Engr. Faisal Rehman
faisal@uetpeshawar.edu.pk
Syed Haseeb Shah
*Department of Civil Engineering*
University of Engineering & Technology, Peshawar
Engr. Irfan Ahmad
*irfan.ahmed@uetpeshawar.edu.pk*


# Monte Carlo based Algorithm for Scoring Grid Systems

*Abstract*—We evaluate Monte Carlo based algorithm for scoring grid systems. The former were used to describe features of the area under study. This was done to select suitable features for City Information Modelling. These algorithms measures path of shortest distance. However, shortest distance does not always ensures shortest time. The algorithm works by pairing multiple iterations of two random numbers in a polygon. The mean of shortest path between these points is calculated. Thus grid systems with shorter paths will return a lower value as compared to grid systems with longer paths. This was applied to GIS systems in urban planning to rate the feasibility of different grid systems. It is a heuristic algorithm that tells us to about the nature of grid system. Instead of measuring the geometry of every path in a grid it eases the workflow of an urban planner by automation.

***Index Terms*—**Urban Planning; Grid Scoring

## Introduction

This paper defines a new feature for city information modelling by measuring the average length of N randomly generated paths in a grid system. A grid system with mostly longer paths would have a greater normalized local score (NLS). For benchmark, the algorithm was applied to different street systems namely gridiron, loose grid and radial grid. Monte Carlo method, Dijkstra algorithm were utilized for the task. It was found that radial grid systems are the best in comparison to loose grid and grid iron. Grid iron scored the worst because of shorter paths that are not allowed to move along diagonally.

### A. *Random Points Generation:*

After specifying polygonal boundary of a city, 1000~5000 random points (latitudes and longitudes) were generated inside the polygon. These points were paired at randomly in the form [(lat, long), (lat, long)].

### B. *Dijkstra Algorithm:*

The shortest path between the algorithms was calculated using the processing toolbox. Dijkstra algorithm (Mehlhorn & Sanders, 2008) finds the shortest path between nodes of graph that have positive cost value. *Pseudocode* of the algorithms is as under;

```
function Dijkstra(Graph, source):

  create vertex set Q

  for each vertex v in Graph:
    dist[v] ← INFINITY
    prev[v] ← UNDEFINED
    add v to Q
  dist[source] ← 0

  while Q is not empty:
    u ← vertex in Q with min dist[u]

    remove u from Q
    // only v that are still in Q
    for each neighbor v of u:
      alt ← dist[u] + length(u, v)
        if alt < dist[v]:
          dist[v] ← alt
          prev[v] ← u

  return dist[], prev[]
```

To use the algorithm in processing toolbox paste the following code in Python shell of QGIS.

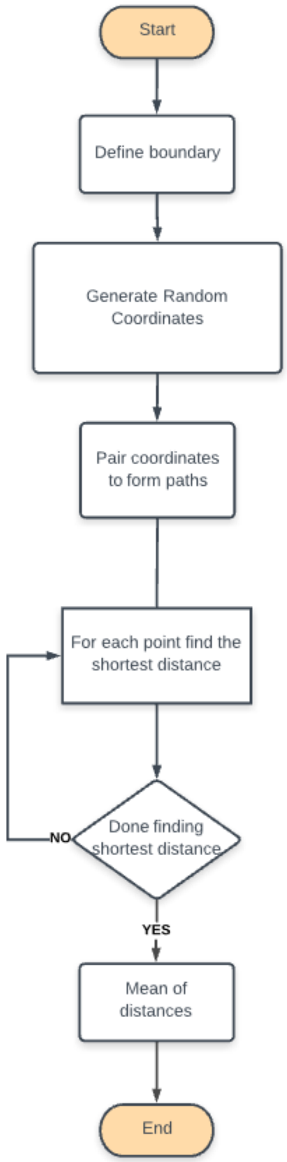
```
def shortest_path(a, b, c, d):
    res = processing.run(
"native:shortestpathpointtopoint",

{'INPUT':'<path_to_file>.geojson|layernam
e=NewOrleans',

'STRATEGY':0,'DIRECTION_FIELD':None,

'VALUE_FORWARD':'','VALUE_BACKWARD':'',

'VALUE_BOTH':'','DEFAULT_DIRECTION':2,

'SPEED_FIELD':None,'DEFAULT_SPEED':50,
                          'TOLERANCE':0,

'START_POINT':'{},{}'.format(a,b) + '
[EPSG:4326]',

'END_POINT':'{},{}'.format(c,d)  + '
[EPSG:4326]',

'OUTPUT':'TEMPORARY_OUTPUT'}
                         )
```

Where (a, c), (b, d) are longitude and latitude of the two points.

faisal@uetpeshawar.edu.pk *Department of Civil Engineering* irfan.ahmed@uetpeshawar.edu.pk
University of Engineering & Technology, Peshawar


### C. Monte Carlo-Dijkstra algorithm:

Flowchart of Monte Carlo algorithm coupled with Dijkstra algorithm was used to score the grid systems.



