Using an Aggregating Index for Delivery

Introduction

The purpose of this document is to describe the features of a hierarchical hexagonal tree structure, in particular Uber's H3 index, and sketch a method by which we can use the index to localise marginal route length calculations analysis when building an "efficient" delivery route.

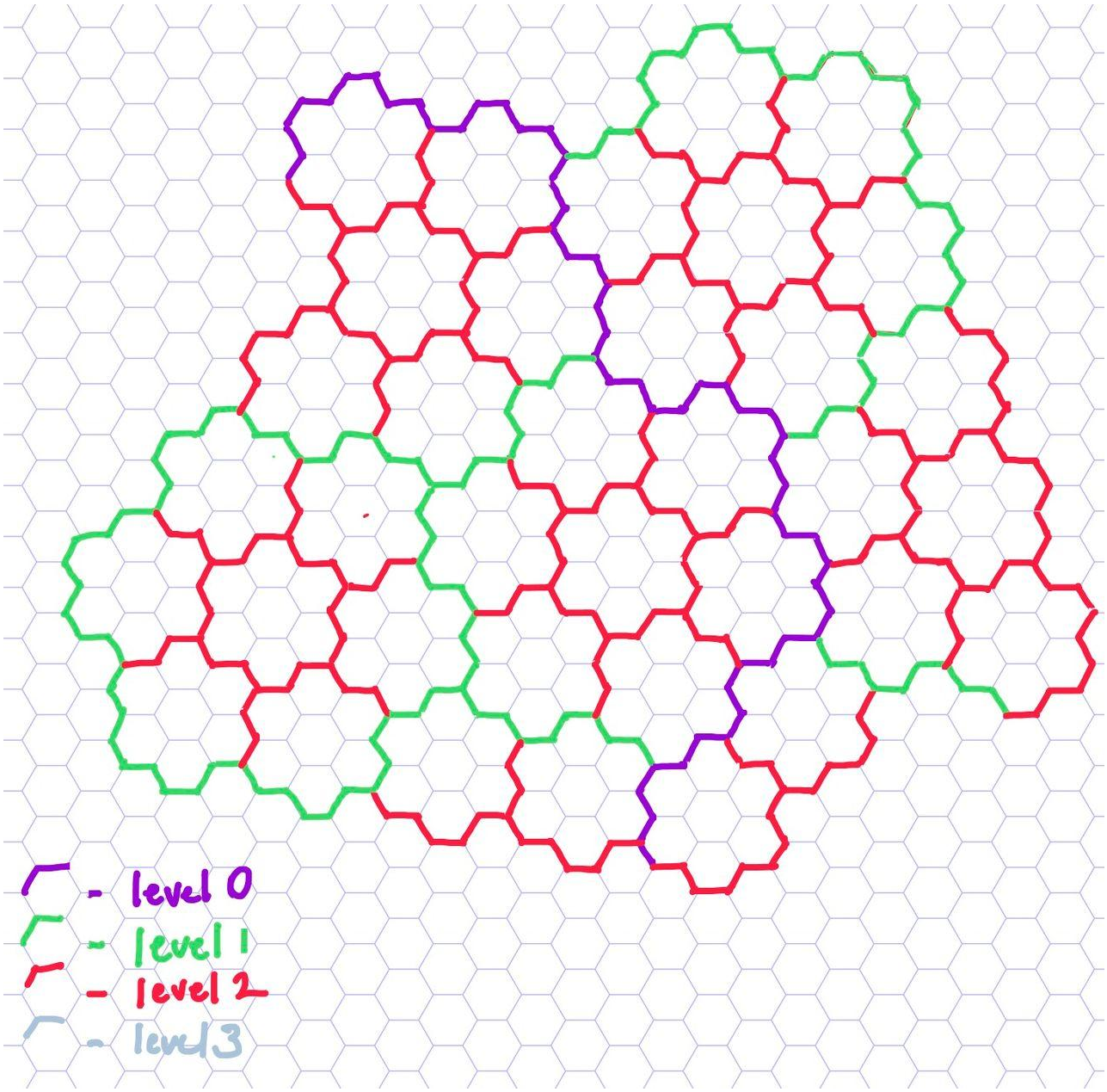
Geographical indices encode location information, including point position and area. Most geographical indices have a hierarchical structure that allows a user to operate at a scale appropriate for their purpose.

