Approaches to Joint Base Station Selection and Adaptive Slicing in Virtualized Wireless Networks

Kory A. Teague

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Allen B. MacKenzie, Chair

Luiz DaSilva

R. Michael Buehrer

Mohammad J. Abdel-Rahman

August 1, 2018 (TBD)

Blacksburg, Virginia

Keywords: TBD

Approaches to Joint Base Station Selection and Adaptive Slicing in Virtualized Wireless Networks

Kory A. Teague

(ABSTRACT)

This work received support in part from the National Science Foundation via work involved with the Wireless @ Virginia Tech research group.

Contents

List of Figures

List of Tables

List of Acronyms

3GPP Third-Generation Partnership Project

5G-NR 5G New Radio

BS Base Station

CapEx Capital Expenditures

DEP Deterministic Equivalent Program

DSA Dynamic Spectrum Access

EB Exabytes

GA Genetic Algorithm

IoT Internet-of-Things

MIMO Multiple-Input Multiple-Output

mmWave Millimeter Wave

MNO Mobile Network Operator

MVNO Mobile Virtual Network Operator

NVS Network Virtualization Substrate

NwoB Networks without Borders

OpEx Operational Expenditures

PPP Poisson Point Process

QoS Quality of Service

RAN Radio Access Network

RP Resource Provider

SAA Sample Average Approximation

sDEP Sampled Deterministic Equivalent Program

SP Service Provider

SSLT Scalable, Spatially-correlated, Log-normal distributed Traffic

VNB Virtual Network Builder

VWN Virtualized Wireless Network

ZB Zetabytes

Chapter 1

Introduction

Mobile carriers have seen explosive growth in both the volume of users and the demands of those users within the networks they operate. New and evolving data-driven applications, such as audio/video streaming, social networking, and the Internet-of-Things (IoT) have also placed increasing demand upon the networks. In 2016, the amount of IP data handled by mobile networks exceeded 86 Exabytes (EB); it is projected to reach almost 200 EB in 2018, and 580 EB in 2021 [1]. Due to this exponential growth, incremental approaches to improve the network will fail to satisfy demand. As this growth continues in the near future, new architectures like 5G and its associated technologies will be needed to keep pace with the demand.

However, deployment of these technologies and networks can be a costly, prohibitive venture. To meet these demands requires a similar increase in capital (CapEx) and operational

expenditures (OpEx). As volumes and costs rise and margins shrink, approaches to reduce these expenditures become increasingly necessary. Resource infrastructure sharing has been a common practice for mobile network operators (MNOs) going back to 2G and 3G networks. First, MNOs needed to offer coverage for their users in regions where they had no infrastructure, leading to the creation of roaming agreements, eliminating the need for deploying new infrastructure in that region and reducing CapEx. Second, by sharing passive elements of the infrastructure, such as physical sites, tower masts, power, and air-conditioning, the CapEx of deploying new backhaul and radio access networks (RANs), such as cellular base stations (BSs), has decreased [2].

CapEx reductions from passive resource sharing drove an interest in resource sharing of the active elements of the network. For example, MNOs might share RANs, core networks, BSs, antenna systems, or backhaul, which leads to reductions in both CapEx and OpEx. The ability to share these active resources removed the necessity for network operators to own and maintain a physical network while providing actual MNO-like mobile services. These mobile virtual network operators (MVNOs) function similarly to MNOs, but operate a virtualized wireless network (VWN) comprised of virtual resources instead of physical resources without the associated CapEx. It has been shown that virtualization in this manner can increase overall demand satisfaction of a set of VWNs while decreasing overall cost (i.e., OpEx) by decreasing the idle capacity of the networks [3].

In order to take advantage of increasing virtualizable resources and competition for those resources, a specific problem must be solved: how to select the set of virtual resources

to form a VWN that meets its demands with necessary or maximum demand satisfaction at minimum cost. The solution to this problem is further complicated in the context of multiple MVNOs, each with one or more VWNs with unique demands, assessing a large pool of available virtual resources that can be adaptively allocated as demands shift.

This thesis addresses the topic of resource selection and adaptive slicing within cellular VWNs through the lens of stochastic optimization and investigates two approaches to efficiently reach a solution.

1.1 Trends in Wireless Networking

IP traffic is increasing across all types of networks, and is trending to become more mobile focused. According to the Cisco Visual Networking Index [1], global IP traffic will increase nearly threefold over the 2016-2021 time period, reaching 3.3 Zetabytes (ZB) annually in 2021 from 1.2 ZB annually in 2016. Traffic across the fixed internet backbone is projected to match this threefold pace, growing from 790 EB to 2.2 ZB. However, mobile data traffic is projected to have twice the growth of fixed internet over the same period, increasing almost sevenfold from 86 EB in 2016 to 580 EB in 2021. Traffic from wireless and mobile devices combined will reach 63% of total IP traffic by 2021, up from 49% in 2016. By 2021, smartphone IP traffic (33% of global IP traffic) will alone outnumber PC IP traffic (25%). In both the consumer and business markets, this demand includes enormous growth in video applications, specifically that of video streaming. By 2021, global IP video traffic

will reach 82% of all consumer internet traffic, up from 73% in 2016. By 2021, internet live video streaming will account for 13% of this video traffic, growing 15-fold over the period. Similarly, virtual and augmented reality uses will see the largest increase, growing at a 82% compound annual growth rate, and expected to reach a 20-fold increase between 2016-2021.

The technology underlying the mobile data network needs to continue to evolve with these changing trends and growth. The primary focus of the 5G cellular standard has been to meet these targets in an effective and robust manner. Of specific interest is that of aggregate data rate (e.g., area capacity, the available amount of data a network can facilitate over a unit area) and edge rate (e.g., 5% rate, the minimum data rate that can be reasonably provided to all but 5% of users) of the network. For 5G, the general consensus is that aggregate data rate and edge rate must be 1000x and 100x that of 4G, respectively [4]. To supply these rates, several strategies are being investigated, with three primary technologies being (1) the continuing of cellular densification and offloading, (2) increased bandwidth by expanding into new spectra like Wi-Fi and millimeter wave, and (3) increasing spectral efficiency through advances such as those in massive multiple-input multiple-output (MIMO).

The first strategy is extreme densification and offloading. By making network cells smaller, the number of active nodes increases for the same unit area. This is a common strategy across cellular generations, and a large impetus behind the use of smaller range RANs like microcells and femtocells [5]. Cell sizes have shrunk, dropping from the order of hundreds of square kilometers to now fractions of a square kilometer. The most important benefit of cell densification is that it increases spectral reuse, which reduces the amount of

users competing for the same resources. Theoretically, since signal-to-interference ratio is maintained as the cell shrinks, such densification can be repeated indefinitely as deployments allow [4, 6].

