Solving RND Problems by using PBIL and inexpensive High-Performace Clusters

IBERGRID'08 – Oporto

César Gómez-Martín – Miguel A. Vega-Rodríguez
Juan A. Gómez-Pulido – Juan M. Sánchez-Pérez

{cesar, mavega, jangomez, sanperez}@unex.es

Outline

- Introduction
- Radio Network Design Problem
- Population-Based Incremental Learning
- OpenMosix
- Results
- Conclusions
- Future Work

Introduction

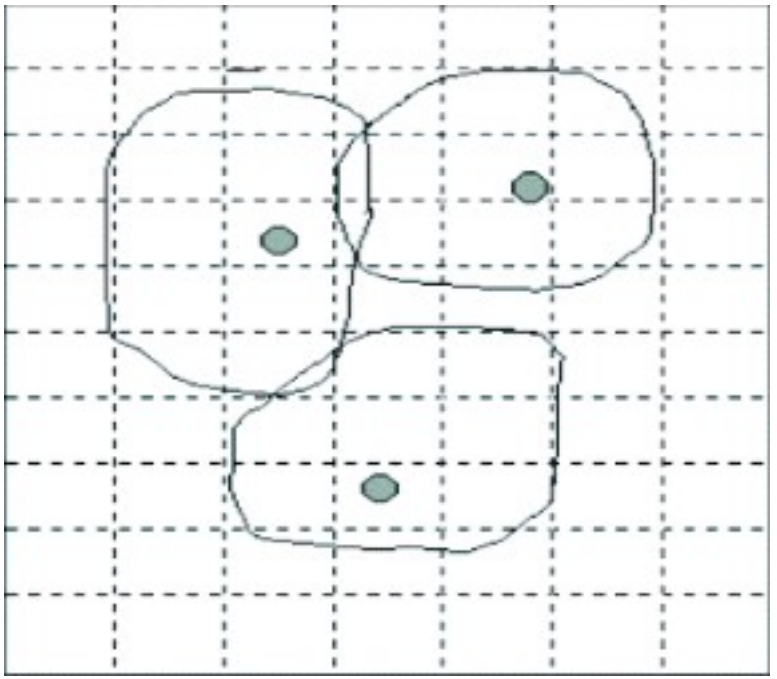
- Evolutionary Algorythms are aimed to optimize functions
- Genetic Algorythms are part of the EAs family
  - Start off a population of candidate solutions (individuals)
  - Meant to evolve to better solutions
- Radio Network Design Problem can be accomplished using GAs