Examples of geographical indices are

1. Ordnance Survey grid reference - used as a basis for UK's comprehensive system of Ordnance Survey maps,
2. Amazon's S2 grid,
3. Uber's H3 grid.

HexTree Index Basics

Hextree levels can be represented as a graphical overlay over a region map



A hex tree has useful properties:

- nearly convex and nearly self-similar,
- distance between a cell and its neighbours is isotropic,
- hierarchal access to location data.

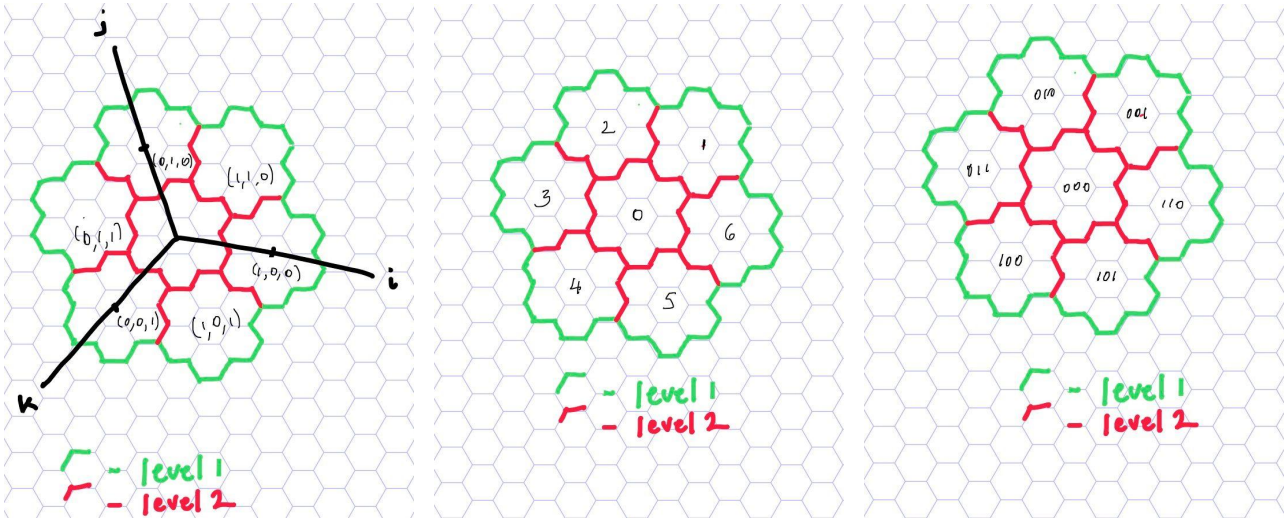
This document focusses on the use of hextree indices to navigate and select subsets of delivery points to enable efficient route-finding analysis. But it can be used in other ways

- auto-generate regions of interest from location features,
- maintain realtime metrics over a geographical area,

- <WIP>

Index Mappings

Some ways to refer to elements of a hextree



coordinate *orientation* *bitwise*

- The *coordinate* representation can be used in a similar way as a cartesian map, to access hex information in a non-heirarchial manner.
- The *orientation* representation is useful for generating transforms between representation layers and underlying cartesian representation.
- The *bitwise* representation is useful for compact representation of a hextree cell. H3 uses 48 bits to reference a 16-deep hextree that represents Sol 3. This tree contains 221 top level "hex-likes" that tile the planet, with a leaf resolution of $\sim 10^{-1}m^2$. So you can potentially use the H3 index to help you rearrange your house furniture.

Using an Index to Deliver

This algorithm assumes that road layouts within a region exploit convexity. The initial version of this algorithm will not take anticipated future volume of demand into account when making slot allocation decisions.

Assume that a marginal cost calculation on a regional route will be limited to calculate delivery routes for at most $N + 2$ locations at a time, for efficiency reasons.

