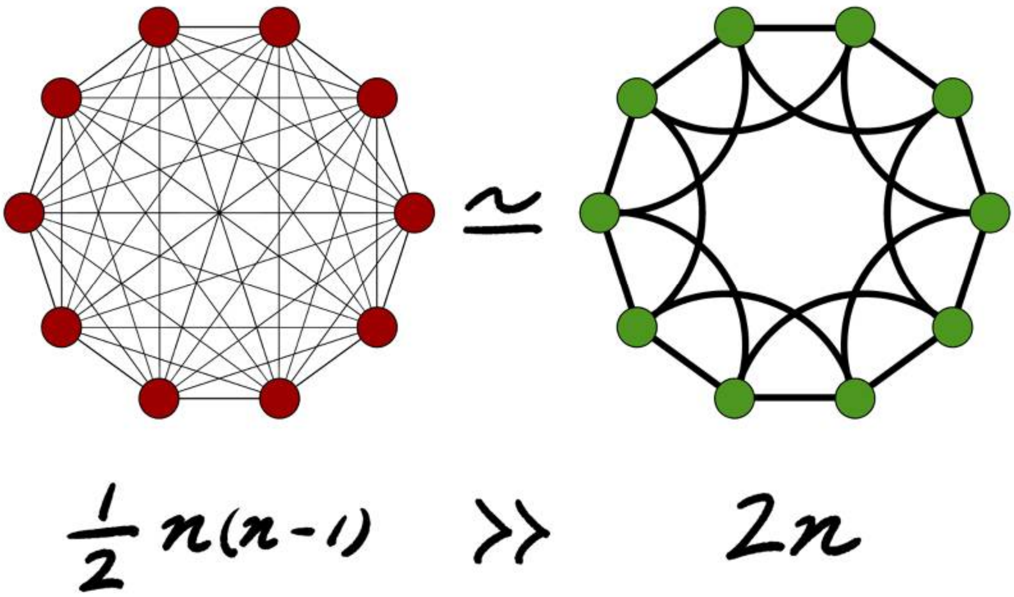
Title: Just Calc It? No!

Author: Junya YOKOKURA



Appropriate Approximation Can Relieve Combinatorial Explosion

Hello me at 12 years old! I'm you at 22 years old. Now I'm a first-year graduate student, and I'm majoring in computational physics. I know you think that science is roughly classified into two domains, theory and experiment. But "computation" is neither of them: rather, it is a domain bridging between the other two.

Many theories are written from "God's perspective": there's already a completely known world (or system, in the context of physics), and equations describe the rules that the system follows. The rules are usually highly generalized and abstract. They won't care about the complexity of the real world, especially real experimental instruments. Therefore, you need many calculations to apply the theoretical equations to the results of experiments. Of course you can do it manually, but in many cases, it is too heavy a task for humans. Then computers step in. Though they can only treat numerical tasks (at the lowest layer of the abstraction), in return, they do it 1,000,000,000 (10^9) to 1,000,000,000,000 (10^{12}) times faster than humans. So now what the experimenters must do is simple: convert the theory into an arithmetic program and run it on the computer! Thank you, computers! This is the dogma of computational science.

So, what is the catch? Computational science is nothing more than a handbook of supercomputers, isn't it? No. The problem is, "computers are still too weak to compute the world". Let us think about simulating molecular dynamics of water. You know water consists of water molecules. Molecular dynamic simulation is just like a simulation game, but instead of game units, we use molecules. Naïvely, you move the molecules slightly, then change their velocities slightly considering interactions and external force and repeat these two steps. But have you ever thought about the number of molecules? Just one drop of water consists of about 1,000,000,000,000,000,000,000 (1021) atoms! This far exceeds the capacity of any existing computers (recall the speed of calculation of computers I told)!

What is worse, "quantum" features make systems not just huge, but exponentially huge. Consider a system made of N identical objects. If we treat it as a "classical" system, each object is always in one definite "classical" state. The overall state of the system is also just a list of these definite states. In that case, we just need to handle the size of N, as I mentioned above. But if we treat the system as "quantum", we need to consider all possible "classical" states together to describe a single "quantum" state. Concretely, one "quantum" state is a combination of all "classical" states, each with a probability-like value. If one object can have M “classical” states, then the total number of possible “classical” states for the system is MN. So the “quantum” state of the system contains at least MN values. This means when N increases, the computational effort grows exponentially! For example, even a 50-particle system requires handling more than 1,000,000,000,000,000 (1015) numbers. In the quantum world, we can simulate only very tiny systems using naïve methods.

This is the problem I’m working on. I’m looking for tricks to let computers finish tasks in a realistic time limit (the limit varies depending on the problem, but it never takes more than one year). To achieve this, there are two common strategies.

One of the strategies is approximation. By reducing unimportant factors of the problem, we sometimes get alternative tasks easier to calculate. But there is a dilemma. Reducing too many leads to poor accuracy. Reducing too little results in slow performance.

Cracking this dilemma and achieving both speed and accuracy—that’s what excites me the most. I wish I could discover a new method someday!

The other strategy is using computers effectively. Though computers are powerful, they require special techniques to unlock their full potential. The key idea is “parallelization”. Think about a simple example from daily life. If many people harvest crops together, the work gets done faster. But what if they’re trying to draw water from a single well with one bucket? Since each person must wait the one before him, adding more people won’t help. It might even make things slower. Parallelization techniques treat this kind of race condition problem by breaking tasks into subtasks and reducing dependencies between them. In parallel computing, we need to control over 1000 units at once. This is difficult, but at the same time so exciting.

Finally, I will give you some advice, to make it easier for you to get through your career.

First, keep reading books. I remember you going to the local library every weekend and borrowing 10 or more books looked interesting. This habit has strengthened your ability to read bulks of text. However, you will get a smartphone next year, and that's the beginning of your lazy habit: you will be used to reading small texts and collecting information. This is also an important skill in the information society. But you eventually will be addicted to the Internet and lose interest in reading books. Now I get tired even reading articles that are less than 10 pages. I lost the chance to get knowledge just because of laziness! How awful. So, please keep the habit, and keep the ability to read long texts.

Second, you should get exercise. I know you don't like sports or exercise. But without any exercise, you will finally get tired during normal daily activities. You'll need more energy than other people to work. You don't have to do hard exercise, rather, easy exercise is better to keep doing. I recommend you develop a habit to take a walk every day.

Anyway, I hope you to live a better life. Good luck!

Acknowledgement:

I want to say thank you to my instructor, Mark Vagins, for his comments and advice. His feedback helped me understand which parts I needed to improve. I also want to thank my peer reviewer, Yosuke Asami, for checking my work and giving useful suggestions. His suggestion made it easier for me to fix grammar mistakes.

I really appreciate both for their help.

AI Usage:

I used ChatGPT for grammar checking.

Image Origin:

I drew it using ibisPaintX.