Spatial Skyline Queries

By:
Shailendra (2016csb1059)
Harshvardhan (2016csb1120)

Paper: M. Sharifzadeh et al. The spatial skyline queries. In VLDB'06.

Problem Statement

- Given a set of data points P and a set of query points Q, with each point in P having number of derived spatial attributes decides distance between pair of data and query points.
- Spatial Skyline Queries(SSQ), retrieves those points which are not dominated by any other point in P w.r.t Q considering their derived spatial attributes.

Real Life Importance

- In Trip Planning domain, searching hotel which is not dominated by any other hotel considering w.r.t conference venue, beaches, museums, etc.
- In Crisis Management, finding mostly affected/to be affected areas around crisis locations, example fire locations.
- etc.

Algorithm for SSQ - B2S2

Algorithm $\mathbf{B^2S^2}$ (set Q)

```
01. compute the convex hull CH(Q);
02. set S(Q) = {};
03. box B = MBR(R);
04. minheap H = {(R, 0)};
05. while H is not empty
06.   remove first entry e from H;
07.   if e does not intersect with B, discard e;
08.   if e is inside CH(Q) or
09.       e is not dominated by any point in S(Q)
10.     if e is a data point p
11.       add p to S(Q);
12.       B = B ∩ MBR(SR(p, Q));
13.     else // e is an intermediate node
14.       for each child node e' of e
15.         if e' does not intersect with B, discard e';
16.         if e' is inside CH(Q) or
17.             e' is not dominated by any point in S(Q)
18.             add (e', mindist(e', CH_v(Q))) to H;
19. return S(Q);
```

Figure 5: Pseudo-code of the $\mathbf{B^2S^2}$ algorithm

Some important elements

- Convex Hull
- Box B
- RTree

Experimentation

- **Dataset:** Randomly generated using python's numpy library.
- Analysis of following things:
  - Effect of **size of query points**
  - Effect of **MBR area covered by query points**
  - Effect of **size of data points**
  - Effect of **density of query points**
  - Effect of **M value of RTree**

Query Size

- Data Size : 20000
- Query Size from [2,4,6,8,10]

