

Localization of Sensors

Gaurav Sharma

Rahul Ajmera

Group 5

Department of Computer Science and Engineering

Indian Institute of Technology , Kanpur

8th April 2013

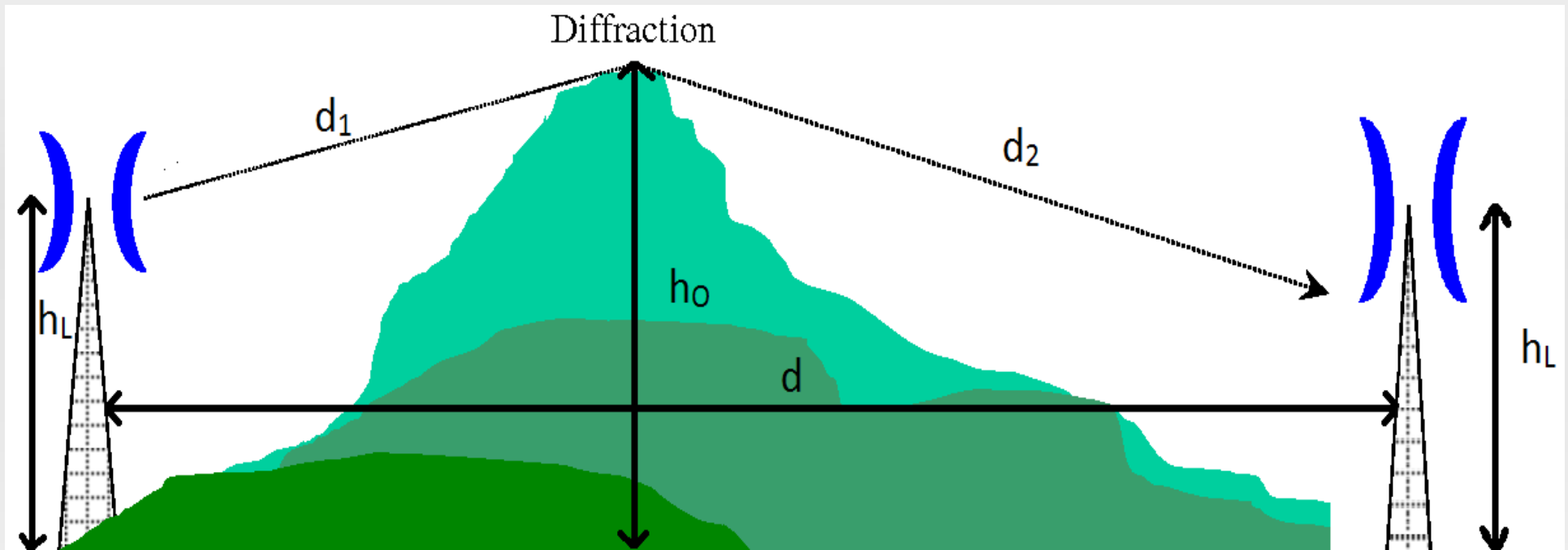
Problem Statement

Given a sensor network of N sensors at locations $S = \{S_1, S_2, \dots, S_N\}$. So, mathematically the localization problem can be formulated as follows: given a multihop network, represented by a graph $G = (V, E)$, and a set of beacon nodes B , their positions $\{x_b, y_b\}$ for all $b \in B$, we want to find the position $\{x_u, y_u\}$ for all unknown nodes $u \in U$.

Radio Propagation Model

- ITU Terrain Model.
- Applicable for all terrains.
- Valid for all distances and frequency.
- Handles diffraction due to obstacles.
- Developed on the basis of diffraction theory and first fresnel zone.
- Obstacles in the first fresnel zone will create signals, with a phase shift of 0 to 180 degrees at the receiver.

Radio Propagation Model



Formula: $A=10-20C_N$ $C_N=h/F_1$ $h=h_L-h_o$

$F_1=17.3 \sqrt{d_1 d_2 / f d}$ where, f is frequency and A is additional loss due to diffraction.

These losses are summed with Friis transmission equation loss to get overall signal strength.