The second strategy is to increase bandwidth through the use of previously unused spectra such as millimeter wave (mmWave) and Wi-Fi. Cellular networks have utilized microwave frequencies ranging from a few centimeters to about a meter in wavelength; this range has become thoroughly occupied and to generate new bandwidth would require expanding to new frequencies [?]. Up to now, mmWave has been unused and in some cases unlicensed due to very poor propagation properties and high equipment costs. However, equipment costs are falling rapidly due to technological maturation. Further, the propagation qualities are increasingly surmountable as cell sizes shrink [?].

cite (1)

cite (2)

The third strategy involves the use of massive MIMO to increase spectral efficiency. MIMO uses multiple transmit and receive antennas to exploit multipath signal propagation, multiplying the capacity of a given radio link. The technology has been used for over a decade as a component of Wi-Fi before being introduced into the 3G and 4G standards [4]. A new approach to be used in 5G is that of "massive MIMO", where the number of transmit antennas at the BS greatly outnumber the number of active users [7]. For example, a given BS might have hundreds of antennas while maintaining data links for tens of users. This provides several benefits, most importantly vastly improving spectral efficiency.

5G must supply these rates at much higher energy and cost efficiencies, ideally matching or exceeding the capacity increases to avoid increasing overall network energy use and OpEx.

However, technologies that have been investigated to adequately increase the capacity of the network have several major hurdles to meet in order to be implemented at the desired energy and cost efficiencies. Massive MIMO requires the deployment of a vast number of antennas, which requires new BS architectures that have issues with scalability and cost. Millimeter-wave is more expensive than the more mature hardware of typical cellular bands. Decreasing cell size for cellular densification allows for smaller, cheaper BSs, but this decreased cost may not keep pace with the required number of increased deployments. [4]

3GPP (the Third-Generation Partnership Project) is currently working on finalizing the standard for 5G implementations. In December 2017, 3GPP froze the first half of release 15 of the 5G standard, covering 5G New Radio (5G-NR) which establishes specifications for new standalone 5G deployments. It is expected that 3GPP will freeze the second half of release 15, establishing the specifications of non-standalone 5G which utilizes existing LTE networks, in Summer of 2018. Further work on release 16 and beyond is still in progress.

1.2 Virtualization, Virtualized Wireless Networks, and the Networks without Borders Paradigm

One approach towards minimizing CapEx and OpEx of networks has been the utilization of resource sharing. Resource sharing encompases the sharing of resources between multiple networks and can take the form of *passive sharing*–referring to the sharing of physical sites,

tower masts, cabling, power supplies, and other components that are not actively on part of the network architecture–and *active sharing*–referring to the sharing of the active network architecture itself, such as backhaul and RAN. The practice has been utilized since 2G and 3G networks as a tool for reducing CapEx in expanding the network [2]. Since then, resource sharing has become more common; it is now available, standardized [8], and implemented in many major carrier networks. As reported by Costa-Perez et al. [9], a 2010 market survey [10] found that over 65% of European MNOs have deployed mobile infrastructure sharing in some form. It was further reported [9] that 20% of cells carry about 50% of total network traffic, with the remaining 80% of cells still causing OpEx with less utility. Through active resource sharing, networks can reduce or avoid redundant deployments and wasted capacity, reducing overall CapEx and OpEx.

The increasing prominence of active resource sharing challenges the traditional model of ownership of the various network layers and elements. Once it became feasible for network operators to utilize resources owned and maintained by other operators, it became possible for these MNOs to operate networks primarily or only using these shared resources. A given shared resource can be decoupled from a specific physical resource. This enables it to be adaptively associated with any of a given pool of qualifying physical resources as network conditions allow, establishing the shared resource as a virtual resource. Further, virtual networks can adapt to changing network conditions, adding and removing virtual resources as capacity requirements change. For example, an MVNO with a network of virtual resources can add additional virtual resources during peak hours when additional capacity for end

user satisfaction is needed, and removing unneeded resources during times of low demand to reduce OpEx of the network.

Other research has virtualization to improve performance in wireless networks. Panchal and Yates [11] have shown on an LTE testbed that active inter-operator resource sharing improves performance of overloaded networks in terms of decreased drop probability and overloaded sectors. Sharing methods that included virtualization provided further, albeit marginal, performance improvements at an increased complexity. In this capacity, improved performance allows for smaller networks reducing CapEx and OpEx. Costa-Perez et al. [9] found in LTE testbeds that network virtualization substrate (NVS), a suggested virtualization technique, provides improved overall throughput compared to networks without resource sharing.

1.2.1 Virtualization and the Network Value Chain

This concept of virtualization partitions the classical wireless networking value chain, allowing for specialization of segments of the value chain into new entities such as resource providers and service providers [2]. Traditional MNOs control every segment of the typical mobile network value chain (Fig. 1.1 [12]), from spectrum to the end user. With the introduction of virtualization techniques, MVNOs can obtain access to bulk network services available from an MNO. This allows for MVNOs to specialize without the significant CapEx or responsibility to deploy and maintain the radio infrastructure [12]. For example,

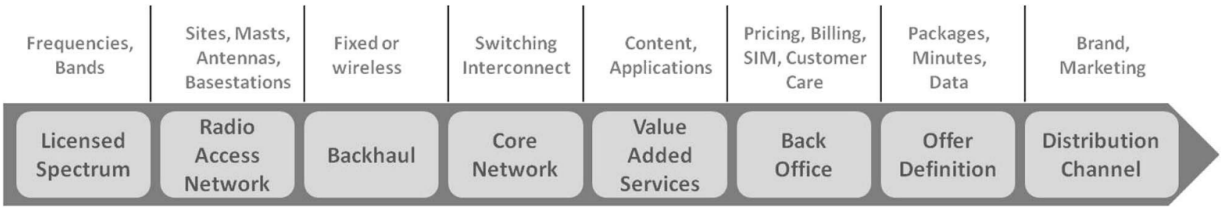


Figure 1.1: Typical MNO value chain [12]

an MVNO could focus on marketing, working solely within the distribution channel to the network's customers, or the MVNO could establish itself earlier in the value chain, focusing on operating the network from the core network.

Specialization of networks and the entities involved in the network can improve the cost efficiency of the network. According to Beckman and Smith [2]: "Extensive vertical integration is a characteristic of an immature product. As the product increases in complexity, it is no longer possible to [provide] an end-to-end solution." In both examples, the MVNO adds value to the traditional value chain by specializing in segments (e.g., marketing or service creation) that are different from the segments (e.g., network maintenance) still handled by the owner and operator of the network resources.