Chart: Query Size Vs CH points

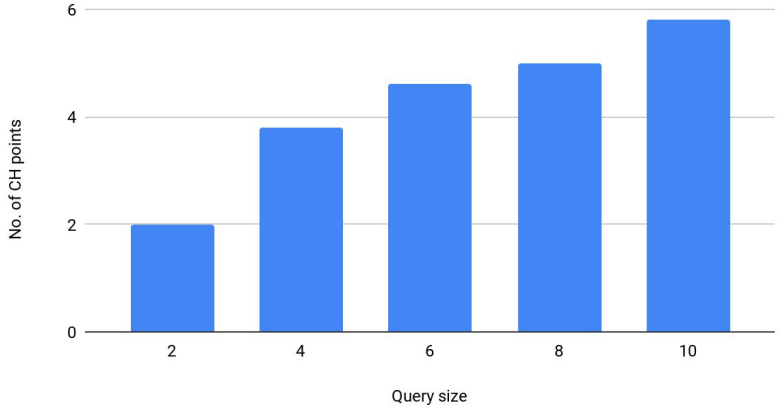


Chart: Query Size Vs Nodes accessed

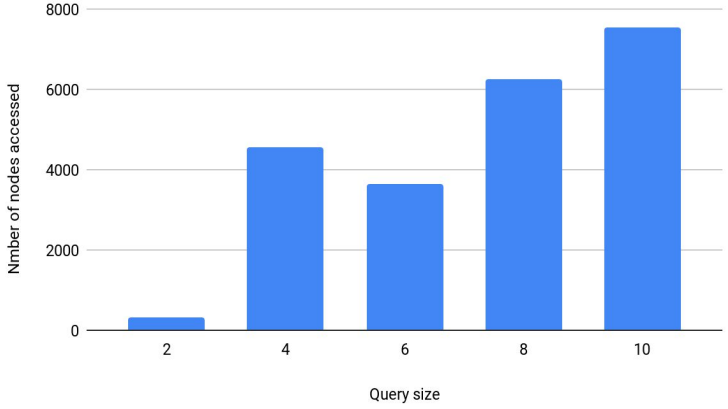


Chart: Query Size Vs Dominance Check

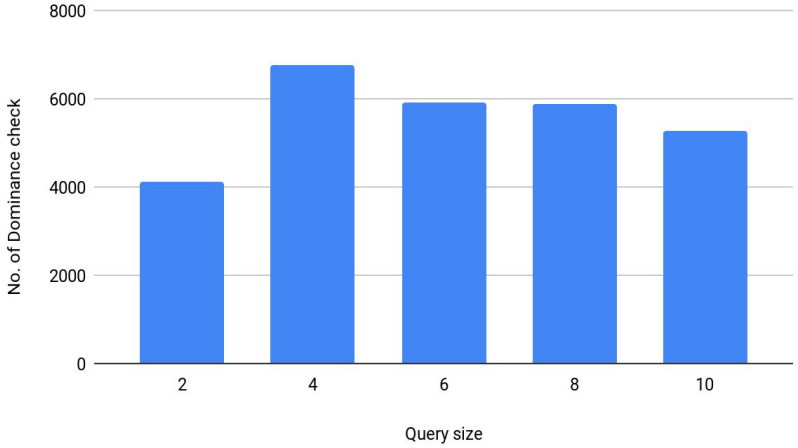
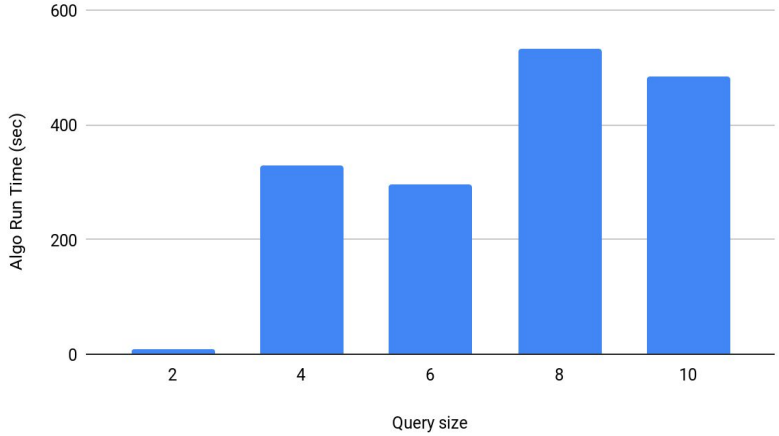


Chart: Query Size Vs Algo Run Time



Query MBR

- Data Size : 20000
- Query MBR lies between 0 to 1%.

