, knowledge, and calculation problems are all assumed away, and the result is a system of linear equations, which Engelhardt proposes solving using Gaussian elimination. While many numerical methods of solving systems of equations require a guess-and-check process, Gaussian elimination provides the most precise answer possible without needing to test what could be a long series of possible solutions. And because the number of operations it requires is predictable, it is possible to calculate the amount of time a computer of a particular speed will require to solve the problem.

With these tools in hand, Engelhardt considers three scenarios and tabulates the amount of time it would take the combined TOP500 supercomputers (in 2013) to solve the system of equations. In the first scenario, one thousand heterogeneous goods are divided among one thousand heterogeneous consumers. This computation would take about three seconds to solve. In the second scenario, the number of goods is decreased to one hundred, but the number of consumers is raised to three hundred million, slightly less than the population of the United States in 2013. This computation would take 9.7 million years. The third scenario considers eighty thousand goods (the number tracked by the US Bureau of Labor Statistics when they calculate the consumer price index) and six billion consumers (slightly less than the world population at the time). This computation would take 10.5 quintillion years, an obviously impractical amount of time for solving large-scale economic problems. In short, simply accounting for the fact that different people have different preferences for different goods implies that there is so much data that even this pure distribution problem with no production and no uncertainty requires a massive amount of time to solve.

Why does the computation time explode beyond the case of one thousand goods and one thousand consumers? Gaussian elimination scales quite poorly: computation time increases by the third power of the size of the problem. Thus, a problem that is ten times larger takes one thousand times longer to solve. Engelhardt concludes that the market entails a "division of computation." Each individual consumer (and, if we add in production, each entrepreneur) solves a small part of the overall problem, which makes such a computation by the market feasible.

Cottrell calls Engelhardt's argument into question, noting two issues with the algorithm that Engelhardt chooses: (1) Gaussian elimination scales poorly, and an iterative technique could perform far better in this case. Such techniques require several guess-and-check rounds to arrive at an acceptable solution, but if each iteration is sufficiently faster than Gaussian elimination, then we are better off using an iterative method. (2) Gaussian elimination is not a valid solution algorithm for the problem posed by Engelhardt. The algorithm works for systems of linear equations, but the problem posed by Engelhardt involves, in addition to linear equations, a large number of non-negativity constraints which take the form of inequalities which sometimes bind and sometimes do not. According to Cottrell, then, an alternative method is necessary to provide a valid means of solving the problem and desirable because of the poor scaling of Gaussian elimination.

Cottrell proposes an alternative algorithm, which assumes an equal distribution of goods, calculates the average marginal utility for each good, adjusts allocations to equate each individual marginal utility to the average, and scales them to match available resources. The scaled allocations provide the second guess, which is used to recalculate the average marginal utility for each good. Unlike Gaussian elimination, which is cubic in its scaling, Cottrell claims his proposed algorithm scales linearly so that increasing the size of the problem tenfold increases the processing time tenfold. Cottrell, using his own desktop computer with a single-core i7 processor, claims to solve Engelhardt's first scenario in 0.6 seconds, significantly less than the three seconds that Engelhardt suggested the combined TOP500 supercomputers would require to solve the same problem. While Engelhardt's second and third scenarios are substantially larger than the first, supercomputers are substantially more powerful than Cottrell's desktop. The speed of modern supercomputers is measured in petaflops. A one-petaflop machine can perform one quadrillion floating-point operations in one second. Cottrell estimates that the large problem of the third scenario could be solved in about five minutes by a one-petaflop machine, which is about one one-thousandth as powerful as the most powerful supercomputer currently available.