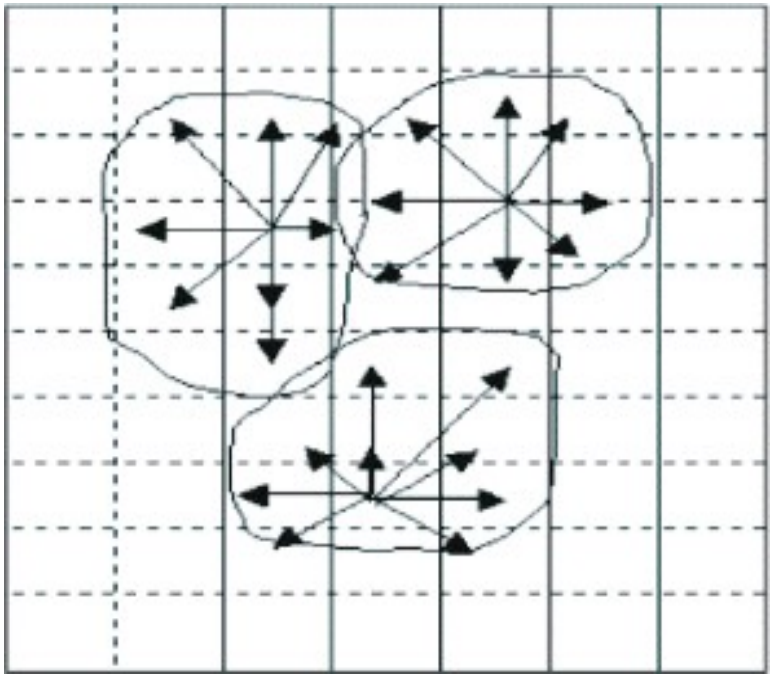
Radio Network Design Problem

- Efficiently design radio networks to cover a geographical area with a set of transmitters
- The goal is to use as less transmitters as possible to cover the maximum area
- RND is an NP-Hard problem
  - Sometimes we need to cover the whole map, but sometimes it is not neccessary

Radio Network Design Problem

- Terrain must be discretized to form a grid of cells





Radio Network Design Problem

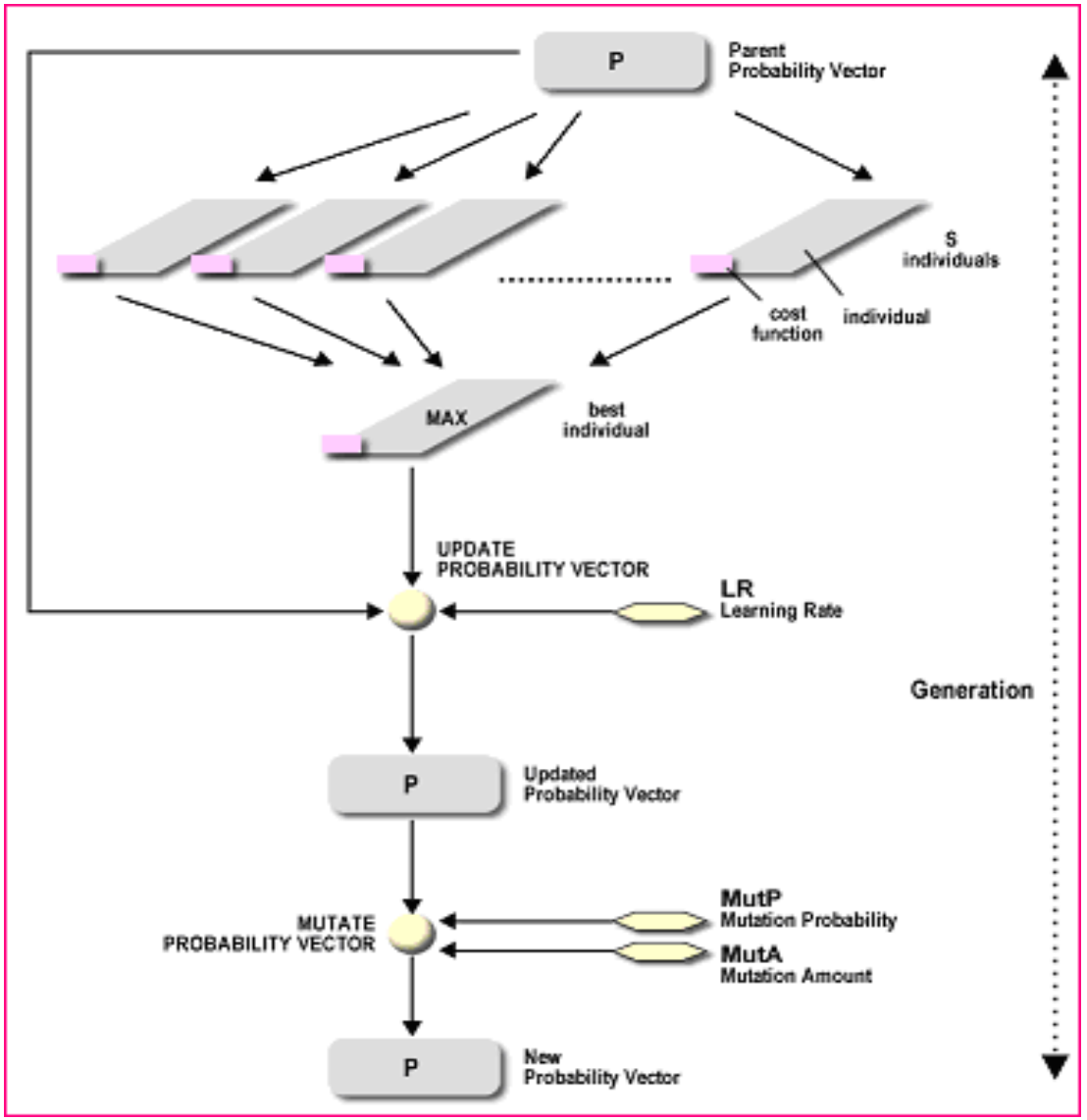
- Fitness function with 2 main objectives
  - Minimize the number of transmitters
  - Maximize the number of covered points
- Every individual of the population will be represented as a grid with all the terrain locations containing:
  - A number to indicate the coverage level
  - A token if a transmitter is located in that position