Algorithm

1. Localise the region - locate the minimal hex that fully encloses region boundary points. This hex is the unique ***region base***. The illustrations below show a region base hex divided into three tiers of sub-hexes.
2. For each delivery request that is made, we add a delivery point P to the region.
3. Starting from the base hex, consider the number of deliveries within a hex. If the hex is not the region base, neighbouring hexes can also contain delivery points with associated delivery slots. For each hex retains a record of the first and last timeslot allocated.
   a. If a hex has $\leq N$ delivery points, compare the marginal cost for each feasible timeslot that the new delivery adds to the route, and pick the slot that with the minimal marginal cost.
      i. If the timeslot is $\leq$ the first timeslot for the cell, then we consider the last timeslot for other cells whose last timeslot $\leq$ the first timeslot. Similarly,
      ii. if the timeslot is $\geq$ the last timeslot for the cell, then we consider the last timeslot for other cells whose first timeslot $\geq$ the last timeslot.
   b. If a hex has $> N$ delivery points, consider the child hex that contains the new delivery point.
      i. If the child hex contains only the new index, use a route structure that takes the average delivery position for each of the other child hexes that are nonempty, and assign the new delivery point to a timeslot that minimises the marginal cost of the augmented route.
      ii. If the child hex has $1 < n \leq N$ delivery points, make the child hex containing the delivery point, the focus and return to **3a**.
      iii. If the child hex has $> N$ delivery points, make that child hex the focus and return to 3b.

Example

In this example, we have timeslots labelled in order $S = \{1, 2, 3, 4, 5\}$. We set the optimiser limit at 5.

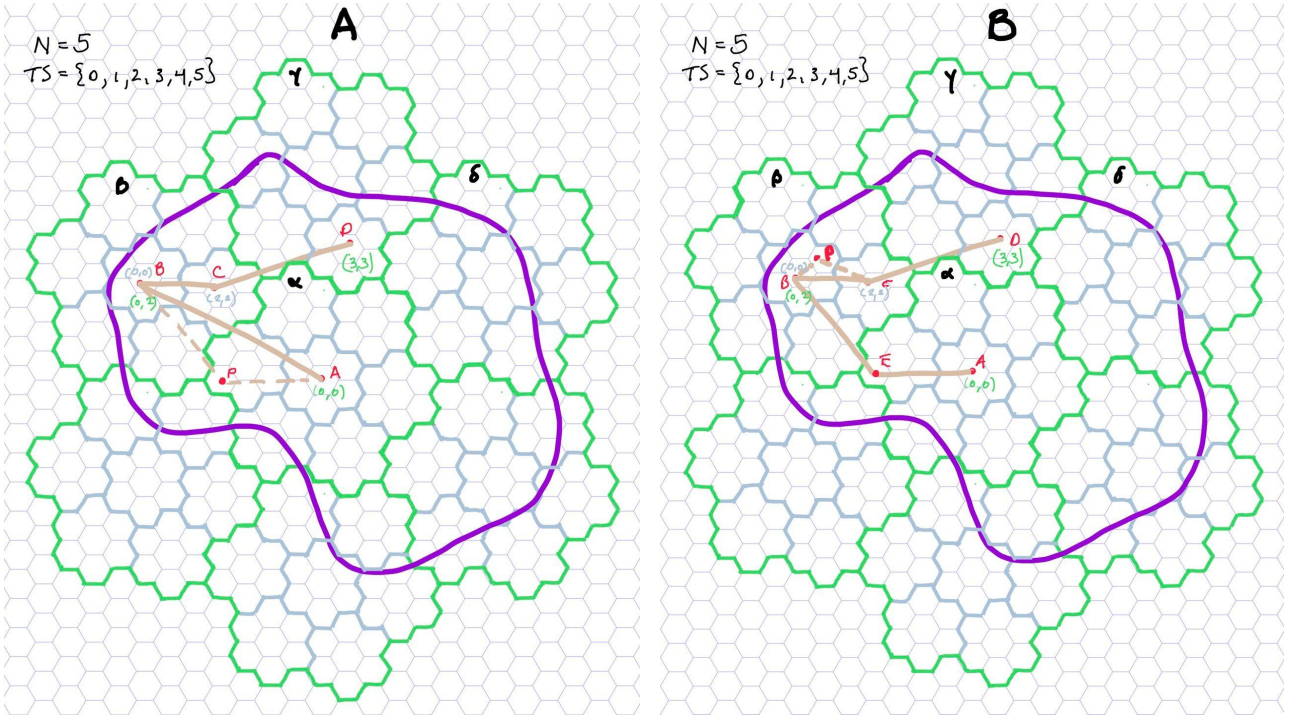
In **A**, delivery points $\{A, B, C, D\}$ and the additional delivery P are all part of the route calculation, as the number of points in the region base $\leq N = 5$. Compare the