COTTRELL'S ALGORITHM CORRECTED

Cottrell challenges the belief that computation is a difficult problem, but he makes a significant simplification error when conceptualizing it. The problem posed by Engelhardt explicitly allows interactions between goods in the utility function. "Utility functions will be assuming a quadratic form, with some interaction among goods being allowed (so the utility of consuming one good may be affected by the quantity of another good consumed)" (Engelhardt 2013, 231-232). But, as may be seen in the following equation, Cottrell's algorithm ignores such interactions:

$$U_i = \sum_{j=1}^{m} (a_{ij}q_{ij} + b_{ij}q_{ij}^2)$$

This equation says that the utility experienced by person i is the sum of utilities for each individual good j, where the utility from good j depends on the amount of it consumed by person i and the square of that amount. Here, Cottrell's version deviates significantly from Engelhardt's description and dramatically simplifies the problem because its concept of utility involves no interaction between goods—for example, it does not account for the utility from a baseball bat depending on whether its owner also has a baseball. Even using Engelhardt's inefficient method of solving the problem, Cottrell's formulation decreases the required time by a factor of the square of the number of goods because Gaussian elimination's computation time grows at a rate of the cube of the number of equations involved. The equations in Engelhardt's original problem equal the number of goods, *m*, multiplied by the number of consumers, *n*, so that the computation time is $(mn)^3$. Cottrell changes the structure of the problem. With no interactions, the distribution of each good can be treated as a separate problem: one can optimize the distribution of baseballs first, then bats, then gloves, and so on, without any one problem affecting the others. As a result, rather than one big problem with *mn* equations to solve, there are *m* separate optimization problems, each with *n* equations to solve. The computation time for each problem will be proportional to n^3, so the computation time for solving the *m* problems is proportional to mn^3. Because it is no longer necessary to cube the number of goods, Engelhardt's first scenario can be solved in three one-millionths of a second—even with the inefficient scaling of Gaussian elimination. An interaction term can be added to Cottrell's algorithm with a small tweak to the above equation:

$$U_i = \sum_{j=1}^{m} (a_{ij} q_{ij} + \sum_{k=1}^{m} b_{ijk} q_{ij} q_{ik})$$

This equation says that the utility experienced by person *i* is the sum of utilities for each individual good *j*, where the utility from good *j* depends on the amount of it consumed by person *i* as well as the product of the amounts of good *j* and good *k*. If *j* is equal to *k*, then the product is the square of the quantity of good *j*. The important point here is that one good affects the utility of another good.

The corrected equation scales worse than Cottrell's simplified version. Typically, algorithms, like the supercomputers that use them, are rated according to the number of floating-point operations required. In Cottrell's conception, each person–good pair requires four arithmetic operations, so the total number—and, therefore, the processing time—scales proportionally with the number of people and with the number of goods.[4] But, in the corrected conception, the

[4] We assume that the proportion of arithmetic operations that are floating-point stays stable as the size of the problem increases. This may not be strictly true, but Cottrell's experiments are consistent with this assumption.

number of operations required per person–good pair itself grows with the number of goods because each good interacts with the other goods in the utility function. Specifically, the number of arithmetic operations required for calculating an individual's utility function can be found using the expression $2m + 3m^2 - 1$, where m is the number of goods. For example, for one baseball bat, we only need to compute the utility from that one good, a process which requires four arithmetic operations. In this case, the utility equation is as follows:

$$U_i = a_{i1}q_{i1} + b_{i11}q_{i1}q_{i1}$$

Because there is only one good, the sums turn into single terms, and we can easily count the number of operations—in this case, one addition and three multiplications—that must be performed. But given a baseball bat and a baseball, the number grows significantly, rising to fifteen operations. Here is the utility equation after expanding the summations:

$$U_i = a_{i1}q_{i1} + b_{i11}q_{i1}q_{i1} + b_{i12}q_{i1}q_{i2} + a_{i2}q_{i2} + b_{i21}q_{i2}q_{i1} + b_{i22}q_{i2}q_{i2}$$

