Genetic Algorithm-based Resource Allocation for Virtualized Wireless Networks

Kory Teague
School of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA
Email: koryt@vt.edu

Dr. Mohammad Abdel-Rahman
School of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA
Email: mo7ammad@vt.edu

Allen MacKenzie
School of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA
Email: mackenab@vt.edu

***Abstract*—The abstract goes here once written.**

I. Introduction

In this section we include interesting and necessary background information. Detail virtualized wireless networks, the problem being investigated, and setup the "story" behind the paper. Motivations and "why" in a few paragraphs.

II. System Model

In this section, we detail the system model as utilized for the paper. Include use and references to the log-normal demand model, demand points, resource generation, etc.

Establish the assumptions for the system: a homogenous set of base stations (S), a log-normal demand model which generalize a set of demand points (M), demand is satisfied if it lies within coverage (no path-loss, fading, interference, etc.), potential of slicing resources (base stations),

Establish the log-normal demand model as the demand model being tested against. Include how it is generated in the reference material:

* Formulate an approximate Gaussian field through the sum of 'M' sinusoidal-product fields (the product of two sinusoids, one in y, one in x), with each component sinusoid having a random frequency (from zero to omega max) and phase shift (from zero to 2pi)

* With large enough 'M', this should generate a spatially-correlated Gaussian field with autocorrelation dependent on omega max.

* Normalize this sum to be a standardized 2D-Gaussian field.

* Convert the Gaussian field to Log-Normal by applying location and scale and taking the log of the result.

The reference material for generating this demand model performs this by generating a mesh or grid of points, and allocating the above methodology to the mesh. I maintain the field as a continuous function instead, to allow for points not residing at regular intervals (for PPP later).

When discrete points are needed - such as set M for the optimization problem - the continuous model is sampled via a non-stationary PPP. Each point in the nsPPP is generated by creating a PPP of iid points, and randomly keeping each point with probability equivalent to the log-normal demand field at that point's location.

III. Solution Approach

In this section, we detail the two methods for trying to optimally allocate demand to available resources. Each subsection will include a separate implementation: the optimization problem as formulated by Mohammad and implemented in CPLEX, and the genetic algorithm approximation as developed by Kory and implemented in MATLAB.

A. Optimization Problem

In this subsection, we detail the optimization problem from base formulation (the stochastic two-stage problem) and reformulation to its deterministic equivalent program. Include model-affecting caveats which are important to the implementation of the DEP problem (e.g. the continuous model is discretized as demand points to create set M). A large portion of this subsection is Mohammad's responsibility, namely the initial and DEP formulation, discussion, and references, but already partially provided through previous paper documentation.

Introduce the stochastic optimization problem, building off the previously defined system model, with support

Reformulate the problem via introducing the DEP of the stochastic optimization problem with slicing.

Expound on the program by introducing sampling methodology used to create the set of scenarios. The full problem space is infinite, as it lies in the continuous domain; by sufficiently sampling the problem space, the DEP will arrive to a sufficiently close solution.

B. Genetic Algorithm

In this subsection, we detail the genetic algorithm approximation problem. No need to go into too much depth for the genetic algorithm itself, but include any information necessary to detail how the genetic algorithm works. That is, discuss chromosome formulation, crossover and mutation information, initial condition, special schema (e.g. elitism, uniqueness), and (perhaps?) pseudo-code. Further, detail any other differences of the overall system model for the genetic algorithm compared to the DEP problem.

Introduce the approximation, accepting suboptimal results for an improvement in operation time. Describe the genetic algorithm; avoid going into too much detail as GA are sufficiently well known, but go into implementation and problem-specific considerations. Describe chromosome construction (each bit a base station in the considered area), initialization of the GA (random binary string), selection for the next generation (fitness-based roulette), operator implementation (single-point crossover and bit string mutation), uniqueness, elitism, and termination/halting criteria.

Introduce the GA fitness function, formulated to correspond to the stochastic optimization problem and its DEP. Due to correspondence to optimization problem formulation, include important differences, if any. Perhaps how, in calculating fitness, this uses a grid-/mesh-based grid to emulate the continuous model instead of discrete points; to calculate fitness, it sums the pixels (instead of demand points) within BS coverage. Also need to introduce the considerations for overdraft - might want to look into a different term - and overcapacity and why it can happen for the algorithm.

IV. Simulation Results

In this section, we detail the simulation procedures. This includes data generation, assumptions, differing models (small and larger scale data sets?), resulting data, how the data is evaluated. Include the resulting data and potential takeaways from the data. If solutions are as expected, expand and expound upon it. If not, then hypothesize why.

Most of this section cannot be worked on until the appropriate data is generated via simulations. Need optimization and approximation results to compare/contrast against each other. Approximation data currently seems good, but need optimization results to ensure satisfactory results, especially over a larger data set.

V. Conclusion

The conclusion goes here. Look back on results and reiterate main takeaways. Include possible avenues for further research and expansion to the model. Perhaps power control, more nuanced demand-resource allocation (other than basic, simple voronoi), path-loss addition, applying slicing to the genetic algorithm more directly, integrating over the regions in the voronoi GA instead of summing (could be faster; math could be interesting in this or further papers), etc.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999. Sample bib from template.