By focusing on the strengths provided by virtualization, more value can be generated through specialization. Doyle et al. [12] investigates the value chain with this segmentation in mind and introduces the Networks without Borders (NwoB) approach as a new service-oriented network with a proposed new value chain (Fig. 1.2 [12]). The network under the NwoB approach is entirely service-oriented, where the network responds to services and connectivity is tailored for the service. Services have a wider meaning than the voice, text,

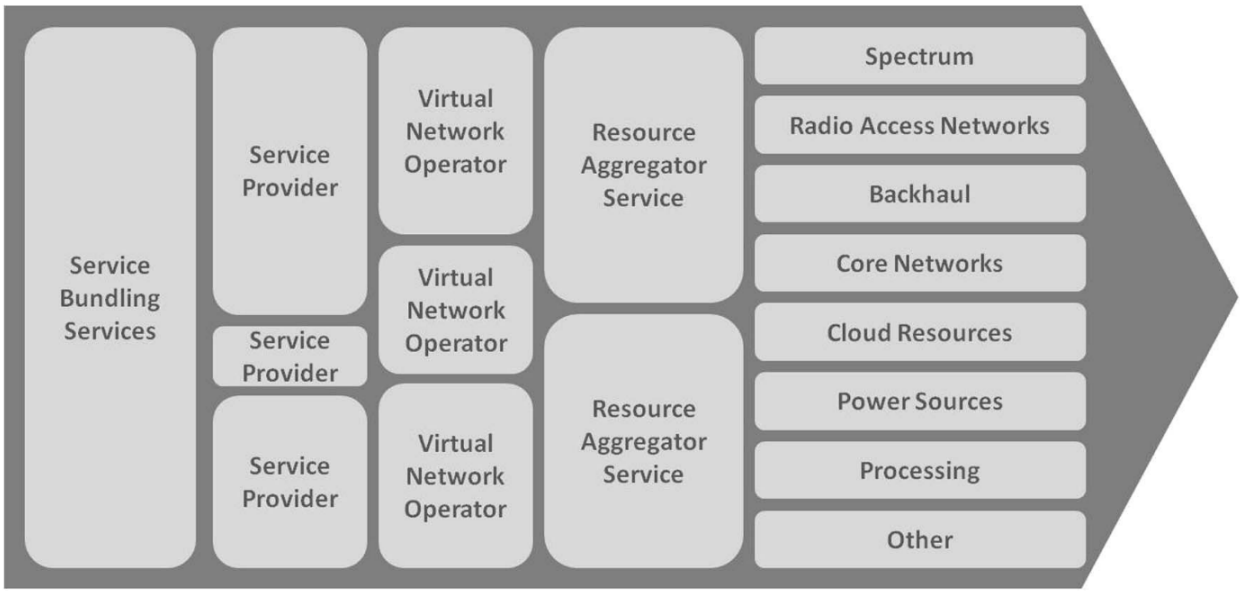


Figure 1.2: Proposed network value chain under the NwoB paradigm [12]

and data of a typical MNO. Services also include that of Netflix-like or real-time video streaming, IoT applications, or various types of over-the-top services. Each service would be provided by a service provider that compensates the virtual network operator operating a virtual network constructed specifically for the purpose of that service; the virtual network is the service. Unlike an MVNO which manages resources provided to it by agreement, the virtual network operator manages slices of virtual resources from a pool of all resources as provided through resource aggregating services.

The benefits of this paradigm as proposed by Doyle et al. [12] are four-fold. First, it provides specialization and independence for each stage, allowing service providers to focus on generating value from services provided. Second, networks can be specialized for a service, reducing OpEx through extensive resource sharing. Third, as resources are virtualized and pooled together, any resource (e.g., typical RAN, Wi-Fi, mmWave, raw spectrum) could

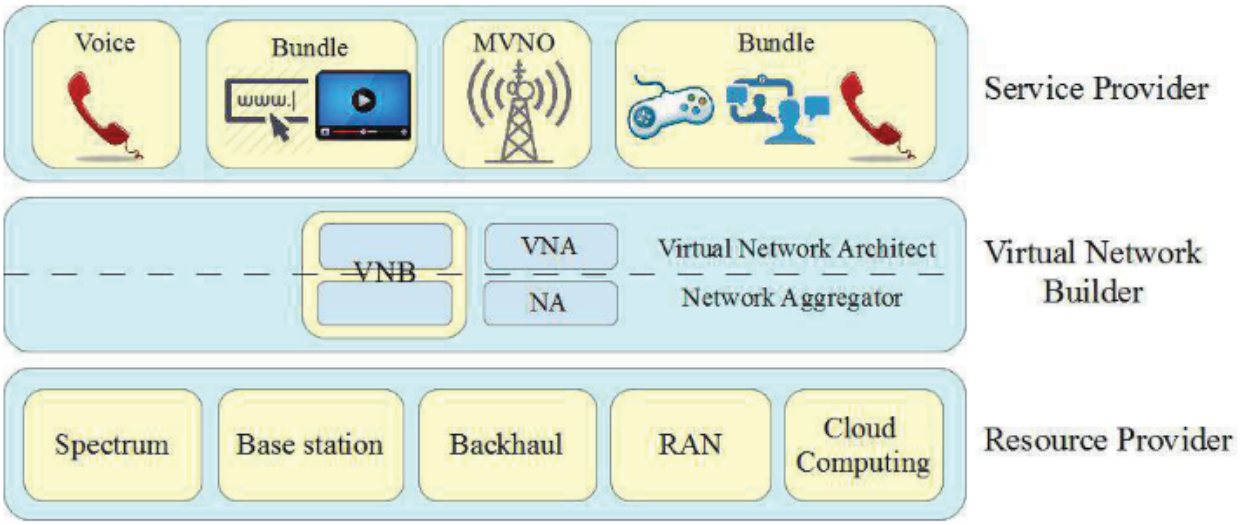


Figure 1.3: VWN architecture as used in this work [3, 13]

be added with the pool and utilized for a network as its properties fit the network's needs. Fourth, it lowers the barrier for entry and establishes services for new entities to fulfill.

1.2.2 Virtualization Architecture in this Work

Recognizing the critical nature of virtualization and resource allocation, this thesis develops and analyzes two methods for constructing virtualized wireless networks built on a virtualization architecture [3,13] inspired by the NwoB paradigm presented by Doyle et al. [12]. Fig. 1.3 [3,13] illustrates the three primary roles in this architecture: (1) the Resource Providers (RPs), (2) the Virtual Network Builders (VNBs), and (3) the Service Providers (SPs).

RPs deploy and maintain the physical resources that are to be virtualized and offered for use within the virtualization framework and are the various entities that occupy the rightmost column of segments (i.e., resources) in the NwoB value chain (Fig. 1.2 [12]). These

resources can be in the form of any network-capable resource. For example, the resources could be BSs as provided by a traditional MNO, a company- or individual-owned WLAN, femtocell access points, available licensed or unlicensed spectrum, or cloud computing. An RP is then any entity that offers a virtualizable resource, such as a traditional MNO, company, or individual. RPs maintain the resources, but also determine how the resource would be sliced and shared.