The first three terms describe the utility from the first good, and the last three describe the utility from the second. Adding up the arithmetic operations, one multiplication in each of the first and fourth terms and two multiplications in each of the other four terms plus five addition operations to sum the terms gives a total of fifteen. The expression $2m + 3m^2 - 1$ generalizes this equation. The m represents the number of goods, each of which has its own first term requiring one multiplication. The m^2 represents the number of interactions between goods, each of which has a term requiring two multiplications. The number of additions will be $m + m^2 - 1$, since there is a term for each good and each pair of goods, and the number of additions for a certain number of terms is that number of terms minus one. Thus, while the number of goods doubles from one to two, the number of operations more than triples from four to fifteen. As the number of goods increases linearly, the square term grows exponentially at a rate that lets us approximate the number of operations to be in proportion to the square of the number of goods. This corrected algorithm grows proportionally with the number of people but exponentially by the square of the number of goods.[5] While this algorithm is still far more efficient than Gaussian

[5] We assume only interactions between pairs of goods and not interactions between triples—for example, peanut butter, jelly, and bread—or quadruples. Adding this

elimination, which was cubic for both people and goods, it is also far less efficient than the algorithm Cottrell uses to solve the simpler problem without interactions.

What kind of practical difference does this make? Consider the scenarios that Cottrell tried on his desktop (an i7 processor has a speed of about five gigaflops—about one two-hundredth the speed of a one-petaflop machine) but with some extrapolation:[6]

Figure 1: Cottrell's Scenarios Corrected

Scenario	Cottrell Results (experimental)	Cottrell + Interactions (extrapolated)
1 good, 1 consumer (extrapolated)	6×10^{-7} sec	6×10^{-7} sec
1,000 goods, 1,000 consumers	0.6 sec	600 sec (10 minutes)
10,000 goods, 1,000 consumers	6 sec	60,000 sec (≈16.67 hours)
10,000 goods, 10,000 consumers	60 sec	600,000 sec (≈6.94 days or about 1 week)

Here, even the simple problem of one thousand goods and one thousand people, solved in less than a second using Cottrell's version of the algorithm, takes ten minutes to solve when interactions are taken into account. Meanwhile, the problem that is ten times bigger for both goods and people takes ten thousand times longer than Cottrell claims because the time needs to be multiplied by the number of goods a second time.

What about Engelhardt's third scenario, which involves eighty thousand goods and six billion consumers? If we use the Cottrell

level of complexity would go further than Engelhardt did in his original paper and only strengthen the case made here.

[6] See appendix A for the details of the arithmetic in this section.

algorithm (without interactions) and the computing power of the TOP500 supercomputers in 2013, then the amount of computing time falls from the original estimate of approximately 10.5 quintillion years to a mere 0.006 seconds! In this case, it is not clear that computation is a problem for central planning. Adding interactions increases the time from 0.006 seconds to about eight minutes, fairly close to Cottrell's estimate of five minutes despite the errors in his setup. Scaling, however, makes a significant difference for the computation time when we account for the true size of the problem in the modern world.

UPDATING THE COMPUTATION PROBLEM

The computation problem is inherently technological, and, like any other technological problem, changes in available resources and technical know-how may lessen its difficulty or even make possible a practical solution. Adding large numbers became significantly easier with the abacus, slide rule, pocket calculator, and modern computer. Cottrell's critique demonstrates that efficient algorithms that scale well can help computers solve problems faster, but increases in processing speed can achieve the same end. In 2013, the combined processing power of the TOP500 supercomputers was 233 petaflops (approximately 233 quadrillion floating-point operations per second). Today, the single fastest supercomputer is about four times as powerful. Rated at one exaflop, it can perform one quintillion floating-point operations per second. What, in the past, was impossible for five hundred supercomputers together is now surpassed by a single supercomputer. Surely, the computation problem—whatever its size right now—will continue to shrink as technology improves.

However, from the perspective of the world as it is today, Engelhardt understates the size of the problem in two ways. First, he uses a world population figure of six billion, but today there are about eight billion people on Earth. Second, the number of goods Engelhardt chooses—eighty thousand—drastically underestimates the real number in circulation. While the US Bureau of Labor Statistics tracks eighty thousand goods for the consumer price index, Amazon.com sells about 373 million unique products, including those sold by third parties on Amazon Marketplace (T., Bradley 2023).

