

Wireless Access Point Configuration Problem

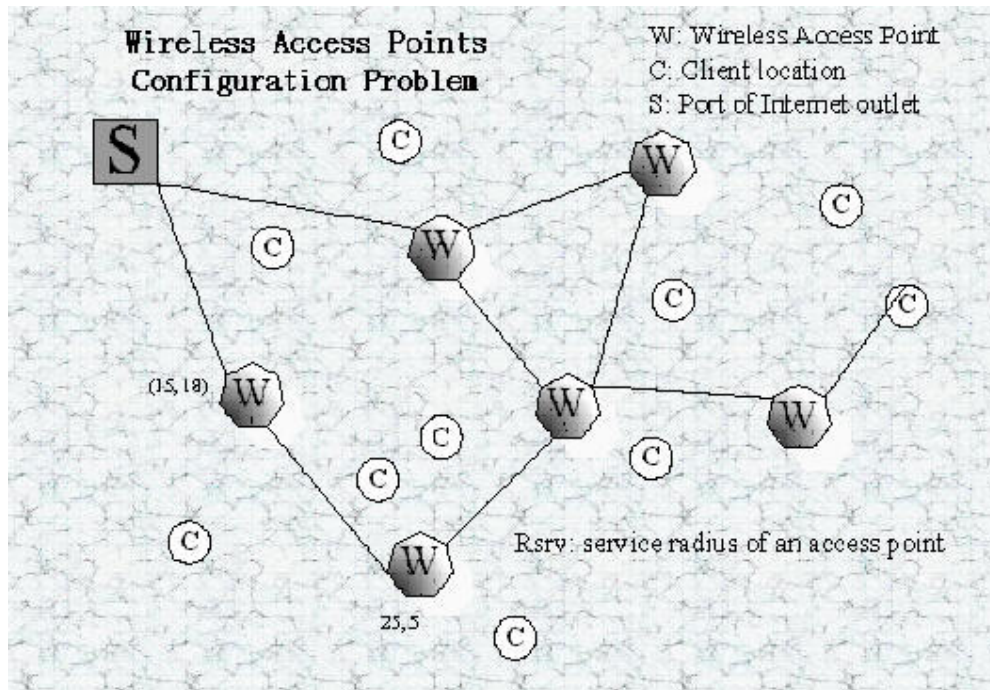
A benchmark problem for open-ended topology and parameter search

Jianjun Hu (hujianju@msu.edu)

Wireless Access Point Configuration problem is proposed as a benchmark problem for testing our Genetic Programming algorithm in simultaneous open-ended structure and parameter evolution (search)

I. Introduction to Wireless Access Point Configuration problem:

A small city is planning to provide wireless Internet service to its citizens, who are located around the city map below (labeled as circled C). A bunch of wireless access points then need to be placed at several places to cover all the clients as each access point has limited service radius. To reduce the cost, a design solution with minimal number of access points and minimum length of the wires connecting those access points is regarded as optimal. **So the question is, how to decide the number of necessary access points and which place should we put.** To make it harder, we can ask to maximize the average signal strength at all clients, or add some constraints on the location of the access points. As this problem (not the [simplified version](#)) requires simultaneous topology and parameter search, many issues exist.



Where is the structure to be evolved?

Every wireless access point must be either directly linked to the source radio station or linked to another access point which has been linked to the radio station (directly or indirectly) by wires whose cost is proportionate to its length. All the connected graph is the structure to be evolved

What are the parameters to be evolved?

The locations of access points are the parameters to be evolved. Their range can be constrained in a range, e.g. $(x,y)|x,y \in [0, 1000]$

II. Problem statement:

Given a set of clients and the position of the source internet connection outlet, find a best configuration of the access points such that:

1. Each client is within the signal transmission radius R_{sig} of at least one access point. (fully covered).
2. using the least number of access points. Each access point has some cost.

$f_{penalty} = -C_{rs} * N_{rs}$ where C_{rs} is the cost of each access point, N_{rs} is the number of access point.

3. Minimize the cost of the wires used to connect the access points. The cost is proportionate to the length of the distance between two connected access points. [To make things easy, we only need to evolve the location points and use Minimum Spanning Tree algorithm to find the topology or connections.](#) To test the GP synthesis engine, we ask the GP evolve the Minimum Spanning Tree topology.
4. Maximize the average signal intensity of all client sites. (optional)

$$S_{Intensity} = K \sum_{\substack{i=1 \\ |p_c - p_{r_i}| \leq R_{sig}}}^N \frac{1}{p_c - p_{r_i}}$$

5. The coordinates of sources, access points and clients are integers.

III. Varying the difficulty of this problem:

- 1) Increase the number of client sites.
- 2) Constrain the possible position area of access points or add other constraints
- 3) Allow multiple radio station sources.
- 4) Allow several existing access points.

- 5) Include multiple objective
- 6) Assume the cost of access points with different construction costs
- 7) Similar problems including water system, etc.

IV. Network encoding in Genetic Programming

Edge-encoding (Sean Luke).
 Cellular Encoding/node encoding (Gruau)
 Hybrid encoding (koza, this package)

V. Open Question:

Is this problem a NP problem?
 How is it related to TSP? Can it be a TSP in GP?
 Are there some existing similar problems in graph theory literature?

Fitness Definition

```
//number of covered clients
double nCoveredClients = g->pClientGraph->getNumVertices()-ncoveredClients;
//penalty of the total length of edges
double wireCost = g->getSumWeight();
if(wireCost==0) wireCost=1.0;
//cost of building each access point
double stationCost = g->getNumVertices();
double coverFit = 20.0*(double)nCoveredClients/g->pClientGraph->getNumVertices();
double wireFit = 1000.0/(1000.0+wireCost);
double stationFit = 10.0/(10+stationCost);
//cout<<coverFit<<" "<<wireFit<<" "<<stationFit<<endl;
//total fitness value
finalfitness= coverFit+wireFit+stationFit;

//penalty if not connected
if(!g->isConnected()) finalfitness=finalfitness/2 ;
ind->hits = 0;
ind->r_fitness = 0;
ind->s_fitness = nCoveredClients;//The objective raw fitness of an individual
ind->a_fitness = finalfitness; //used for selection process
```