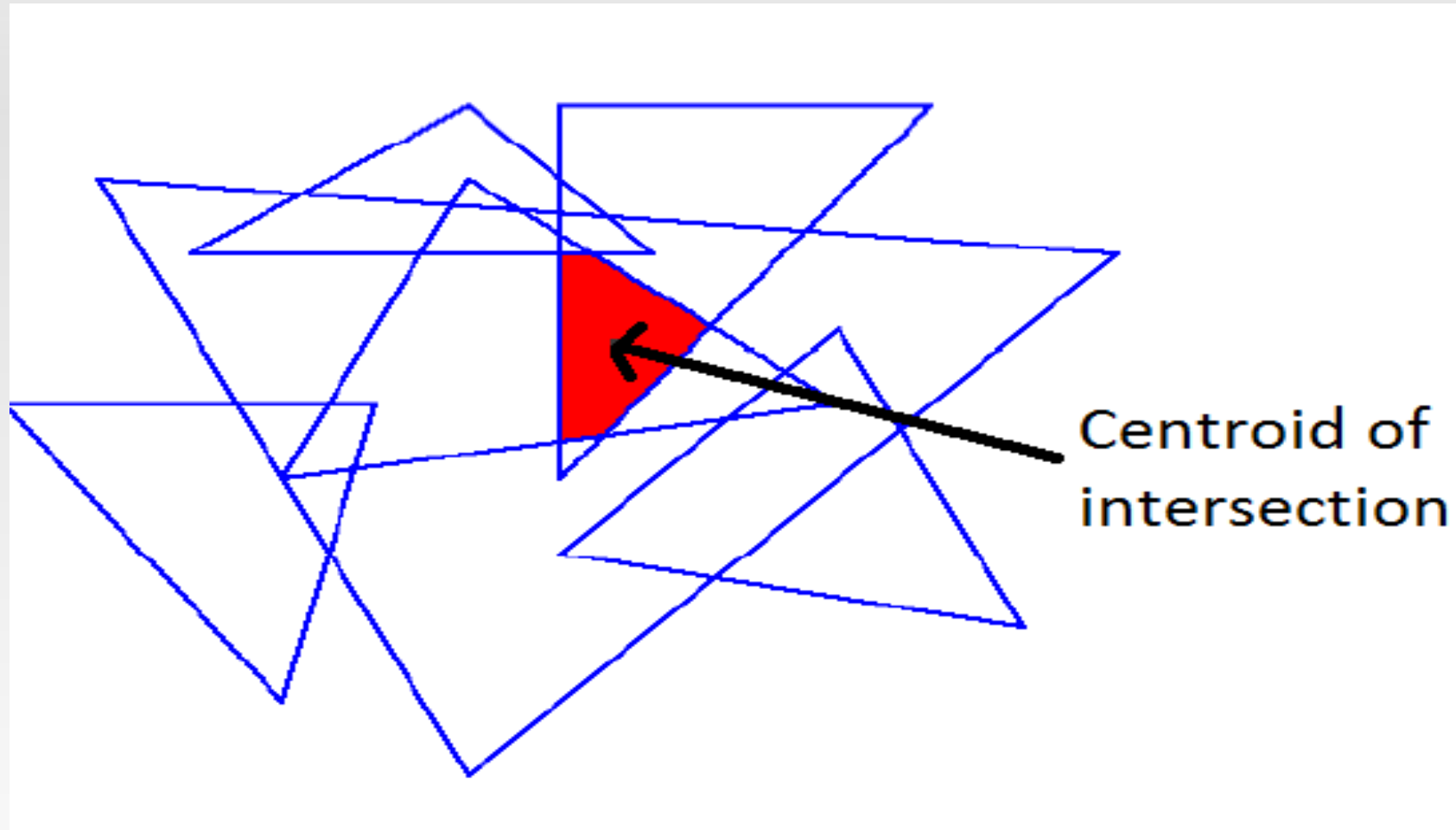
Implemented Algorithms

- APIT
- Diffusion
- Moving Sensor
- Moving Target
- Gradient
- Simulated Annealing
- Centroid

APIT

- Node are assumed to be listening to beacons, and storing distances from each one.
- Based on the signal strength, it determines the triangles in which the node lies.
- Centroid of the intersection area of all such triangles is estimated to be the localized coordinates.