Chart: Query MBR age VS. CH points

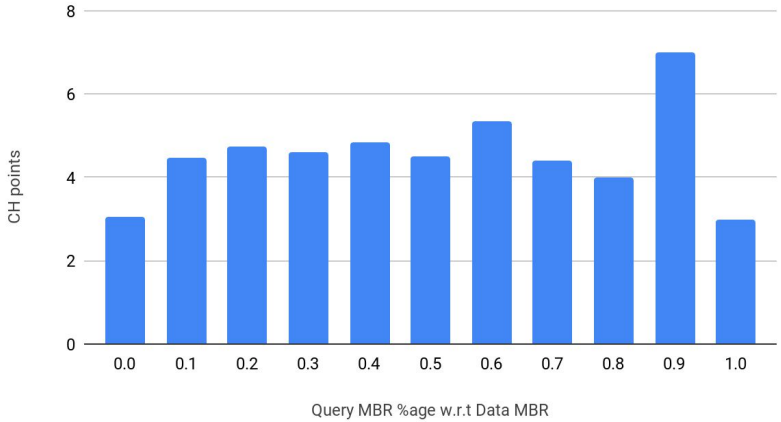


Chart: Query MBR age VS. Nodes Accessed

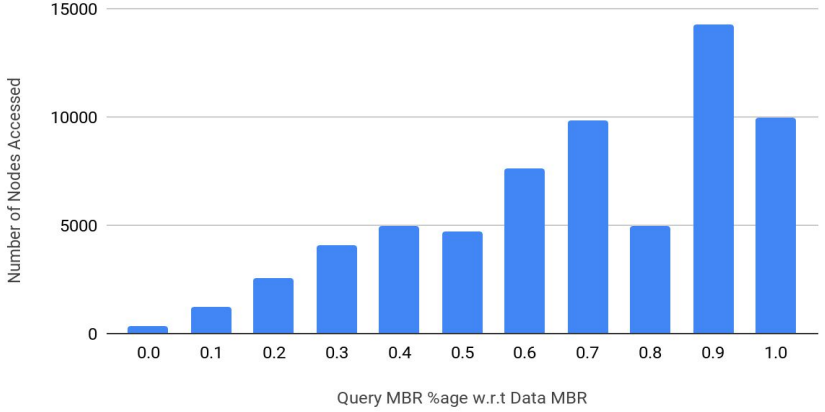


Chart: Query MBR age VS. Dom. Check

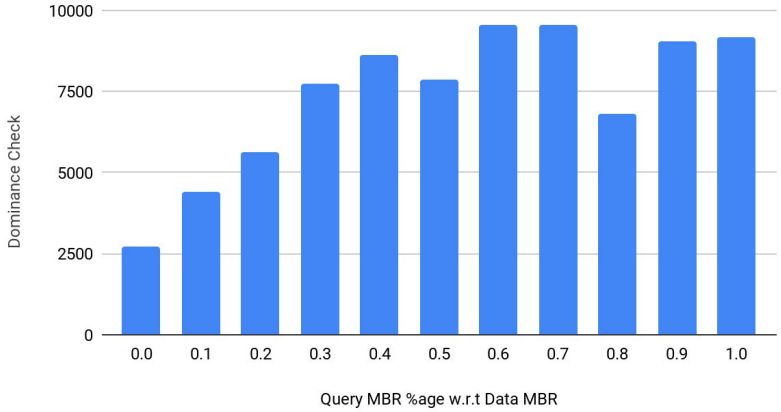
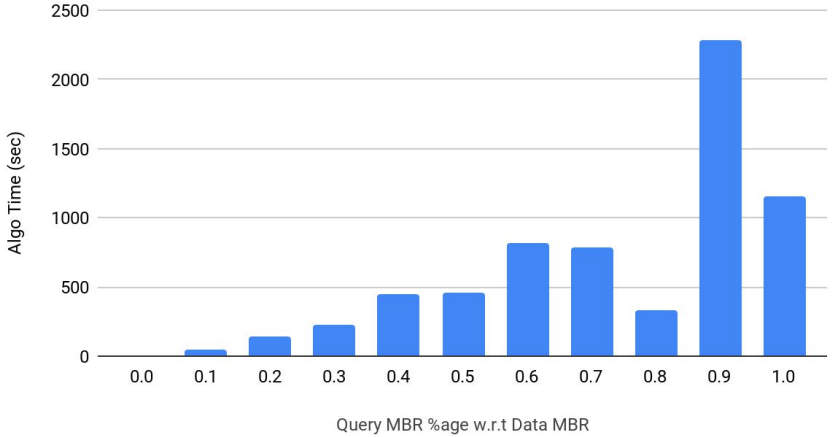