The VNB acts as resource aggregator, VWN constructor, and as intermediary between SPs and RPs. Therefore, the VNB acts as a combination virtual network operator and resource aggregator in the NwoB value chain (Fig. 1.2 [12]). The VNB aggregates the resources maintained by individual RPs to establish the pool of available virtual resources. The VNB also coordinates with SPs to understand the demands of their services and constructs VWNs tuned specifically to these demands. By understanding the needs of the services provided by the SPs, the VNB will evaluate which virtual and virtualizable resources available from the RPs are needed to construct the optimal[1] network for the SPs' needs, coordinate with the necessary RPs to obtain access to these resources for a given wholesale (OpEx) cost, and construct the network for the SPs to operate. Multiple VNBs can coexist, each with their potentially overlapping set of RPs from which to aggregate resources.

SPs operate similarly to the service providers in the NwoB approach. Primarily, an SP determines a service that they wish to provide, understands and enumerates the demands

[1]In this network context, "optimal" is loosely defined to mean a network that provides the maximum demand satisfaction for the SP at the minimum cost to be paid to the RP. These two requirements – maximum demand satisfaction and minimum cost – are frequently contradictory and need to be balanced by the VNB.

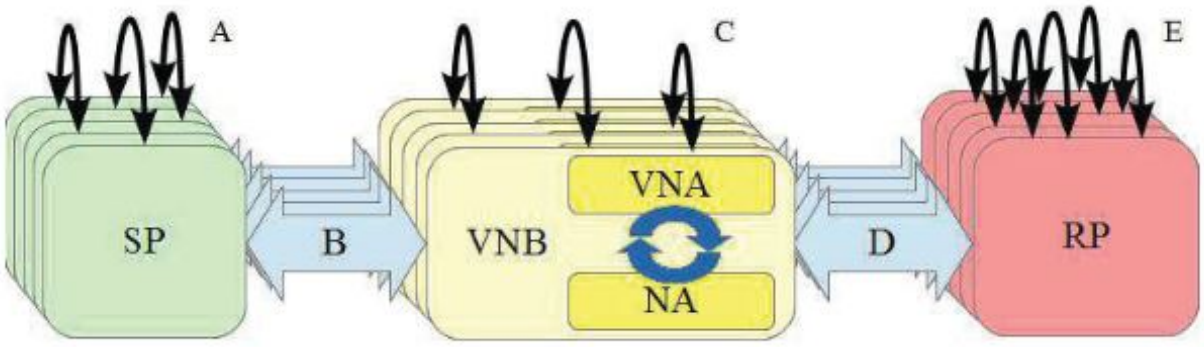


Figure 1.4: Interactions between roles in the VWN architecture [13]

that are to be satisfied for that service, and provides the service over the VWN to their end users. SPs can provide a wide range of services over the network. The service could be a traditional MNO or be providing MNO-like services, such as voice calling and texting. Services could cover specific applications, such as IoT, teleconferencing, augmented or virtual reality, or emergency services. Other examples include traditional over-the-top services, such as Netflix-like or real-time (live) video streaming, social media (Facebook, Twitter, etc.), messaging (Skype, Groupme, etc.), or news/content feeds. Further, an SP could also bundle several services, either through a single VWN built for the bundle, or by bundling services provided by several SPs.

Between these three entity roles, various interactions become possible. The most common interactions are illustrated in Fig. 1.4 [13]. The interactions between the various entity roles are: (A) among SPs; (B) between the SPs and the VNBs; (C) among VNBs; (D) between the VNBs and the RPs; and (E) among RPs. It should be apparent that across each of these interactions is the imposition of costs as exchange for the transfer of services, networks, and resources.

Interaction (A) describes associations among various SPs. This would typically occur in situations where a SP desires to bundle the services of several SPs, or when a SP wishes to utilize a specialized network operation from another SP. Generally, this interaction would be performed manually over timescales of weeks or months.

Interaction (B) describes associations directly between SPs and VNBs. This would be one of the most common interactions within this framework. This interaction is bidirectional. In the first direction, SPs would provide the VNB they are coordinating with the specific demands and needs for the service they are providing. In the opposite direction, the VNB utilizes these conveyed needs and demands to construct a VWN and provide it for use to the service provider. Ideally, this interaction is highly or entirely automated, with the interactions varying from minutes or hours to weeks or months based on the level of automation and the specifics of the interaction. It will require optimization techniques and/or machine learning to achieve satisfactory results in this interaction.

Interaction (C) describes associations among various VNBs. Generally, such interactions may occur when a VNB does not have access to the appropriate virtual resources to satisfy interaction (B) interactions. Obvious examples include not having the necessary resources to provide adequate coverage over geographical areas or capacity in high-density environments. These interactions would generally be performed manually over timescales of weeks or months.

Interaction (D) describes associations between VNBs and RPs. Similar to interaction (B), this would be the other of the most common interactions within this framework. It is

also very important, as it establishes the mapping between the virtual and physical resources and builds the substrate that the framework is built upon. VNBs interact with the RPs by making requests for new resources and releasing unneeded resources. RPs interact with the VNBs by issuing updates, such as any changes to the resources in the VNBs' available pool of resources. Updates such as these are potentially highly disruptive to the VNBs as the updates can impact a large number of VWNs managed by the VNBs. With further similarity to interaction (B), this interaction is highly dependent on automation; based on the level of automation, this interaction may occur over timescales of minutes or hours to weeks or months.

Interaction (E) describes associations among various RPs. In this interaction, various RPs establish connections with each other to facilitate proper mapping of physical resources to virtual through the use of quality of service (QoS) parameters that define the abstracted resources. For example, a small-scale RP containing only an individual-owned femtocell could connect with a larger RP via this interaction so that the resource within the small-scale RP is visible for association with a VNB over interaction (D) as handled by the larger RP. These interactions could take seconds to weeks depending on the complexities of the RPs, their resources, and the amount of human involvement.

Other work has been completed using this architecture. Abdel-Rahman et al. [3] constructed several resource allocation models, including one-stage programs, two-stage programs, and a one-stage stochastic program, to investigate the efficacy of this virtualization architecture upon a preexisting set of resources. The implementation focused on interaction

(B) from the perspective of the VNB, and showed that virtualization decreased the cost and idle capacity of the networks and increased demand satisfaction of the networks.

Cardoso et al. [13] expanded on this work by introducing a two-stage stochastic program to optimize interaction (B). The two-stage stochastic resource allocation similarly reduces cost and idle capacity of the VWN compared to the network without sharing. However, no direct comparisons are made with the non-stochastic programs tested by Abdel-Rahman et al. [3].

Gomez et al. [14] utilized this architecture from an economics perspective. Using a matching markets framework, they investigated the interaction of association between SPs and VNBs, such as the methods for how SPs indicate their needs and how VNBs indicate their VWN capabilities, and the fees that SPs will pay to partner for a VNB. Gomez expanded on this work in her Ph.D. dissertation [15].