APIT



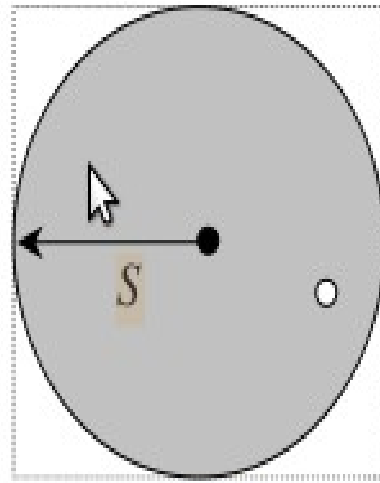
Diffusion

- Location of a node is centroid of its one hop neighbors.
- Iteratively allocate location of a node to be average of it's neighbors.
- Terminate when steady state is reached, or after a fixed number of loops.

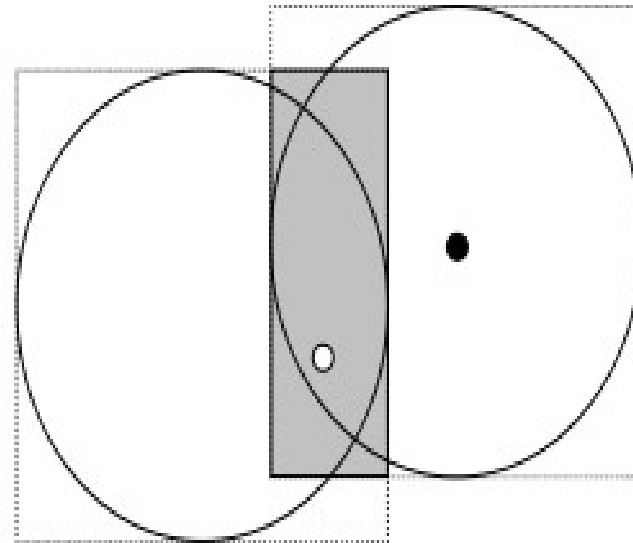
Moving Sensor

- A beacon is randomly moved in the grid and it broadcasts its coordinates to the neighbors at all time.
- Every time a node senses the beacon, it generates a new quadratic constraint that it uses to further reduce the uncertainty in its position.

Moving Sensor



(a)



(b)

- (a) Node sensing beacon for the first time and constraining itself in the shaded region.
(b) Beacon moves to another location, node senses again and reduces its region.

Moving Target

- All the nodes are moved in the grid to detect beacons.
- When it detects a beacon it introduce a bounding box constraint on its position.
- If the node does not encounter any beacon than it bounding box remains the whole grid.
- If a beacon at (x', y') after travelling (x, y) than a square bound of side twice the node range is imposed at $(x' - x, y' - y)$.

Moving Target

Gradient

- All beacons initiates a gradient by sending its neighbors a message with its location and a count set to one.
- Each recipient remembers the value of the count and forwards the message to its one hop neighbors with the count incremented by one.
- Hence a wave of messages propagates outwards from the beacon, while each sensor maintains the minimum counter value received from all beacons.

Gradient

- Hop size of each beacon is calculated by using the number of hops and actual distances from other beacons.
- A error function of two variables x, y for each node is minimized, which is defined by:

Error = $|\Sigma(x-x_i)^2 + \Sigma(y-y_i)^2 - \Sigma d_i^2|$ summed over all beacon nodes(i).

Centroid

- Triangulation of beacons is used, to determine triangle containing the nodes, based on the signal strength.
- Contribution to localization $\propto 1/\text{Area of Triangle}$
- Fast Algorithm.