Chart: Query MBR age VS. Algo Time



Data Size

- Data Size varies from [10000, 20000, 40000, 60000, 80000]
- Query Size = 2

Chart: Data Size Vs. Ratio of nodes accessed

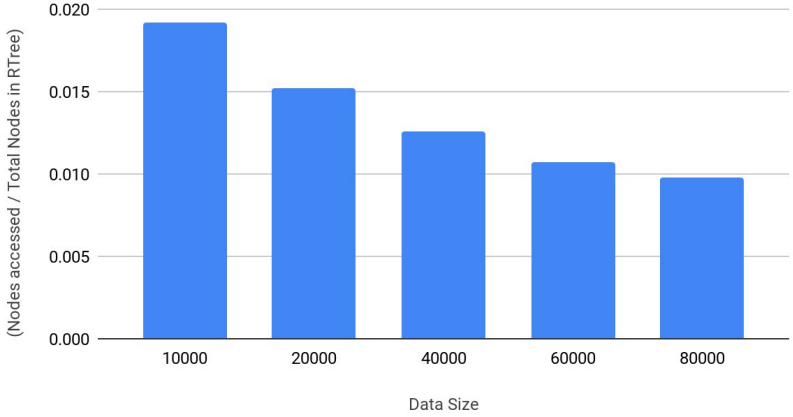


Chart: Data Size Vs. Nodes accessed

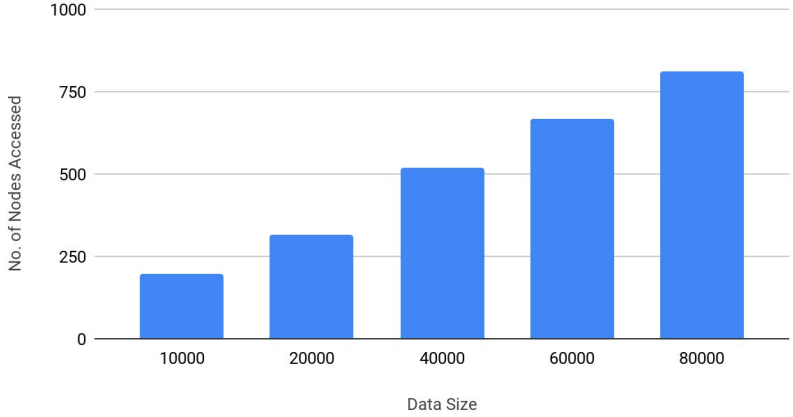


Chart: Data Size Vs. Dom. Check

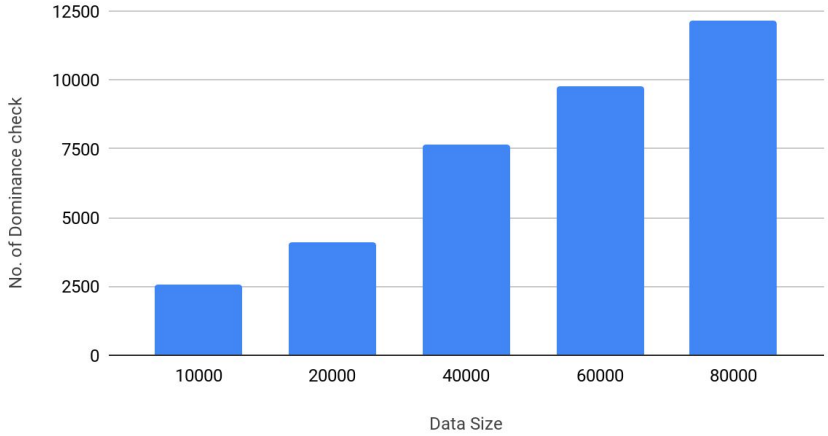
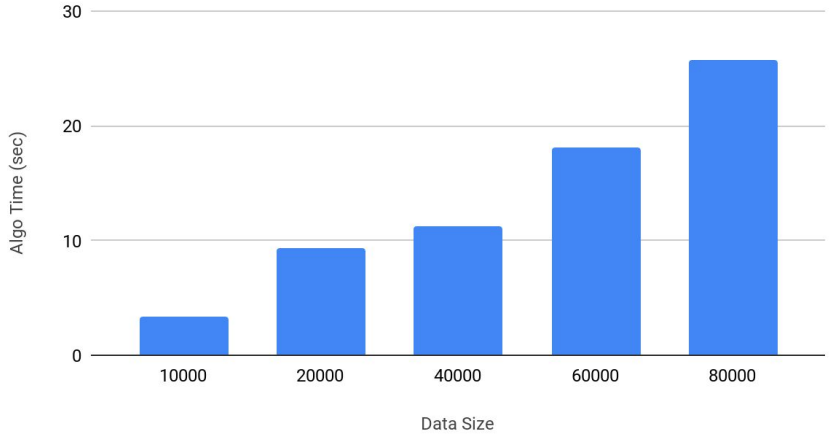


