A Genetic Algorithm Approach for Approximating Resource Allocation in Virtualized Wireless Networks with Log-Normal Distributed Traffic Demand

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

As it stands, the problem has been simplified such that the story may not include VWN in its implementation. Only a single network is being created from a larger pool of resources, and no resource slicing occurs. However, this might be utilized as a foundation for further VWN research by expanding the methods with slicing and multiple derived networks.

The rest of this paper is organized as follows. In section II, we detail and define the system model assumed for our resource allocation methods. In section III, we consider our resource selection and demand allocation approaches. In section IV, we simulate the described approaches and evaluate their performance. Finally, in section V, we discuss our conclusions and possible future extensions.

II. System Model

We consider a geographical area of width X and length Y that contains a set $\mathcal{S} = \{1, 2, \ldots, S\}$ of base stations (BS). The capacity of BS $s \in \mathcal{S}$ is denoted by r_s and its cost is denoted by c_s.

Within this area is a continuous spatial distribution indicating the overall traffic demand. Traffic demand in real-world cellular networks can be approximated using a spatial log-normal distribution (*cite*). We model this spatial traffic demand using a continuous form of the SSLT (Scalable, Spatially-correlated, and Log-normally distributed Traffic) model as proposed by Lee, Zhou, and Niu (*cite : F - Spatial Modeling SSLT*).

To generate this spatial distribution over the area of consideration, an initial Gaussian field, $\rho^G = \rho^G(x, y), x \in [0, X], y \in [0, Y]$, is generated by:

$$\rho^G(x, y) = \frac{1}{L}\sum_{l=1}^{L} cos\left(i_l x + \phi_l\right) cos\left(j_l y + \psi_l\right) \tag{1}$$

where $\mathcal{L} \overset{\text{def}}{=} \{1, 2, \dots, L\}$ is a set of the products of two cosines with angular frequencies $i_l, j_l \sim \mathcal{U}(0, \omega_{max}), l \in \mathcal{L}$ and phases $\phi_l, \psi_l \sim \mathcal{U}(0, 2\pi), l \in \mathcal{L}$. As L increases in size, ρ^G approaches a Gaussian random field with a spatial autocorrelation dependent on ω_{max}.

The approximate Gaussian distribution ρ^G is then normalized to $\rho^S = \rho^S(x, y), x \in [0, X], y \in [0, Y]$, which has a standard normal distribution. The final log-normal distribution, $\rho = \rho(x, y), x \in [0, X], y \in [0, Y]$, is determined by assigning location and scale parameters μ and σ:

$$\rho(x, y) = exp\left(\sigma\rho^S(x, y) + \mu\right) \tag{2}$$

Let $\mathcal{M} \overset{\text{def}}{=} \{1, 2, \dots, M\}$ be the set of demand points; the value of total traffic demand at each point is denoted by d_m.

Let $u_{ms} \in [0, 1]$, $m \in \mathcal{M}$, $s \in \mathcal{S}$, represent the normalized capacity (with respect to r_s) of BS s at point m, i.e., the normalized maximum rate that a user can receive at point m from BS s. $u_{ms} = 0$ when m is outside the coverage area of s and $u_{ms} = 1$ when m is within a small distance of s.

We assume that a BS $s \in \mathcal{S}$ can be allocated between multiple demand points, and $\delta_{ms} \in [0, r_s]$, $m \in \mathcal{M}$, $s \in \mathcal{S}$ represents the rate of BS s that is allocated to point m.

Throughout this paper, stochastic variables will be differentiated from deterministic variables with a tilde ($\sim$) placed above the symbol.

III. Solution Approach

This might be a good place to discuss more context as a lead up to the problem being solved.

In this section, we detail our approaches for optimally allocating demand to available resources. First, the formulate the problem as a capacitated set-cover problem solved optimally via a two-stage stochastic optimization problem. Second, the optimization problem is approximated to a more computationally solvable form as a deterministic equivalent program. Finally, a genetic algorithm is used to further approximate the original problem.

A. Problem Formulation

In this subsection, we consider the initial stochastic optimization problem, which is the basis of the following reformulations and approximations.

Mohammad will need to help write or look over this section.