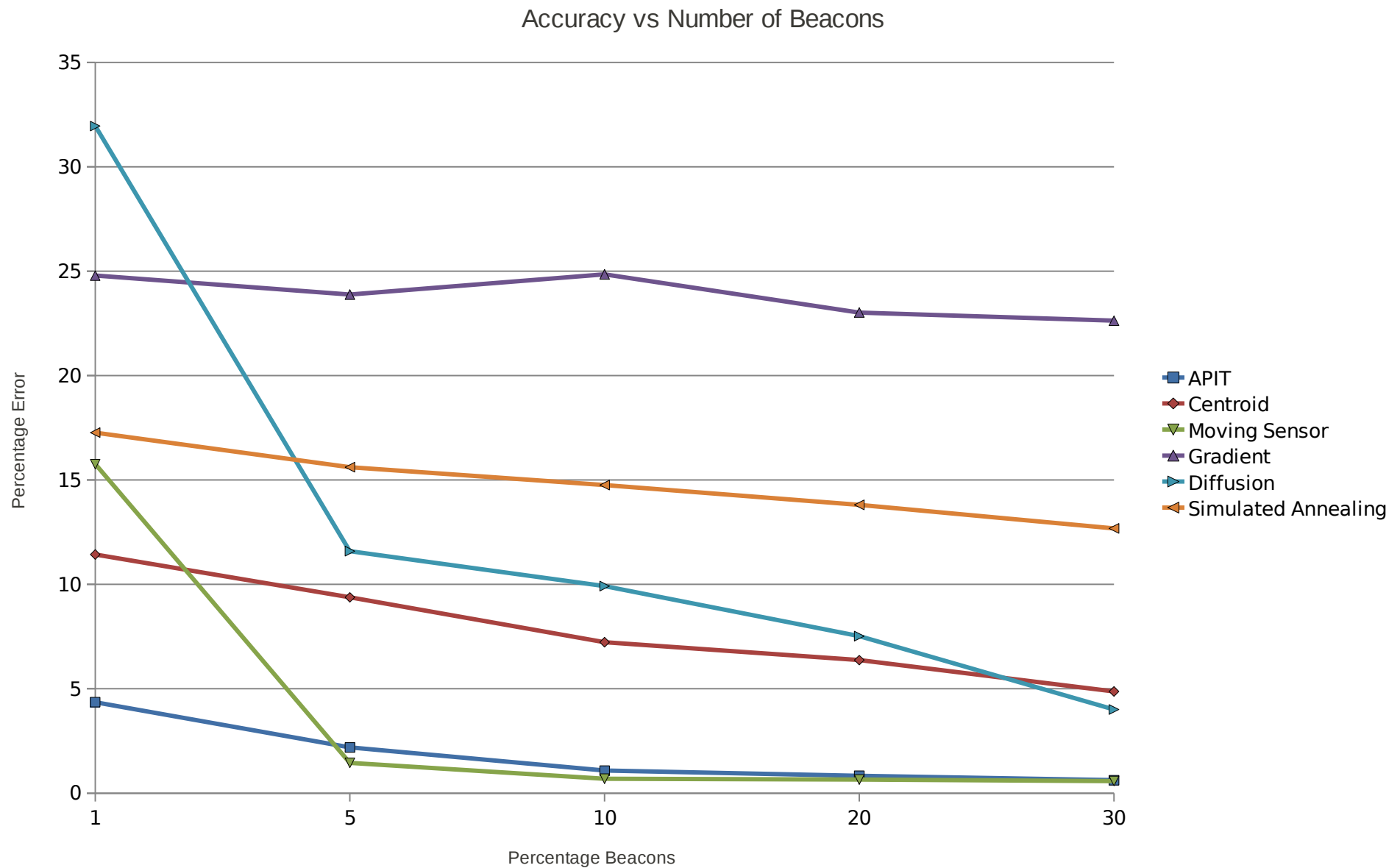
Simulated Annealing

- System starts from random state and comes to equilibrium.
- A small perturbation (random displacement) is given to nodes and Cost Function is calculated for that state.
- If cost decreases than new state is accepted but if cost increases than the state is accepted with probability, so that it does not get stuck at local minima.