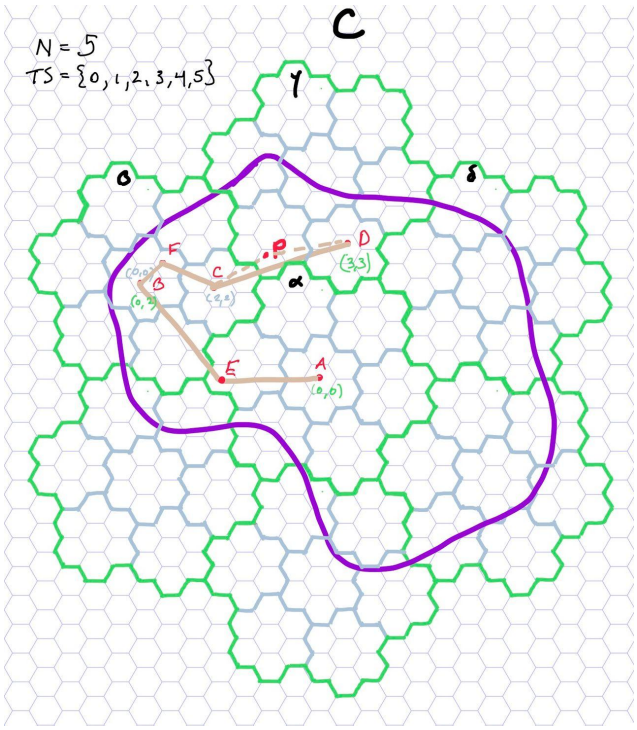
minimising timeslot $\arg\min_{(P,s\in S)} L(P,s)$ that picks the timeslot providing the shortest route connecting A, B, C, D, P.

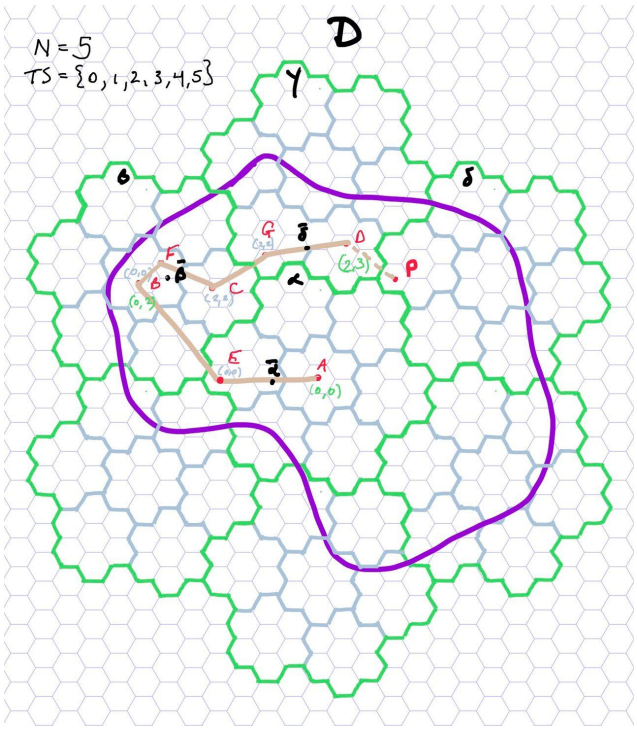
In **B**, delivery points $\{A, B, C, D, E\}$ and the delivery P to be added exceed 5 in number, so we consider the child hex $\beta \ni P$. We consider the delivery points $\{B, C, D, E\}$ and the timeslots $S = \{0, 1, 2, 3\}$. E is an active constraint when $s = 0$ and D is an active constraint when $s = 3$.

In **C**, delivery points $\{A, B, C, D, E, F\}$ and the delivery P to be added exceed 5 in number, so we consider the child hex $\gamma \ni P$. We consider the delivery points $\{C, D\}$ and the timeslots $S = \{2, 3\}$. C is active for all $s \in S$. Notice that we update the timeslot interval for γ to $(s_{\text{first}}, s_{\text{last}}) = (2, 3)$.

In **D**, delivery points $\{A, B, C, D, E, F, G\}$ and the delivery P to be added exceed 5 in number, so we consider the child hex $\delta \ni P$. There are no other points in δ, so we calculate a route based on the "delivery centroids" $\{\overline{\alpha}, \overline{\beta}, \overline{\gamma}\}$ for each hex in the parent hex (the region base here), combined with P, for each timeslot $S = \{0, 1, 2, 3, 4, 5\}$.







References

1. H3: Uber's Hexagonal Hierarchical Spatial Index