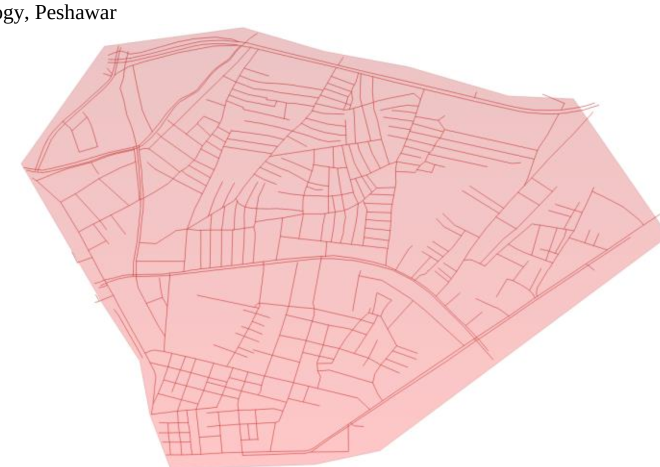



*Figure 1: Shakirabad Peshawar Pakistan*

| Shakirabad, Peshawar | | |
|---|---|---|
| Iteration | Points | Normalized Local Score |
| 1 | 229 | 0.290 |
| 2 | 232 | 0.339 |
| 3 | 228 | 0.331 |
| 4 | 231 | 0.306 |
| 5 | 231 | 0.300 |
| 6 | 4528 | 0.298 |
| 7 | 4563 | 0.284 |
| 8 | 2218 | 0.270 |
| 9 | 2238 | 0.298 |
| 10 | 2229 | 0.282 |

Mean NLS: 0.300*

**Grid Iron:**

Grid Iron allow paths along the rectangular axis. Diagonal path, which is the shortest between two corners of a rectangle is not allowed so the algorithm has to travel along the horizontal and vertical paths. The sum of lengths is obviously larger than the length of diagonal path. It's the primary reason why the NLS of grid iron is high.

### D. Results:

**Loose Grid:**

Loose grids have inclined roads as well as rectangular roads. This means that there exist a set points in the grid where diagonal path is allowed between two points. The path does not have to travel along rectangular grid therefor they have lower score in comparison to grid iron.

*Figure 2: Phase 6 Hayatabad Pakistan*

| Phase 6, Peshawar | | |
|---|---|---|
| Iteration | Points | Normalized Local Score |
| 1 | 200 | 0.447 |
| 2 | 200 | 0.437 |
| 3 | 500 | 0.449 |
| 4 | 499 | 0.456 |
| 5 | 2498 | 0.423 |
| 6 | 2499 | 0.436 |
| 7 | 2499 | 0.403 |
| 8 | 2499 | 0.438 |
| 9 | 2497 | 0.427 |
| 10 | 2499 | 0.415 |

Mean NLS: 0.433*

### *E. Radial Grid:*

The geometry of radial grid is such that it utilizes the benefits of equidistant paths from grid iron and shortest diagonal paths due to rotation of local grid irons. It is composed of multiple grid irons inclined to each other in a way that they all point to the center of the city. The inclination of grid iron provides a shorter path between different grid irons.

| Central New Orleans, US | | |
|---|---|---|
| Iteration | Points | Normalized Local Score |
| 1 | 2215 | 0.372 |
| 2 | 2236 | 0.375 |
| 3 | 2236 | 0.347 |
| 4 | 2245 | 0.366 |

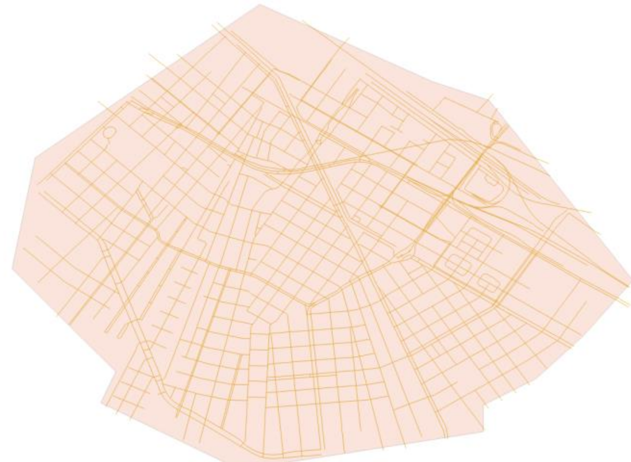


*Figure 3: Central New Orleans, USA*

Mean NLS: 0.366*

## CONCLUSION

NLS is a measure of the distance between two nodes of a road network. Closer nodes have lower NLS values. Therefore a grid system have lower NLS value would have on average shorter paths from any location on the grid to another. The reason for introduction of this algorithm is to automate the scoring of grids for city planners. Instead of measuring individual polygons or curves in grid system this algorithm provides a heuristic approach for scoring grids.

## REFERENCES

[1] Mehlhorn Kurt, Sanders Peter, Algorithms and Data Structures: The Basic Toolbox, Springer, 2008

[2] Lawhead Joel, 2015. QGIS Python Programming Cookbook, ISBN 978-1-78398-498-5

[3] Zhang Z., Jigang W. and Duan X., 2010, Practical algorithm for shortest path on transportation network. International conference on computer and information application, 978-1-4244.

[4] Borch, C., and Kornberger, M. (eds.). (2015). *Urban Commons: Rethinking the City*. New York, NY: Routledge.

[5] Catalina Turcu (2013) *Re-thinking sustainability indicators: local perspectives of urban sustainability*, Journal of Environmental Planning and Management, 56:5, 695-719

[6] Goldstein, N., Dietzel, C., Clarke, K.: Don't *Stop 'Til You Get Enough - Sensitivity Testing Of Monte Carlo Iterations For Model Calibration. In: Proceedings*, 8th International Conference on GeoComputation, University of Michigan, Eastern Michigan University (2005)

[7] Berry, B.J.L., Marble, D.F.: *Spatial analysis, a reader in statistical geography. Prentice-Hall, Englewood Cliffs (1968)*

[8] Geertman, S., Stillwell, J., and Toppen, F. (eds.). (2013). Section-2, *Planning Support Systems for Sustainable Urban Development.* Berlin: Springer.