Simulated Annealing

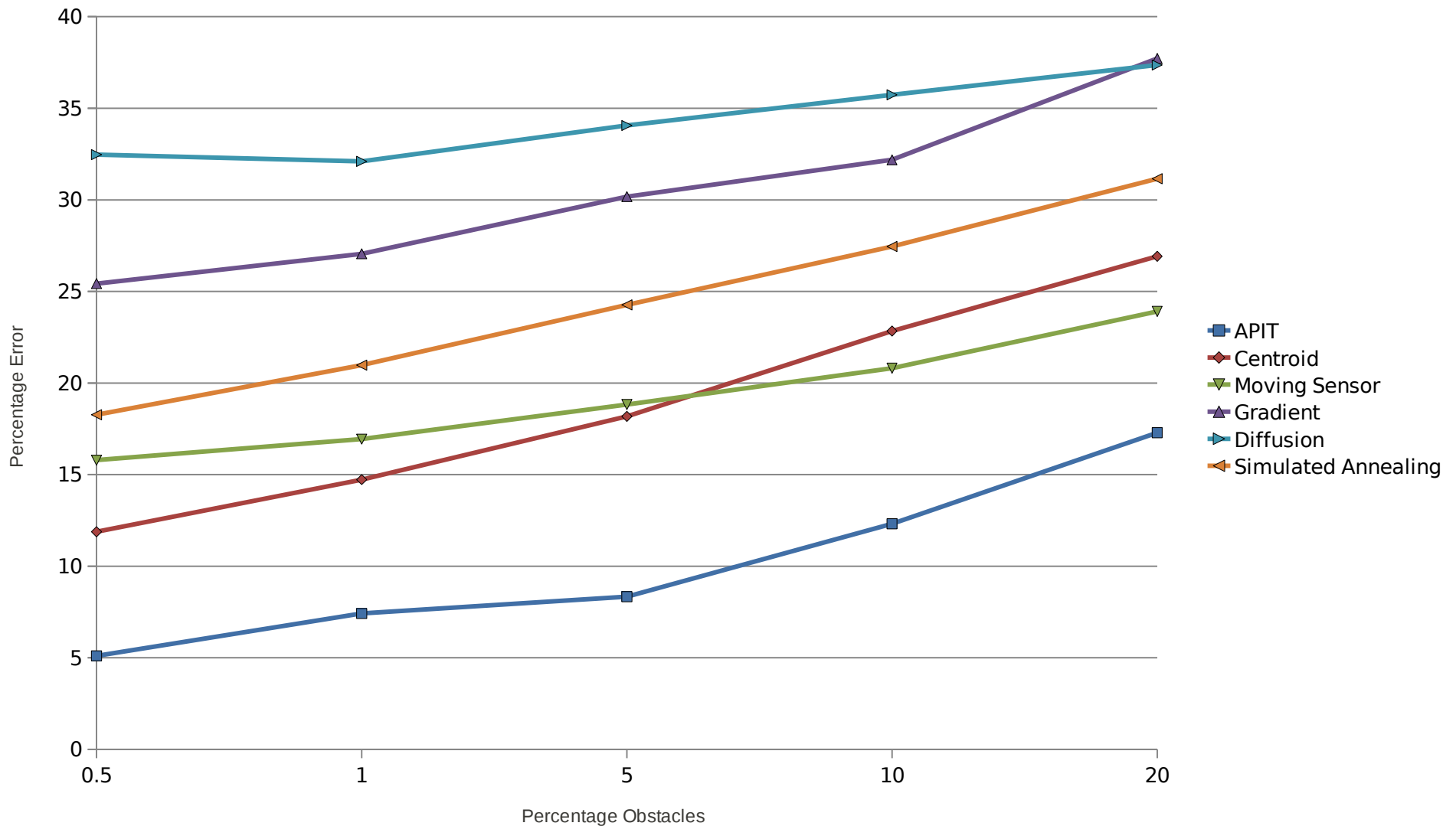
- Cost Function
- Probability Function

Results



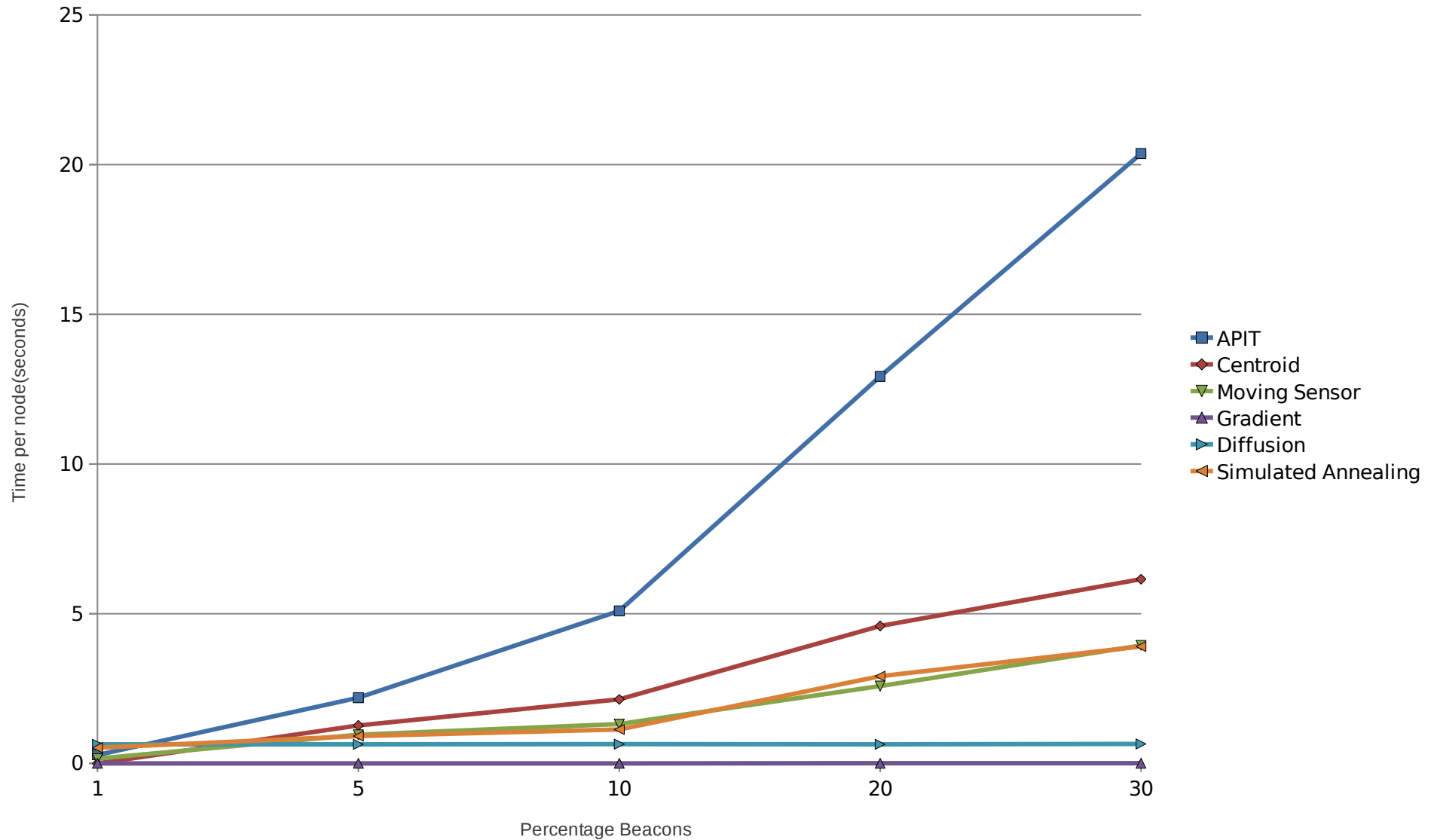
Results

Accuracy vs Number of Obstacles

