Sai Saketh Vootla

NUID: 001568984

**OPTIMIZATION PROBLEM WITH NVIDIA CUOPT**

An optimization problem is a mathematical problem in which the goal is to find the best possible solution from a set of available options, subject to certain constraints. In other words, it involves finding the "optimal" solution that maximizes (or minimizes) some objective function, given certain conditions or limitations. For example, an optimization problem might involve finding the cheapest way to produce a certain number of goods, or the fastest route to deliver goods to multiple destinations

NVIDIA CuOpt is a software library developed by NVIDIA that provides optimization algorithms for solving various types of problems. It is built on top of the NVIDIA CUDA platform, which enables it to take advantage of the parallel processing capabilities of NVIDIA GPUs (graphics processing units) to accelerate the performance of the optimization algorithms. CuOpt can be used for a wide range of applications, including machine learning, data analysis, and engineering design.

Parallel Heuristics is a term used in the field of artificial intelligence to describe an algorithm that can solve problems by using different strategies. A problem about finding the shortest route between two points, you could use one strategy to find it and another strategy to find the fastest route. The two algorithms will then work together on solving the problem. Optimization problems are incredibly compute-intensive with massive operational costs. GPUs bring the throughput capabilities needed to fuel the most ambitious parallel heuristics while supporting the most challenging constraints of the real world.

The NVIDIA CUOPT workflow helps to solve complex vehicle routing problems without the tradeoff of accuracy, it is very fast in comparison of the benchmark of other GPU’s, every iteration can be reoptimized as new tasks come. The Optimization develops a new route as with the real time changes such as weather, traffic and generates an optimized route very quickly. CUOPT is not just another SDK it delivers more dynamic planning to the simulation of massive operations.

A picture containing logo

Description automatically generated

*Fig 1(a): NVIDIA CUOPT WORKFLOW*

NVIDIA CUOPT can have hundreds of complex combinations with multiple variants such as Capacitated Workloads, Time Windows, Heterogenous, Periodic, Open routing and constraints such as Task and Vehicle priorities, Task to agent mapping, Precedence, Slack, Breaks. CUOPT can handle all these iterations and produce wonderful results within just few seconds.

CUOPT is based on a hardware/software design with GPU accelerated Parallel Heuristics and method heuristics. It uses Parallel methods to generate the initial solutions from various positions in a short space. The solutions are then further improved with the help of Parallel intensification and diversification. Other techniques are used to escape the local minima and find the best solution, once a budget is decided by the user the best solution is picked up by CUOPT and recommended.

Diagram

Description automatically generated

A parallel core engine is a type of computing architecture that is designed to allow multiple processing units, such as CPU cores or GPUs, to work together in parallel to perform tasks more efficiently. This can be useful for applications that require a lot of computing power, such as machine learning, data analysis, and scientific simulations.

In a parallel core engine, the workload is divided into smaller sub-tasks that can be processed simultaneously by different cores or processing units. This allows the overall task to be completed faster, as the different units can work in parallel to complete the sub-tasks. The results of the sub-tasks are then combined to produce the final result.

The Technique is natively mapped to the GPU architecture from multi-level parallelism perspective with also the memory hierarchy of the GPU. All the points are a set of tasks which come with a thread block of induvial root assignments

Graphical user interface

Description automatically generated with medium confidence

*Fig 2(a) – Parallel Core Engine*

The Parallel Core Engine works on a high-level parallelism having multiple blocks of individual set of threads to perform unique tasks to arrive at the optimal solution. Each threaded block consists of a feasible initial solution. Each threaded block improves on initial solution using multiple heuristics.

Graphical user interface

Description automatically generated

*Fig 2(b) – Parallel Core Engine*

* Each thread evaluates the cost improvement of a different move in short memory, that can be 2,3, 4, …n hops. The evaluation is done intraroute and interroute for the
* Checks for constraints violation of corresponding move, there are move executed with smart heuristics all this involves many parameters and hyperparameter tuning
* Execute moves based on the smart meta heuristics

As there are many blocks like this and each block performs all the tasks in a parallel fashion to give the results to CUOPT which then checks for the most optimized solution.

The performance is a key metric to compare the real-time working of NVIDIA CUOPT. The key factors which influence performance are-

1. Accuracy
2. Runtime
3. Scalability

Chart, bar chart

Description automatically generated

*Fig 3(a) – Performance across cuOpt runtime duration*

The performance is required to understand the difference in accordance with the previously used hardware/framework to observe how well it can perform the same implementation of task.