Still reading and parsing these papers. Double check.

The focus of this thesis is on optimization approaches largely in the context of interaction (B). This problem involves establishing how SPs convey the demands needed by the VNB to construct an optimal VWN for the service provided by the SP. Further, the construction of the optimal VWN is sought within a short amount of time so that interaction (B) can be completed over shorter timescales (e.g., minutes or hours) instead of longer (e.g., days, weeks, months). With an optimal VWN in mind, construction of the VWN is inherently an optimization problem, and the search of expedient solutions lays within the study of optimization.

1.3 Review of Optimization Methods

In this thesis, I address the problem of the creation of optimal networks by a VNB that satisfy the specific demands of SPs using a pool of resources provided by a set of RPs. This is naturally an optimization problem, in which some objective function is either minimized or maximized. At it's most basic, optimization techniques (e.g., linear programming, integer programming) will find the set of input parameters that minimize or maximize a single decision variable – the value of the objective function – in context of a set of constraints.

1.3.1 Stochastic Programming

Standard linear and integer programming requires complete, certain knowledge of all parameters that affect the functions or model being optimized (i.e., the model's parameters and functions must be deterministic). Communications, especially wireless communications, can be highly non-deterministic as the communication channel introduces a large amount of uncertainty. Stochastic programming provides a powerful mathematical tool to handle optimization under such uncertainties.

Stochastic programming has been recently exploited to optimize resource allocation in various types of wireless communications operating under uncertainties. Abdel-Rahman et al. [3] exploit stochastic optimization within the framework of the virtualization architecture presented in Section 1.2.2 to minimize the cost of resource allocation by introducing probabalistic QoS guarantees. Cardoso et al. [13] expand on that work by introducing a second stage

to balance maximizing demand satisfaction while minimizing cost. Further examples include resource allocation in dynamic spectrum access (DSA) networks [16], optimal orchestration of LTE-U networks utilizing Wi-Fi access points [17], resource allocation in opportunistic LTE-A networks considering end user rate demand satisfaction [18], resource allocation for OFDMA-based cognitive radios considering primary user system interference [19], and predictive resource allocation for energy-efficient video streaming to mobile end users [20].

Introducing stochastic parameters and constraints allows the optimization model to consider probabilities within the optimization. In the case of resource allocation in networks, it may be possible to allocate enough resources to satisfy all end-user demand. Such an optimization may require too many resources to be economical considering the law of diminishing returns, with the solution being cost prohibitive. It is much cheaper to solve such that 95% or 99% of demand is satisfied, leaving some demand unsatisfied.

However, standard linear programming techniques cannot solve models with stochastic parameters. Stochastic programming therefore requires converting the stochastic program into its deterministic equivalent program (DEP) which replaces all stochastic variables with deterministic variables [21]. The process of forming a DEP from a stochastic program involves converting each stochastic variable into a set of all possible scenarios and scenario probabilities. These scenarios and scenario probabilities are present within the model as a new dimension and weight for the now-deterministic variable. To fully encapsulate the stochastic variable, the deterministic equivalent variable may be composed of an infinite set.

Resource allocation problems are typically some form of integer programming–in which

all decision variables (unknowns) are integers–or mixed integer programming–in which some decision variables (unknowns) are integers. Both integer and mixed integer programs are generally[2] considered NP-hard[3]. As the programs increase in scope, they become more computationally complex to solve; accounting for the scenarios of the previously stochastic variables further increases this complexity. Finding the optimal solution may require more time than is feasible; in the worst case, these problems run in exponential time complexity.

1.3.2 Metaheuristic Approaches

The use of heuristic or metaheuristic algorithms can provide close-to-optimal solutions in much better time. Examples include hill climbing, simulated annealing, ant colony optimization, and particle swarm optimization. Each of these approaches are iterative techniques.

Hill climbing starts with an arbitrary solution and makes incremental changes to variables, finding a new solution. If the new solution is better than the previous, the new solution is iterated upon. This continues until no further improvements can be made. Hill climbing will only find the local maximum close to the initial arbitrary solution, and is best in convex problems where the only local maximum is guaranteed to be the global maximum.

Simulated annealing is inspired by the process of annealing found in metallurgy, where

[2]Some subclasses of integer and mixed integer programs are efficiently solvable, but these are the exception. Several classic NP-Complete (a subset of NP-Hard) problems [22] are integer programs and mixed integer programs.

[3]Finding the minimum resource allocation that provides coverage over a geographic area falls within a category of problems referred to as *minimum set cover problems*. It is apparent that the problem considered in this thesis–specifically the stochastic program proposed in Section 2.2–is some form of minimum set cover problem; specifically, it might be referred to as a capacitated set cover problem. Minimum set cover problems are provably NP-hard and typically rely on approximation solutions to solve in a feasible amount of time [23].

metal is heated to the point where atoms can migrate, reducing defects in the crystalline structure. In simulated annealing, the model has some notion of temperature, which represents the internal energy of the system, and states, which represent possible solutions to the system being optimized. The system has an initial state, each state has an associated energy, and the system is attempting to reach the state of lowest energy. On each iteration, the heuristic considers a neighboring state, and chooses to transition to the new state with a probability dependent on the energy of the current state, the energy of the neighboring state, and the temperature. This transition can lead from a lower energy state (better) to a higher energy state (worse), and will do so more often while it has a higher temperature. Gradually, the system will cool and decrease the temperature, which causes the system to tend to select states with lower energy; as the temperature drops, the systems overall energy drops. When the temperature reaches zero, the system will only transition to states of lower energy (i.e., that are more optimal), reducing to the hill climbing algorithm.

Ant colony optimization is inspired by the behavior of ants. A colony of ants move around independently trying to find food, laying pheromones on the taken path. Upon crossing paths, ants have a probabilistic chance to follow the new path based on the strength of the pheromones of the new and old paths. Over time, pheromones evaporate, and paths less taken will weaken. Longer paths, since they take longer to traverse and will be reinforced less often, will also weaken. This has benefits over approaches like simulated annealing because it adapts in real time.

In particle swarm optimization, a number of candidate solutions, called particles, are

created that move semi-chaotically. In each iteration, every particle will move according to its velocity. Each particle has it's best known position, and it's velocity updates in a way that is guided by their own best known position and the swarm's best known position. This allows a large portion of the search space to be investigated, with candidate solutions exploring regions containing local maxima until it settles to exploit and find the best found local maxima.