In addition to the undercounting of consumer goods, Engelhardt also completely omitted producer goods. To get around the calculation problem, Engelhardt assumes that all production is a one-step process so that producer goods (and producers themselves) are not included in the total number of goods in circulation. Formulaically, there is a fairly simple way to incorporate these goods into the computation problem without having to commit to any specific production function: we will treat the distribution of producer goods among producers as a part of the overall problem of the distribution of goods. In other words, we make no distinction between consumer goods and producer goods. The problem then involves the distribution of consumer and producer goods among users seeking direct end-satisfaction and producers seeking to use goods for producing other goods.

The structure of the problem stays the same if we make the further assumption that a "pseudo-utility" function, similar to the utility that applies to consumers, applies to producers. One could interpret this pseudo-utility function as an indirect, derived, or imputed utility coming from the future consumer goods expected from the production processes. For example, eating fruit off a tree is single-stage, direct consumption. Someone who picks berries and stores them for later performs a two-stage production. Three stages of production take place—and require three separate calculations—when someone picks berries and trades them to a second person who eats them. Our argument does not consider the interconnections and sequences of transactions but only the sheer number, or volume, of transactional pairs, each of which consists of the user of the good (whether the final consumer or a producer using a producer good) and the good itself.[7]

Based on the following assumptions, let us consider a new scenario that reflects current realities.

(1) Computations are performed on a one-exaflop machine.

When combining processing power from multiple computers, data transfer between computers can drastically slow down

[7] An actual solution to the problem would require data about the interconnectivity of the various goods—that is, the complements and substitutes of a good as well as its alternative uses. The argument presented here is only concerned with the total number of transactional pairs that need to be calculated.

computing time. The Frontier supercomputer at the Oak Ridge National Laboratory is rated at 1.1 exaflops, based on actual performance, which includes any data transfer that occurs during the computation process. This machine combines just under 48,000 processors, and much of its speed comes from the parallel operation of the processors. If a process cannot be parallelized, then it would not benefit from the full speed. But in the corrected Cottrell-style algorithm, it is possible to parallelize most of the calculations so that at least 48,000 computations could be performed simultaneously, allowing us to take full advantage of Frontier's processing capabilities.

(2) The computation time increases linearly for consumers and by the square of the number of goods.

This assumption is based on the correction to Cottrell's algorithm. For iterative solutions like this one, the "check" step cannot be skipped. The process of checking solutions scales proportionally to population and to the square of the number of goods so any guess-and-check iterative algorithm will scale at this rate or worse. In other words, a doubling of the number of people doubles the time and a doubling of the number of goods increases the time by a factor of four.

(3) The number of iterations is not dependent on the size of the problem.

Unfortunately, the number of iterations required to achieve a specific level of precision is typically unpredictable. A computer might even guess correctly on the first try. Trials using a Cottrell-style algorithm show no clear relationship between the size of the problem and the number of iterations required to find a solution, so we will assume there is no relationship at all.

(4) We assume that there are 8.2 billion consumers and 373 million transactions (goods).

Today, the world population is estimated at just under eight billion (Moore 2022). We will treat businesses as consumers and add to this the number of businesses, which is estimated at just over three hundred million worldwide. The number of goods matches the number listed on Amazon.com (Amazon Scout 2023). This assumption makes the rather unrealistic supposition that every good in the world is a consumer good and is listed on Amazon.com, but it gives us a starting point.[8]

Using the corrected version of Cottrell's guess-and-check iterative algorithm, and assuming 8.2 billion consumers, 373 million goods, and a one-exaflop machine, the calculation would require over three quintillion equations and take approximately 108,529 years to compute. While this is a shorter amount of time than 10.5 quintillion years, it is still not practical.[9]

One option to make a solution more practical is to include multiple time periods in the computation. If a computer provides a distribution plan that keeps the economy going for the period it takes to compute a new plan, then the impracticality is less severe. For example, a program that computes a plan for every day of the week only needs to find a solution once every seven days. But this is not workable if the same physical goods in different time periods are treated as different goods. Time-indexing would increase the number of goods, and increasing the number of goods increases the computation time proportional to the square of the number of goods. For example, if we divide the 108,529 years up into just two periods of about 54,000 years each, the number of goods involved in the calculation doubles—but this quadruples the computation time. This approach increases the severity of the computation problem and makes it impossible to solve.

