ReadMe for gc_cover_py3_v2 iPython notebook

Jeff Nivitanont, University of Oklahoma

March 12, 2018

1 Introduction

Packages:

- netCDF4 - used to read in netCDF4 formatted files
- NumPy - part of the SciPy suite, provides data structures such as n-d arrays and series
- pandas - Python Data Analysis library provides easy to read and manipulate DataFrame
- shapely - Python wrapper for C-GEOS (Geometry Engine - Open Source), which is a port of the JTS (Java Topology Suite)
- GeoPandas - Geospatial Data Analysis library built from pandas
- matplotlib - Primary plotting library for python
- CartoPy - add-ons for plotting geospatial information in matplotlib
- descartes - add-ons for plotting polygons in matplotlib

We are given the problem of optimizing the scan pattern for the GeoCARB satellite instrument set to launch in 2022. GeoCARB is a geostationary satellite that will be positioned at 0-85W and make daily observations of atmospheric CO2, CO, NH4, and SIF over the Americas. The underlying mathematical problem related to optimizing the scanning pattern is called the Geometric Set Cover problem. Given a set of points in 2-D space and a set of covers, what is the optimal covering set? This problem is known to be NP-Hard so there are no efficient methods for straight-forward computation. However, there are heuristic methods we can utilize for finding a set cover, though they do not necessarily yield optimal covers. We start our exploration into this problem with the Greedy Heuristic.

2 Cover Set

We use the shapely package to create the scan block Polygons from the coordinates of the corners from the scan blocks indicated in the netCDF4 file. We plot them on our map, shown in Figure 1 using the matplotlib and CartoPy libraries.

3 Universe Set

Using built-in GeoPandas dataset we select North America and South America geometries for our set to be covered. The Data is 1:110m from NaturalEarth and includes cultural and physical information (i.e. countries, pop, geometries, etc.). NaturalEarth data is a free for use and supported by NACIS (North American Cartographic Information Society). After obtaining our geometries, we truncate our set to only include land between 50N and 50S and exclude eastern Nova Scotia, which lies outside our scanning blocks, shown in blue in figure 2.

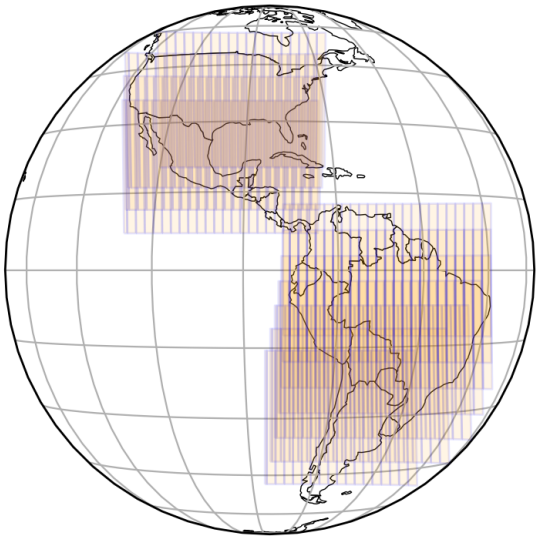

Figure 1: The set of covers.

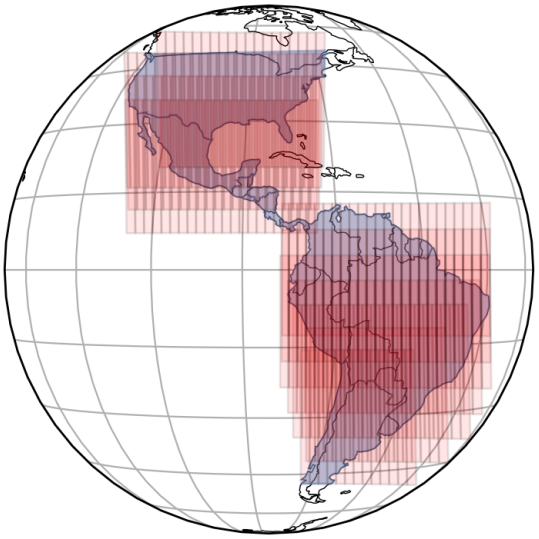

Figure 2: Universe set in blue.

4 Greedy Algorithm

After establishing our universal set to be covered and having our set of subcovers (scan blocks), we are ready to implement the Greedy Heuristic Algorithm for finding covering sets.

4.1 Greedy Algorithm with Weight Function

$\mathbb{U}$ = Universe Set
$\mathbb{S}$ = Set of Covers
$\mathbb{W}$ = Weight Function
begin
  $\mathbb{I} = \emptyset$
  $\mathbb{C} = \mathbb{U}$
  while($\mathbb{C}$ not empty**):**
    $\mathbb{L} = s_i \in \mathbb{S}$, such that $\mathbb{W}(s_i)$ is the maximum of $\mathbb{W}(\mathbb{S})$
    $\mathbb{I} = \mathbb{I} \cup \mathbb{L}$
    $\mathbb{C} = \mathbb{C} \setminus \mathbb{L}$
  return $\mathbb{I}$

4.2 Weight Function

Our current weight function is as follows:

$$w(s, \mathbb{I}, \mathbb{C}, \mathbb{L}) = (s \cap \mathbb{C})^2 - (s \cap \mathbb{I})^2 - d(s, \mathbb{L})$$

$$\mathbb{W}(s) = \begin{cases} \frac{w(s)}{\text{Airmass Score}}, & \text{if } w(s) \geq 0 \\ w(s) * (\text{Airmass Score}), & \text{if } w(s) < 0 \end{cases}$$

5 Results

After running the algorithm, we obtain a covering set from the algorithm stored in a GeoPandas.GeoDataFrame named "coverset" plotted in red in Figure 3.

6 Details

More detailed descriptions of the code can be found in the notebook.

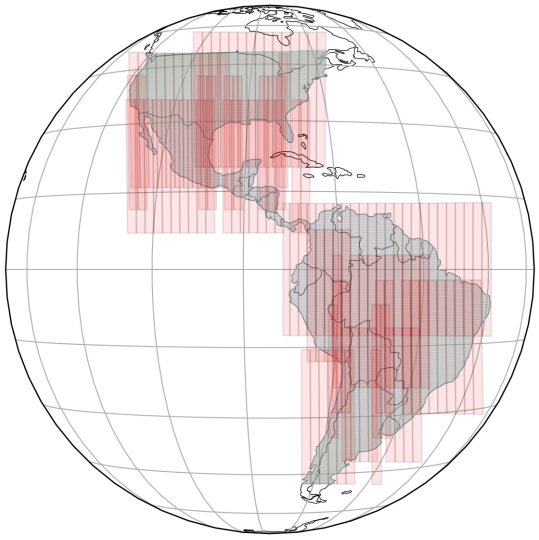

Figure 3: Scan blocks selected by algorithm.