Chart: Data Size Vs. Algo Time



Data density

- Query Size = 2

Chart: Data density Vs. ratio of nodes accessed

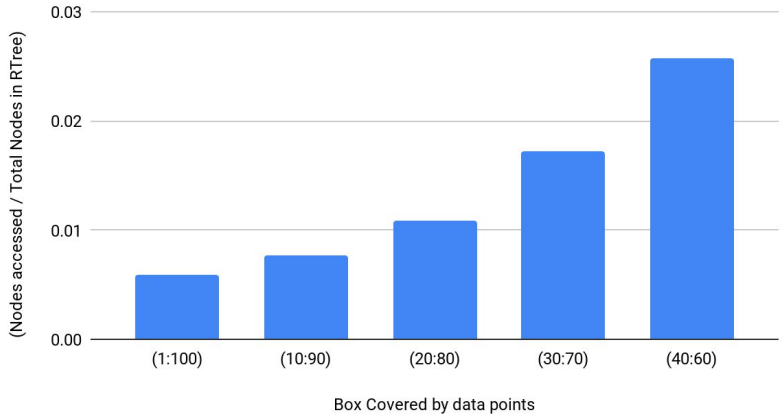


Chart: Data density Vs. Nodes Accessed

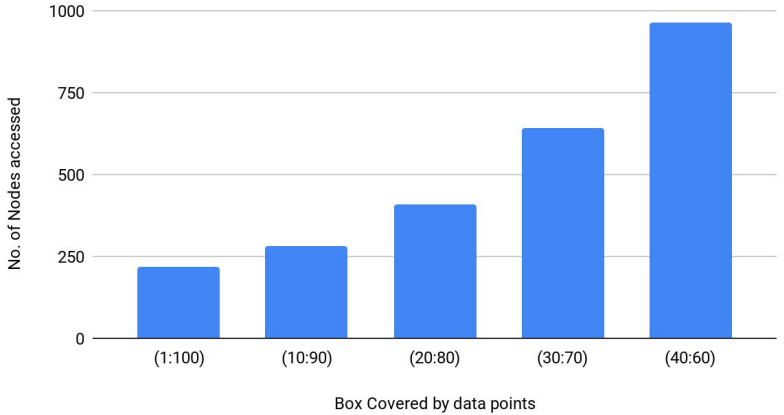


Chart: Data density Vs. Dom. Check

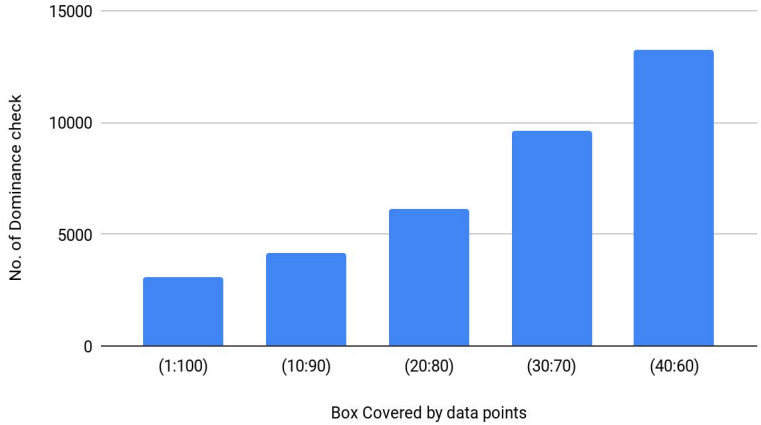
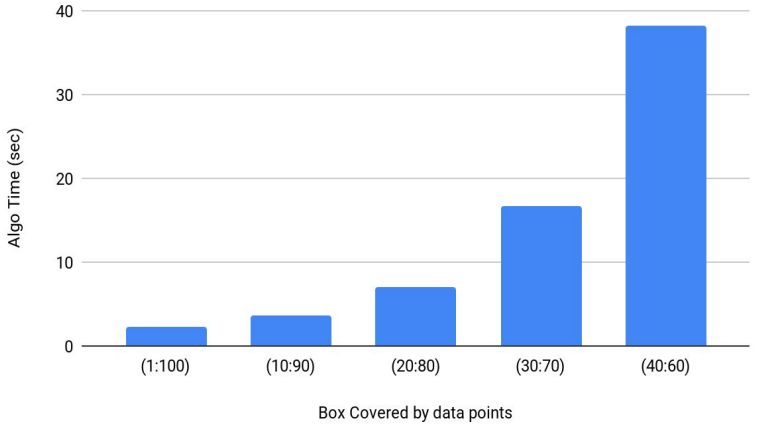