PBIL

- PBIL (Population Based Incremental Learning)
- PBIL is a method that combines GAs with competitive learning for function optimization
- PBIL generates a probability vector in every generation which will be the prototype for the function space

PBIL

- The probability vector is initialized
- Evolves depending on the parameters
  - Number of individuals
  - Learning Rate
  - Mutation Probability
- Vector represents the best frequency planning for the RND Problem



OpenMosix

- Package that turns networked computers running GNU/Linux into a cluster
  - Automatic load balance
  - No code changes
  - Node autodiscovery
  - Fail-over daemon
  - No extra packages
  - Simple to install and configure

Results

Generations	Fitness
10	61.056587
100	61.670708
1000	66.010689
10000	**75.832756**

Table 1. Different number of generations (Individuals=100, Learning-rate=0.2, Mutation-probability=0.5, Mutation-shift=0.5)

Learning-rate	Fitness
0.1	73.182152
0.2	**76.479187**
0.3	66.010689
0.4	57.633095
0.5	56.200607

Table 2. Different learning-rates (Generations=1000, Individuals=100, Mutation-probability=0.5, Mutation-shift=0.5)

Results

Individuals	Fitness
10	60.058262
100	65.591293
1000	66.583519
10000	**68.616165**

Table 3. Different number of individuals (Generations=1000, Learning-rate=0.2, Mutation-probability=0.5, Mutation-shift=0.5)

Mutation Probability	Fitness
0	71.682999
0.1	**74.468987**
0.2	72.357515
0.3	72.282376
0.4	72.155988
0.5	72.147692

Table 4. Mutation probabilities (Generations=1000, Individuals=100, Learning-rate=0.2, Mutation-shift=0.5)

Results

Mutation Shift	Fitness
0.1	**79.304161**
0.2	76.088631
0.3	73.495293
0.4	58.784767
0.5	36.514057

Table 5. Mutation shifts (Generations=1000, Individuals=100, Learning-rate=0.2, Mutation-probability=0.1)

1 Intel(R) Xeon(TM) CPU 3.06GHz (2 CPUs)	openMosix cluster	Speedup	Efficiency
4h59m49s	48m39s	**6.163**	**0.88**

Table 6. Speedup-Efficiency (Simulations=10, Generations=10000, Individuals=100)

Conclusions

- PBIL is a good method for solving RND Problems
  - Mutation feature improves results
  - All the parameters should be slightly and carefully adjusted to get better results
- OpenMosix gives a very interesting speedup
  - No need to modify the source code
  - No need to balance the load manually
  - Node autodiscovery
  - Fail-over daemon

Future Work

- Improve our PBIL implementation
  - Using not only the best individuals but also the worst individuals to update the probability vector
  - Replacement of some individuals instead of replacing the whole population
- Grid version of the program

Thank you!

Questions?