

Why is PolyChord unusually effective for Parameter Optimisation?

Using Optimising Satellite Constellations as an Example.

What kinds of trade-offs can PolyChord help you solve?

Suppose my main objective is to give excellent Wifi coverage. I need to accommodate:

- **Gravitational stability**
- **Orbital positioning**
- **Project design, development duration and related costs**
- **How many of the satellites are actually needed and if fewer can be deployed**
- **Connection to a ground control station**
- **Avoiding other objects in space**
- **Overall implementation costs**

*As smaller satellites are being launched in greater numbers, the business of managing their orbits becomes increasingly difficult. PolyChord optimises your launch, design and maintenance processes by uncovering the relationships, or "trade-offs", between different variables to gain your desired outcomes. PolyChord **leverages the value of the data you already have.***

PolyChord produces a map of the optimisation space, rather than just one solution.

Imagine the data landscape as a mountainous region where you want to find all of the peaks. Other tools drop one explorer into the landscape, till he finds a peak. This method, used by Google's TensorFlow and conventional "Deep Learning" only discovers one peak in the landscape - or one possible solution. The difference with PolyChord is we drop an army of explorers, automatically moving the explorer with the lowest altitude to a higher point, until all explorers have concentrated themselves around the peaks. This scouring of the landscape happens in an efficient single run, guarantees a thorough exploration of the landscape and simultaneously gives you a confidence value on your computed solutions. All the peaks are discovered.

We can then ask "questions" of this map to make more informed decisions about how to deploy the constellation against your specific goals.

In its simplest form optimising a satellite constellation can be done for cost, for efficiency, for

safety, for best coverage. **With PolyChord you can optimise for a combination of all of the above,** a task beyond most data tools.

The best laid plans...

Adding staged deployment of new satellites?
Adjusting distribution to meet varying demand?
Expanding the size of your constellation?
Is a new player disrupting the planned flight paths of your constellations?

PolyChord's map of the space can be adjusted, without needing to recalculate the entire solution - meaning you can add new satellites to the mix.

What's wrong with the way most tools try to solve these problems?

Most optimisation tools use "gradient information" - meaning they are tied to certain kinds of variables. Not PolyChord - integers and even categorical variables are no problem, meaning you can ask your optimisation map to exclude any solutions which pass over a particularly risky flight path, or no fly zone.