Chart: Data density Vs. Algo time



M-Value

- Data Size varies from [10000, 20000, 40000, 60000, 80000]
- Query Size = 2

Chart: M-Value Vs. ratio of nodes accessed

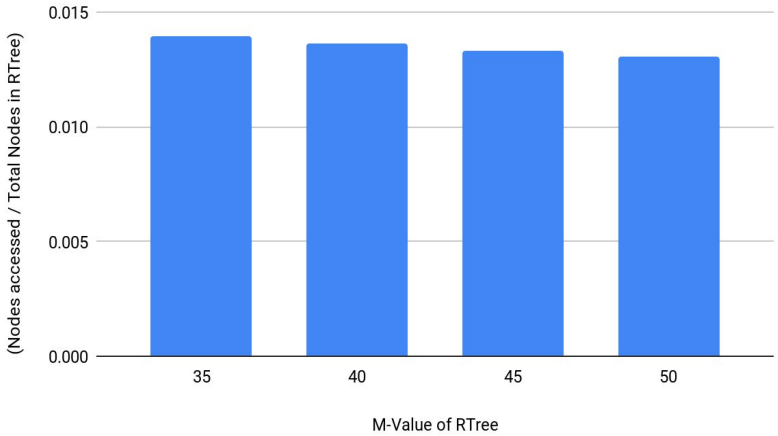


Chart: M-Value Vs. Nodes accessed

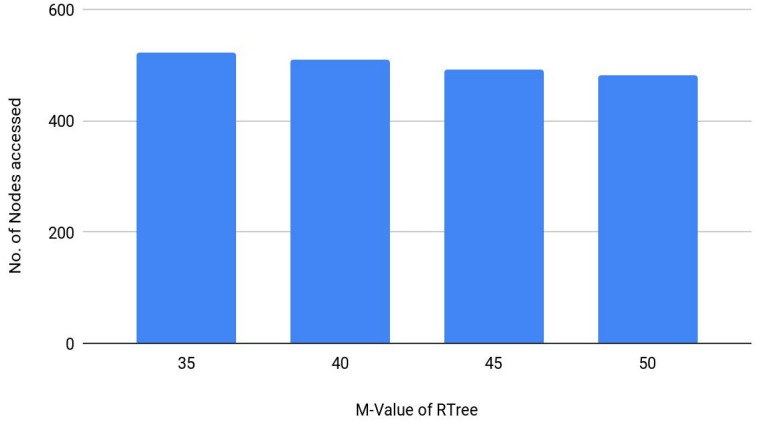


Chart: M-Value Vs. Dom. Check

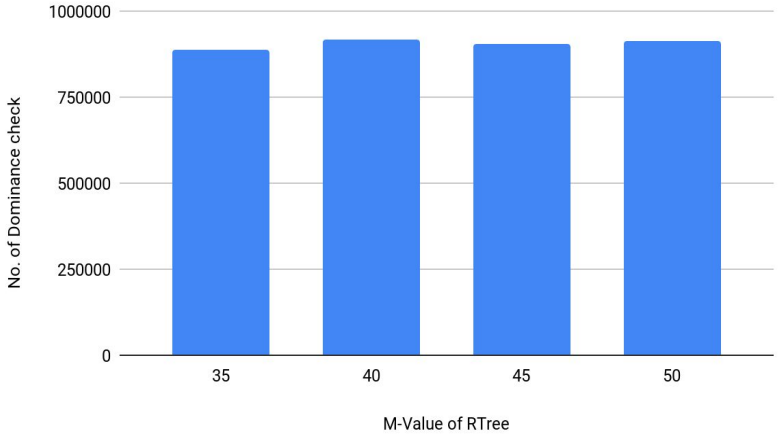
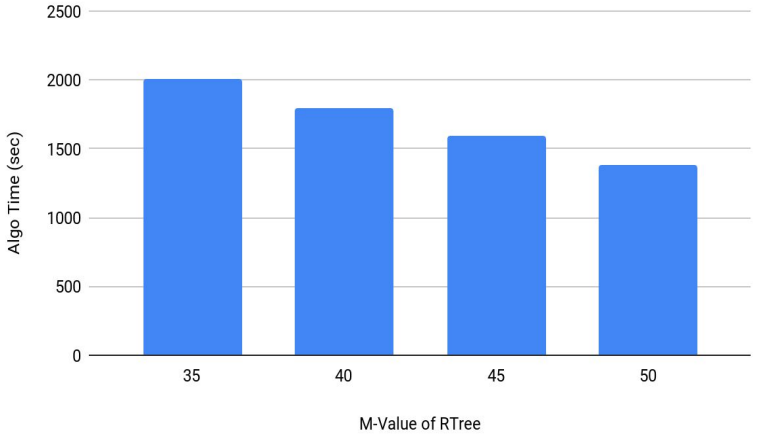


Chart: M-Value Vs. Algo time



Thank you