In this thesis, I utilize a genetic algorithm as an approach for optimization. A genetic algorithm is a form of evolutionary algorithm, a set of algorithms which are inspired by biological evolution and natural selection. Each iteration is called a *generation* and is composed of a number of candidate solutions called *individuals*. Each individual is defined by a *chromosome* which details the specific candidate solution. During each generation, every individual is evaluated on its *fitness*, a function dependent on the individual's chromosome; the higher the individual's fitness, the more optimal the individual. Individuals called *parents* are then randomly selected to pass their chromosome onto the next generation in a process called *selection*; in selection, more fit individuals are more likely to be selected. With a certain probability, groups of parents will undergo *crossover* and exchange the data contained within their chromosomes to form new *children* that are a mixture of the parents; if mixing does not occur, the parents are cloned into the next generation as children. Then, individual bits within the children's chromosomes have a chance to flip, or *mutate*. The resulting children from crossover and mutation form the entire next generation.

Since chromosomes from fitter individuals are more likely to pass on to subsequent genera-

tions, generations gradually become fitter. Through crossover, fit chromosomes may combine to form fitter children that proliferate; less fit children are often also formed, but are generally not selected for later generations. Mutation introduces diversity into the generations, which expand the exploration of the search space. More details, including that of implementation and variants, will be expanded upon in Section 3.2.

Genetic algorithms have been used previously as approaches for simplifying the search spaces of large, complex stochastic optimization problems. For example, Cui et al. [24] used a genetic algorithm where each chromosome defined a subproblem of larger resource allocation optimization problem, and the fitness was evaluated by solving the subproblems with linear programming optimization methods. One approach investigated in this thesis coordinates a genetic algorithm with an optimization program wherein the genetic algorithm solves for and fixes a decision variable to simplify the larger optimization program. Hybrid approaches (e.g., Cui et al.) and other effective metaheuristic algorithms (e.g., ant colony optimization, particle swarm optimization, neural networks, and machine learning) are worth investigating in the context of the posed VWN architecture, but beyond the scope of this thesis.

1.4 Thesis Objective

The objective of this thesis is to develop two approaches to joint resource allocation to construct a set of a VWNs and adaptively slice the selected resources to allocate to the individual VWNs. A model will be presented as the context for these approaches, expanding

upon the VWN architecture proposed in Section 1.2.2. The validity of this model will be restricted to the scope of cellular networks using generic base stations as its resources. The two proposed approaches will be performed within the VNB, and evaluated in four cases that differ in the resources provided by the RPs and service demands to be satisfied by the SPs. Accordingly, the efficacy of these approaches will be measured primarily by the optimality of the solutions, such as cost and network service demand satisfaction, and the run time, providing the VNB with a sufficient solution in a reasonable amount of time.

1.5 Thesis Outline

This thesis is organized as follows. Chapter 2 defines the model used for the resource allocation methods explored in this thesis. Further, Chapter 2 also details the two-stage stochastic optimization problem which optimally performs resource selection and slicing as a basis of approaches presented within this work. Chapter 3 establishes the two approaches investigated to provide solutions to the optimization problem posed in Chapter 2: a sampled deterministic equivalent program which solves the problem as a whole and a genetic algorithm that simplifies the problem by providing an estimated optimal resource selection. **??** tests these two approaches by presenting four data sets that mimic real world cellular networks and evaluates the results. Chapter 5 contains the conclusions and proposed future work in this area.

Chapter 2

Virtual Network Builder Model

This chapter establishes the mathematical foundation for this thesis. First, a geographic model is presented defining an area of interest, the pool of resources maintained by the RPs for use by the VNB, and a characterization of the service demand of each SP. Second, a specific demand model, based on the empirical analysis of collected cellular network traces, is presented. This is the demand model that will be used throughout this thesis. Third, a two-stage stochastic program is developed to model the problem of resource selection and adaptive slicing within the VNB.

2.1 Network Area Definitions

Consider a geographic region, $\mathcal{R}$, of width X meters and length Y meters that contains a VNB and a set $\mathcal{S} \stackrel{\text{def}}{=} \{1, 2, \ldots, S\}$ of virtualized resources (i.e., BSs) the VNB has aggregated

for use in the construction of VWNs. The pool of BSs, $\mathcal{S}$, is mapped to physical BSs owned and maintained by RPs and made available for use through contracts between the RPs and the VNB. The contract-negotiated cost for the VNB to lease BS $s \in \mathcal{S}$ is denoted by c_s. The costs for the BSs used within a constructed VWN are passed to the SPs as part of the overall cost of the network. The rate capacity of BS $s \in \mathcal{S}$ is denoted by r_s and its coverage radius is denoted by b_s.

Let $\mathcal{N} \stackrel{\text{def}}{=} \{1, 2, \ldots, N\}$ be the set of SPs seeking a VWN with coverage within the region $\mathcal{R}$. An SP $n \in \mathcal{N}$ associates with the VNB to create the desired VWN. Through this association, SP n must coordinate with the VNB to indicate the demands of the intended service the VWN would need to satisfy. SP n must know and communicate to the VNB the estimated geographic distribution of the service's traffic demand density (or demand *intensity*) as a function, $\lambda_n(x, y)$, $n \in \mathcal{N}$, $x \in [0, X]$, $y \in [0, Y]$, in terms of $\frac{\text{bits}}{\text{km}^2}$. The demand intensity function could be in the form of a continuous function or as discrete pixels (e.g., a bitmap), and indicates the locations of necessary coverage and the desired capacity within specific regions of the area. Examples of possible maps could be for services such as localized video streaming (specific, localized coverage with high regional capacities) or MNO-like voice lines (broad coverage with comparatively low capacity). For an example of λ_n, see Section 2.1.1.

Further, the SP would also provide the desired or needed percent demand satisfaction rate for the service. Some services have high priority, such as those related to emergency services, and must have nearly if not perfect 100% demand satisfaction. Others, such as

Diversifying α to very according to the various SPs (i.e., $\alpha_n, n \in \mathcal{N}$, or $\alpha_m, m \in$

the aforementioned generic voice lines or video streaming, can withstand some demand to remain unsatisfied as a tradeoff for decreased network leasing or operational costs.

Let $\mathcal{M}_n \stackrel{\text{def}}{=} \{1, 2, \ldots, M_n\}$ be the set of demand points SP $n \in \mathcal{N}$ is attempting to satisfy with its service. Each demand point $m \in \mathcal{M}_n$ is seeking to connect to the VWN operated by SP $n \in \mathcal{N}$ and is located within the domain of $\mathcal{R}$ (i.e., $(\tilde{x}_{m_n}, \tilde{y}_{m_n})$, $\tilde{x}_{m_n} \in [0, X]$, $\tilde{y}_{m_n} \in [0, Y]$). Demand point locations are stochastically determined by the distribution of traffic as described by the demand intensity function λ_n. A realization of these demand points can be found as a two-dimensional non-stationary (or inhomogeneous) Poisson point process (PPP) using λ_n as it's spatial intensity function.