A second option, which is far more practical, is to limit the number of goods. For example, suppose that a one-day planning period is sufficiently short. A supercomputer as powerful as Frontier could solve a problem involving 8.2 billion consumers and 59,272 goods in just under a day. This could be achieved by an actual reduction in the number of distinct goods or by treating physically different goods, such as books with different titles and authors, as effectively the same (we are already abstracting from the location of goods by assuming that transportation is free and instantaneous). But such

[8] Phelan and Wenzel (2023) suggest that there are, at minimum, one billion different goods and services offered in the U.S., based on the fact that about 75 percent of the American economy is service based. When we add labor into the system, because labor transactions must also be centrally planned as well, we could easily add another three to four billion transactions (goods) to the series of equations.

[9] The calculation for the 108,529 years is in appendix B.

a reduction and flattening of goods would defeat the purpose of replacing free markets with technosocialism, which is to enhance our lives and eliminate hardships.

One final approach to reduce the time it takes to solve the problem would be to significantly increase the computing power of supercomputers. We can always add more processors, but there is a limit on the extent to which the operations of a computation can be performed in parallel—some operations require the results of others before they can proceed. But electricity presents an even more important limit. The amount of electrical energy used is the product of power (measured in watts) and time (measured in hours). Operating a seven-watt bulb for one hour uses seven watt-hours of electricity. Operating two such bulbs for an hour—or one bulb for two hours—requires fourteen watt-hours. For the very large problem of 373 million goods for 8.2 billion consumers, the Frontier supercomputer would require about twenty million gigawatt-hours (21 MW of power, multiplied by 108,529 years of calculation). If there is no improvement in the number of floating-point operations per watt of power, then faster computers, by necessity, will require more power. But there is a trade-off between computation time (hours) and the power consumption in any instant (gigawatts). To decrease the number of hours to, say, twenty-four, requires a proportional increase of the number of watts. To produce twenty million gigawatt-hours of energy in twenty-four hours requires a little less than one million gigawatts of power. This presents a practical problem: worldwide electricity output in 2022 was nearly three thousand gigawatts.[10] In short, power production would need to increase significantly.

How can we produce an additional one million gigawatts of power solely dedicated to one function? Wind turbines are a green option. Based on the available data, about fifty acres are required for each megawatt of wind-generated power (Rosenbloom 2006). Using this estimate, about fifty billion acres of wind turbines could generate the necessary power. The total area of the surface of the earth is about 126 billion acres, of which about 70 percent is water

[10] Nearly thirty thousand terawatt-hours of electrical energy were generated in 2022 (Ritchie and Roser 2020). Dividing this by the number of hours in a year results in an average power generation of just over three terawatts, which is equivalent to three thousand gigawatts.

(88.2 billion acres). Covering the earth's oceans with offshore wind turbines would be more than sufficient to provide the needed power, but since wind is not a consistent power source, we would need more than fifty billion acres to ensure a consistent supply of at least one million gigawatts. Wind power is not a practical solution.

A second option is to use solar power, which tends to use space more efficiently. Estimates for how much space a solar farm requires to create a megawatt of electricity vary, but five to ten acres per megawatt is a common estimate (Solar Energy Industries Association n.d.). Based on the lower estimate, a solar farm requires about five billion acres, an area slightly larger than Russia, carefully placed to ensure proper sun exposure throughout the day (a five-hundred-kilometer band around the equator would be about right, stretching from about 2.5 degrees south latitude to 2.5 degrees north latitude[11]).