We introduce $z_s, s \in \mathcal{S}$ as a binary decision variable defined as follows:

$$z_s = \begin{cases} 1, & \text{if BS } s \text{ is selected,} \\ 0, & \text{otherwise} \end{cases}$$

and α as a weighting coefficient between the two stages.

Problem 1

$$\underset{\{s\in\mathcal{S}\}}{\text{minimize}} \left\{ \sum_{s\in\mathcal{S}} c_s z_s + \alpha \mathbb{E}\left[h\left(\boldsymbol{z},\, \tilde{\boldsymbol{u}}\right)\right] \right\} \tag{3}$$

subject to:

$$z_s \in \{0,\, 1\},\, \forall s \in \mathcal{S} \tag{4}$$

where $h\left(\boldsymbol{z},\, \tilde{\boldsymbol{u}}\right)$ is the optimal value of the second-stage problem, which is given by:

$$\underset{\{\delta_{ms},\, m\in\mathcal{M},\, s\in\mathcal{S}\}}{\text{minimize}} \left\{ -\sum_{m\in\mathcal{M}} \sum_{s\in\mathcal{S}} \delta_{ms} \tilde{u}_{ms} \right\} \tag{5}$$

subject to:

$$x_s = \mathbb{1}_{\left\{\sum_{m\in\mathcal{M}} \delta_{ms} > 0\right\}},\, \forall s \in \mathcal{S} \tag{6}$$

$$\sum_{s\in\mathcal{S}} \tilde{u}_{ms} \delta_{ms} \le d_m,\, \forall m \in \mathcal{M} \tag{7}$$

$$\sum_{m\in\mathcal{M}} \delta_{ms} \le r_s,\, \forall s \in \mathcal{S} \tag{8}$$

B. Deterministic Equivalent Reformulation

Mohammad will need to help provide citations for, write, or look over this reformulation.

One major obstacle with solving Problem 1 is that stochastic equations are difficult to impossible to solve with typical computational libraries. In order to be practical, Problem 1 must be reformulated such that it does not contain any stochastic variables.

Our approach for solving the proposed stochastic optimization formulation is to derive their deterministic equivalent programs (DEPs). The DEP is an equivalent reformulation of the original stochastic program, but only contains deterministic variables (*cite - old paper Stochastic Programming*).

Let $\Omega \overset{\text{def}}{=} \{1,\, 2,\, \ldots,\, O\}$ be defined as the set of discrete scenarios, each of which contains a sampled version of the stochastic variables within Problem 1. As O approaches infinity, Ω contains the entire scope of the stochastic variables. With sufficiently large O, Ω approximates the stochastic variables of Problem 1 with deterministic variables. The probability of a given scenario $\omega \in \Omega$ occurs is given by $p^{\{\omega\}}$, $\omega \in \Omega$, where $\sum_{\omega\in\Omega} p^{\{\omega\}} = 1$.

Deterministic Equivalent Program of Problem 1

$$\underset{\left\{\substack{z_s,\, \delta^{\{\omega\}}_{ms},\\ s\in\mathcal{S},\, m\in\mathcal{M},\\ \omega\in\Omega}\right\}}{\text{minimize}} \sum_{s\in\mathcal{S}} c_s z_s - \alpha \sum_{\omega\in\Omega} p^{\{\omega\}} \left(\sum_{m\in\mathcal{M}} \sum_{s\in\mathcal{S}} \delta^{\{\omega\}}_{ms} u^{\{\omega\}}_{ms} \right) \tag{9}$$

subject to:

$$\sum_{s\in\mathcal{S}} u^{\{\omega\}}_{ms} \delta^{\{\omega\}}_{ms} \le d_m,\, \forall m \in \mathcal{M},\, \forall \omega \in \Omega \tag{10}$$

$$\sum_{m\in\mathcal{M}} \delta^{\{\omega\}}_{ms} \le r_s z_s,\, \forall s \in \mathcal{S},\, \forall \omega \in \Omega \tag{11}$$

$$z_s \in \{0,\, 1\},\, \forall s \in \mathcal{S} \tag{12}$$

Within each scenario ω, the log-normal demand field ρ is sampled to provide a set of M discrete demand points. Each sampling of ρ is generated by creating a non-stationary 2D Poisson point process with M points over the region using ρ as the spatial intensity function.

C. Genetic Algorithm

The following are draft notes for this section:

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

The following are draft notes for this section:

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 following are draft notes for this section:

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.