For purposes of visualization or compuation, this non-stationary PPP is generatable using an accept-reject method [25]. A stationary PPP is generated relative to the maximum value of the traffic demand density function within $\mathcal{R}$. That is, a number of points generated within the region is selected from a Poisson random variable with parameter (or mean) $\lambda_{n,\max} * X * Y$, where $\lambda_{n,\max}$ is the maximum value of λ_n within $\mathcal{R}$. Each point is then independently and uniformly distributed (i.e., each point has a location (x, y) with $x \sim \mathcal{U}(0, X)$[1] and $y \sim \mathcal{U}(0, Y)$) over $\mathcal{R}$. Then, each point undergoes the accept-reject procedure to inhomogenize the stationary PPP. Each point is kept with a probability of the ratio of the value of the intensity function at that point's location to the maximum value of the intensity function. That is, for each point in the PPP a uniform random variable, $P \sim \mathcal{U}(0, 1)$, is generated and the point is either *accepted* and kept or *rejected* and discarded according to

[1] $\mathcal{U}(a, b)$ refers to a random variable uniformly distributed over the domain $[a, b]$, where $a < b$.

$$\begin{cases} \text{the } i^{\text{th}} \text{ point is kept,} & \text{if } P \leq \frac{\lambda(x_i, y_i)}{\lambda_{n,\max}}; \\ \text{the } i^{\text{th}} \text{ point is discarded,} & \text{otherwise,} \end{cases} \tag{2.1}$$

where x_i and y_i are the x- and y-coordinates of the i^{th} point of the stationary PPP.

Generally, stationary PPPs and non-stationary PPPs generate a number of points correlating to the intensity value and function, respectively. A specific number, M_n, of points can also be generated as necessary to populate a realization for $\mathcal{M}_n$, $n \in \mathcal{N}$. Instead of generating a random number of points according to a Poisson random variable, points are generated one at a time and individually either kept or discarded as defined in eq. (2.1). Once M_n points have been generated and kept, a non-stationary PPP of $\mathcal{M}_n$ has been generated.

This is allowed because, by definition, each point in a PPP is independent and identically distributed; each point is generated independently and identically distributed according to a uniform distribution. Generating more points is only indicative of a higher intensity PPP. Specifically, the number of points generated in the initial, stationary PPP is linearly dependent on a Poisson random variable with mean $\lambda_{n,\max} * X * Y$; doubling the expected number of generated points correlates with a doubling of $\lambda_{n,\max}$, which itself correlates with a scaled doubling of $\lambda_n(x,\, y)$. Scaling $\lambda_n(x,\, y)$ according to any desired number of points does not change the overall structural characteristics of the underlying described distribution.

Each demand point $m \in \mathcal{M}_n$ loads the VWN of SP $n \in \mathcal{N}$ with point traffic demand denoted by d_{mn}. So that the total demand described by λ_n is allocated by the points in $\mathcal{M}_n$,

the overall demand

$$D_n = \int_0^X \int_0^Y \lambda_n\,(x,\, y)\, dy\, dx, n \in \mathcal{N}, \tag{2.2}$$

of the demand density distribution is evenly distributed such that

$$d_{mn} = \frac{D}{M_n},\, m \in \mathcal{M}_n,\, n \in \mathcal{N}. \tag{2.3}$$

Let $\tilde{u}_{mns} \in [0,\, 1]$ represent the normalized capacity (with respect to r_s) of BS $s \in \mathcal{S}$ at point $m \in \mathcal{M}_n, n \in \mathcal{N}$, associated with SP $n \in \mathcal{N}$ (i.e., the normalized maximum rate that a user can receive at point m from BS s). Specifically,

$$\tilde{u}_{mns} \stackrel{\text{def}}{=} \begin{cases} 0, & \text{if demand point } m \text{ of SP } n \text{ is located more than } b_s \text{ meters from from BS } s; \\ 1, & \text{if demand point } m \text{ of SP } n \text{ is located within a small distance of BS } s; \\ (0,\, 1), & \text{otherwise.} \end{cases} \tag{2.4}$$

It is apparent that $\tilde{u}_{mns}$ will vary according to the path-loss characteristics of the environment, the locations of the demand points, and other various factors. In some instances, it can be beneficial to simplify this definition such that

$$\tilde{u}_{mns} \stackrel{\text{def}}{=} \begin{cases} 0, & \text{if demand point } m \text{ of SP } n \text{ is located more than } b_s \text{ meters from BS } s; \\ 1, & \text{otherwise (i.e., if demand point } m \text{ of SP } n \text{ is located less than or equal to } b_s \text{ meters from BS } s). \end{cases} \tag{2.5}$$

This simplification allows the available pool of BSs, or any subset thereof, to be easily visualized as a Voronoi tessellation [26]. In a Voronoi tessellation, a two-dimensional plane is tessellated into a set of convex polygons, each of which is defined by a single point contained within. All points comprising the area enclosed by a polygon is closer to that polygon's defining point than any other polygons'. Using the simplified definition for $\tilde{u}_{mns}$, the Voronoi tessellation of a set of BS locations functions as a coverage map for those BSs, assuming that all polygons are within the respective ranges for their associated BSs; each polygon represents a region where $\tilde{u}_{mns} = 1$. This binary condition is a simplification that aids in the implementation of the approach used in Section 3.2, and is generally assumed in the rest of this thesis.

From the perspective of the VNB, each SP is only distinguishable by its overall demand characteristics. These demand characteristics are defined by its demand density distribution which defines $\tilde{u}_{mns}$. The VNB must construct a VWN for each SP, but for optimal VWNs to be created, the VNB must consider the demands of all SPs simultaneously and in context of each other. For the VNB, all SP demand points are indistinguishable. Therefore, the VNB considers a single set of demand points

$$\mathcal{M} \stackrel{\text{def}}{=} \bigcup_{i=1}^{N} \mathcal{M}_i = \{1, 2, \ldots, M\}, \tag{2.6}$$

where $M = \sum_{i=1}^{N} M_i$, with demands d_m, $m \in \mathcal{M}$, and stochastic normalized capacities $\tilde{u}_{ms}$, $m \in \mathcal{M}$, $s \in \mathcal{S}$.

I assume that a resource $s \in \mathcal{S}$ 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 resource s that is allocated to point m. δ_{ms} is said to be the *slice* of BS $s \in \mathcal{S}$ that is allocated to demand point $m \in \mathcal{M}$. The VWN constructed for SP $n \in \mathcal{N}$ is comprised of all of the slices allocated to the SP's demand points. That is, the VWN constructed for SP $n \in \mathcal{N}$ is

$$\bigcup_{i=1}^{M} \delta_{is}, \; s \in \mathcal{S}, \tag{2.7}$$

where $\delta_{ms} = 0$, $\forall m \notin \mathcal{M}_n \subseteq \mathcal{M}$ and $\delta_{ms} > 0$, $\forall m \in \mathcal{M}_n \subseteq \mathcal{M}$.

Throughout this thesis, stochastic variables will be differentiated from deterministic variables with a tilde ($\sim$) placed above the symbol (e.g., $\tilde{u}_{ms}$).