A third option is to build nuclear fission power plants, which are quite efficient in terms of land space. They require approximately fifty acres to generate one gigawatt of power. Thus, one million gigawatts would only require fifty million acres of land—an area similar to that of Idaho.

Now, let us relax some of our assumptions. We have assumed that all this production and distribution can be done without labor. But giving each person one task per day increases the number of economic transactions (which we are calling "goods"—in this case, labor goods) by the size of the world's population. The problem of 8.2 billion consumers and 373 million goods is now 8.2 billion consumers and 8.6 billion goods, a twenty-three-fold increase of transactional pairs. We would need 529 times as much power (about 529 million gigawatts) to solve this problem in twenty-four hours—and therefore the land equivalent of 529 Idahos of nuclear power plants (28.3 billion acres or approximately 75 percent of the Earth's land area). If we assume that people perform two tasks per day, then the problem involves 8.2 billion consumers and 16.8 billion goods (again, twice as many transactional pairs). Doubling the number of goods again multiplies the electrical energy requirement by a factor

[11] Earth's circumference at the equator is about forty thousand kilometers. A five-hundred-kilometer band around this would make twenty million square kilometers, each containing about 250 acres for a total of five billion acres.

of four for a total of over two billion gigawatts of power and 113.2 billion acres, which is approximately 90 percent of the entire area of the Earth's surface—land and water. As noted above, the computation scales proportionally with population but increases by the square of goods. But once we acknowledge that people are both consumers and producers, and that the labor of each individual is a unique good, then the scale returns to being approximately cubic. The vastness of the number of economic transactions precludes any real possibility of solving this problem.[12]

Solving a distribution problem for the world population and all the goods available on Amazon.com requires improvements not only in the speed of supercomputers but also in their energy efficiency. In recent years, advances have been made in quantum computing, but a quantum computer has yet to take the top spot in processing speed. Quantum computers also have several practical limitations. They are fast and good at solving optimization problems, but they require a near-absolute-zero temperature to avoid heat-induced error (Zewe 2023) and a near-perfect vacuum (Leybold 2020). It remains unclear whether quantum computers will outshine the most powerful classical computers in terms of speed and energy efficiency, but they cannot currently overcome the computation problem of central planning.

What, then, is the solution? Engelhardt suggests that the market could provide, along with the division of knowledge, a division of computation. If the large problem is broken into smaller pieces, it could be solved in a reasonable amount of time. The market would use prices to summarize the information needed from other pieces of the larger problem. In fact, even Cottrell recognizes the value of markets for consumer goods: "All historically existing socialist economies, and almost all socialist theorists, have seen the distribution of personal consumer goods as a job for a market of some sort." Furthermore, Cottrell's own proposals advocate using markets to distribute consumer goods (Cockshott and Cottrell 1993), with prices (though stated in labor tokens) set at market-clearing levels. In short, to avoid computation's central planning problem, one must abandon central planning.

[12] Note that these power requirements are only for solving the computation problem and do not include the requirements of any other type of production or consumption.

CONCLUSION

Fundamentally, the computation problem is, and has always been, a technological problem and therefore entertains technological solutions. We cannot immediately rule out the possibility that algorithms that are more computationally efficient will come along, but we also cannot immediately rule out improvements in processing speed—whether through the gradual improvements of supercomputers or through the introduction of entirely new technologies like quantum computers and the algorithms they enable.

Today, and for the foreseeable future, the computation problem is one more reason to deny the feasibility of central planning and affirm the value of markets. Even with the technological advances that have been made since Barone discussed the issue in 1908—or even since Lange in 1967—we do not yet have the technology to solve the size of problem that a central planner would face in the modern world. But even if such resources were available, would their best use be in central planning considering that there is already a spontaneous social order—buyers and sellers interacting in markets—that produces very good outcomes? We may wonder, after all the time, expenses, and uses of resources, whether technosocialist solutions are really that much better than a free market.