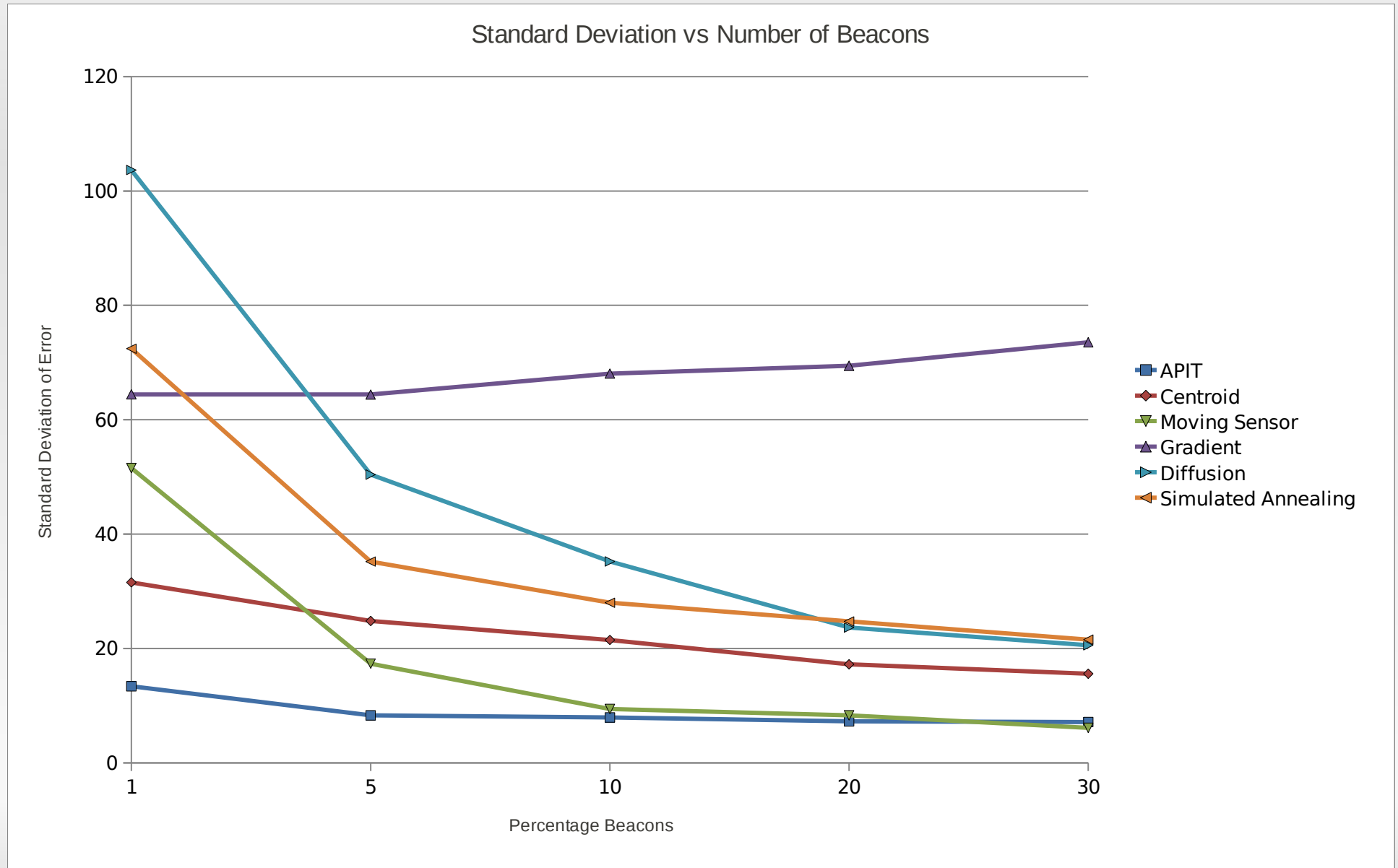


Results

Time vs Number of Beacons



Results



Conclusion

- Based on the qualities required by the system, a cost function of the system can be obtained in terms of **accuracy, tolerance due to obstacles, time required for computation and variation in error.**
- Based on the obtained cost function an appropriate algorithm could be used which **minimizes** the cost.