REFERENCES

Barone, Enrico. 2012. "The Ministry of Production in the Collectivist State." *Giornale degli economisti e annali di economia* 71, no. 2/3 (December): 75–112. http://www.jstor.org/stable/43828055.

Boettke, Peter J., and Rosolino A. Candela. 2023. "On the Feasibility of Technosocialism." *Journal of Economic Behavior and Organization* 205 (January): 44–54.

Cockshott, W. Paul, and Allin Cottrell. 1993. *Towards a New Socialism*. Nottingham, U.K.: Spokesman.

Cottrell, Allin. 2021. "Critique of Engelhardt on Planning." Working paper, Department of Economics, Wake Forest University, September 2. http://users.wfu.edu/cottrell/socialism_book/engelhardt_critique.pdf.

Dapprich, Jan Philipp. 2023. “Optimal Planning with Consumer Feedback: A Simulation of a Socialist Economy.” *Review of Political Economy* 35 (4): 1136–56. https://doi.org/10.1080/09538259.2021.2005367.

Engelhardt, Lucas M. 2013. “Central Planning’s Computation Problem.” *Quarterly Journal of Austrian Economics* 16, no. 2 (Summer): 227–46.

Hayek, Friedrich A. 1945. “The Use of Knowledge in Society.” *American Economic Review* 35, no. 4 (September): 519–30. http://www.jstor.org/stable/1809376.

Lambert, Karras J, and Tate Fegley. 2023. “Economic Calculation in Light of Advances in Big Data and Artificial Intelligence.” *Journal of Economic Behavior and Organization* 206 (February): 243–50. https://doi.org/10.1016/j.jebo.2022.12.009.

Lange, Oskar. 1936. “On the Economic Theory of Socialism.” *Review of Economic Studies* 4, no. 1 (October): 53–71. https://doi.org/10.2307/2967660.

———. 1967. “The Computer and the Market.” In *Socialism, Capitalism, and Economic Growth: Essays Presented to Maurice Dobb*, edited by C. H. Feinstein, 158–61. Cambridge: Cambridge University Press.

Leybold. 2020. “How Vacuum Enables the Future of Quantum Computing.” Leybold. September 28, 2020. https://www.leybold.com/en-us/knowledge/blog/how-vacuum-enables-the-future-of-quantum-computing.

Mises, Ludwig von. 1951. *Socialism: An Economic and Sociological Analysis*. Translated by J. Kahane. New Haven, Conn.: Yale University Press. https://mises.org/library/socialism-economic-and-sociological-analysis.

Moore, Derick. 2022. “U.S. Population Estimated at 334,233,854 on Jan. 1, 2023.” *America Counts: Stories behind the Numbers*, December 29, 2022. https://www.census.gov/library/stories/2022/12/happy-new-year-2023.html.

Nieto, Maxi. 2022. “Entrepreneurship and Decentralised Investment in a Planned Economy: A Critique of the Austrian Reading.” *Historical Materialism* 30, no. 1 (May): 1–31.

Nieto, Maxi, and Juan Pablo Mateo. 2020. “Dynamic Efficiency in a Planned Economy: Innovation and Entrepreneurship without Markets.” *Science and Society* 84, no. 1 (January): 42–66. https://doi.org/10.1521/siso.2020.84.1.42

Phelan, Steven E, and Nikolai G Wenzel. 2023. "Big Data, Quantum Computing, and the Economic Calculation Debate: Will Roasted Cyberpigeons Fly into the Mouths of Comrades?" *Journal of Economic Behavior and Organization* 206 (February): 172–81.

Ritchie, Hannah, and Max Roser. 2020. "Electricity Mix." *Our World in Data*, July 10, 2020. https://ourworldindata.org/electricity-mix.

Rosenbloom, Eric. 2006. "Areas of Industrial Wind Facilities." Online supplement to "A Problem with Wind Power." Industrial Wind Energy Opposition. September 5, 2006. http://www.aweo.org/windarea.html.

Solar Energy Industries Association. n.d. "Land Use and Solar Development." Accessed April 13, 2023. https://www.seia.org/initiatives/land-use-solar-development.

T., Bradley. n.d. "Amazon Statistics: The Key Numbers and Fun Facts." AMZScout. Accessed April 11, 2023. https://amzscout.net/blog/amazon-statistics/.

Zewe, Adam. 2023. "A New Way for Quantum Computing Systems to Keep Their Cool." MIT News, Massachusetts Institute of Technology. February 21, 2023. https://news.mit.edu/2023/new-way-quantum-computing-systems-keep-their-cool-0221.

APPENDIX A: COTTRELL'S ALGORITHM (CORRECTED) CALCULATIONS

The table results of 0.6, 6, and 60 seconds were all reported directly by Cottrell. To arrive at other estimates, we make use of the fact that computation time using Cottrell's algorithm is proportional to the number of goods and the number of consumers. To extrapolate for the one-good, one-consumer case we apply the following formula:

Time for 1 good, 1 consumer = time for 1,000 goods, 1000 consumers / 1000 / 1000 = 0.6×10^{-6}

Because there are no interactions to account for here, this time would also apply in the model that includes interactions. To calculate the times for the cases Cottrell found, we apply the fact that time increases proportionally to the number of consumers but by the square of the number of goods.

Time for 1000 goods, 1000 consumers = time for 1 good, 1 consumer * 1000 * $1000^2 = 6 \times 10^{-7} \times 10^9 = 6 \times 10^2$ = 600 seconds, which is 600/60 = 10 minutes

Time for 10,000 goods, 1000 consumers = time for 1 good, 1 consumer * 1000 * $10{,}000^2 = 6 \times 10^{-7} \times 10^{11} = 6 \times 10^4$ = 60,000 sec, which is 60,000/60 = 1000 minutes = 1000/60 = 16.67 hours

Time for 10,000 goods, 10,000 consumers = time for 1 good, 1 consumer * 10,000 * $10{,}000^2 = 6 \times 10^{-7} \times 10^{12} = 6 \times 10^5$ = 600,000 seconds, which is 600,000/60 = 10,000 minutes = 10000/60 = 166.67 hours = 166.67/24 = 6.94 days

For Engelhardt's third scenario with a Cottrell-style algorithm, we start with the fact that it takes Cottrell's desktop 6×10^{-7} seconds to solve the problem for a single consumer and a single good. But we need to make two adjustments, first for the size of the problem, then for the speed of the computers involved.

Using Cottrell's desktop:

Cottrell's linear scaling:

Time for 80,000 goods and 6 billion consumers = time for 1 good and 1 consumer x 80,000 x 6 billion = $6 \times 10^{-7} \times 8 \times 10^4 \times 6 \times 10^9$ = 288×10^6 seconds

Convert to using 2013's TOP500 supercomputers, which are 233,000,000/5 = 46,600,000 times faster than Cottrell's desktop:

2.88×10^6 seconds / (4.66×10^8) = 0.6×10^{-2} = 0.006 seconds

To adjust for interactions, we multiply this result by the number of goods to account for the fact that the time should be proportionate to the square of the number of goods.

c x 80,000 = 480 seconds = 480 / 60 seconds per minute = 8 minutes.

APPENDIX B: THE UPDATED COMPUTATION PROBLEM

Seconds required for one good, one consumer on Cottrell's desktop: 0.6×10^{-6}

Multiply by the number of consumers and the square of the number of goods to get the number of seconds required by Cottrell's desktop:

$0.6 \times 10^{-6} * 8.2 \times 10^9 * (373 \times 10^6)^2 = 6.85 \times 10^{20}$ seconds

A one-exaflop machine is about 1,000,000,000 / 5 = 200,000x as fast as Cottrell's desktop, so make that adjustment:

6.85×10^{20} / 200,000 = 3.42×10^{15} seconds

3.42×10^{15} seconds / 60 seconds per minute / 60 minutes per hour / 24 hours per day / 365 days per year = 108,529 years