2.1.1 Example Demand Distribution; The SSLT Model

One major assumption made in Section 2.1 is that SPs must communicate the demand characteristics (i.e., λ_n) of their service to the VNB to properly facilitate VWN construction. In this subsection, I establish an example model demonstrating the demand characteristics

to be communicated by generating an example hypothetical demand intensity function. For testing the approaches for VWN construction presented in Chapter 3, this example is the fundamental model used for simulating SP demand in cellular network-based services.

Gotzner et al. [27] have shown that a log-normal distribution[2] can approximate traffic demand in real-world cellular networks. It has also been shown that traffic distributions are spatially correlated [29,31]. Lee et al. [28,32] presented the Scalable, Spatially-correlated, and Log-normally distributed Traffic (SSLT) model to emulate the characteristics of real world cellular data networks. This model is flexible and can be adjusted to simulate numerous cellular networks, and can characterize the demand of a supposed SP in a way that mimics real-world data. I use a variant of the SSLT demand model presented by Lee et al., which I altered to be a continuous function serving as a continuous or pixelated demand density map.

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

$$\lambda^G(x,\, y) = \frac{1}{L} \sum_{l=1}^{L} \cos(i_l x + \phi_l)\ \cos(j_l y + \psi_l) \tag{2.8}$$

where $\mathcal{L} \stackrel{\text{def}}{=} \{1,\, 2,\, \ldots,\, L\}$ is a set of the products of two cosines with stochastic 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, it is expected that λ_G approaches a Gaussian random field according to the central limit

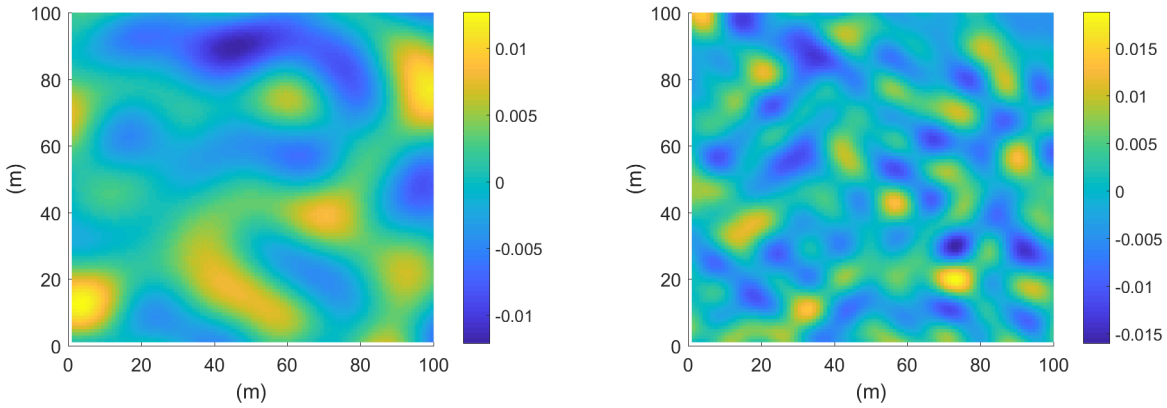
[2]It has also been shown that traffic distributions in cellular networks can be more accurately approximated by a Weibull distribution [28], by mixtures of log-normal distributions [28, 29], or by an α-stable distribution [30].

theorem.

According to Lee et al. [32], λ^G is spatially correlated with autocorrelation function

$$R\left(dx,\, dy\right) = E\left[\lambda^G\left(x,\, y\right)\ \lambda^G\left(x+dx,\, y+dy\right)\right] = \frac{1}{4L}\ \mathrm{sinc}\left(\omega_{\max} dx\right)\ \mathrm{sinc}\left(\omega_{\max} dy\right). \quad (2.9)$$

The autocorrelation function is notably dependent on the maximum angular frequency defining ρ^G, $\omega_{\max}$. As $\omega_{\max}$ increases, the demand of adjacent regions become less correlated. $\omega_{\max}$ is effectively a measure of the inhomogeneity of ρ^G. This effect of $\omega_{\max}$ on the inhomogeneity of λ^G is shown in Fig. 2.1; Fig. 2.1b fluctuates more rapidly than Fig. 2.1a, which is generated from a smaller $\omega_{\max}$.



(a) $\omega_{\max} = \frac{2\pi}{30}$ (b) $\omega_{\max} = \frac{4\pi}{30}$

Figure 2.1: Two example Gaussian random fields (i.e., λ^G) for generating the SSLT demand model, varying by $\omega_{\max}$. $L = 100000$, $X = 100$, $Y = 100$

The approximate Gaussian distribution λ^G is then normalized to a standard normal distribution[3], $\lambda^S = \lambda^S\left(x,\, y\right),\ x \in [0,\, X],\ y \in [0,\, Y]$,

[3]A standard normal distribution is a Gaussian distribution with mean $\mu = 0$ and variance $\sigma = 1$.

$$\lambda^S(x,\, y) = \frac{\lambda^G(x,\, y) - \overline{\lambda^G}}{\sqrt{\mathrm{Var}\left(\lambda^G\right)}}, \tag{2.10}$$

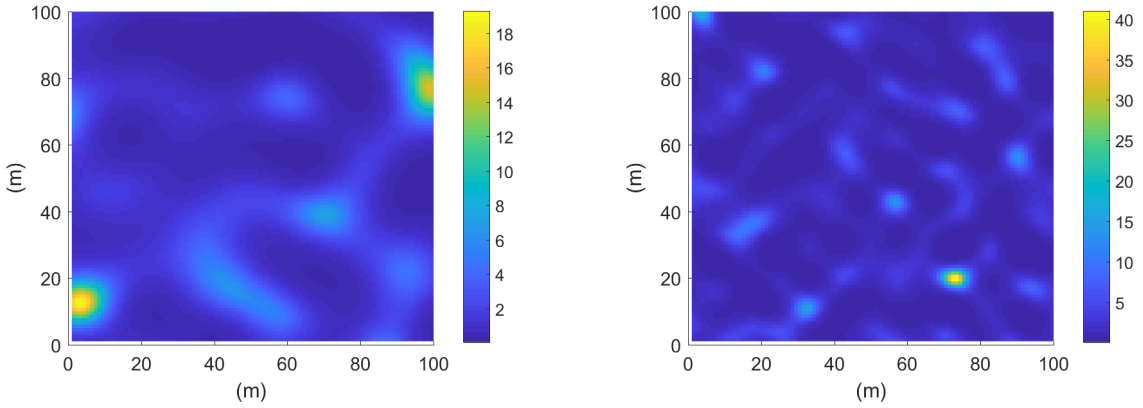
where $\mathrm{Var}\left(\lambda^G\right) = \mathbb{E}\left[\left(\lambda^G\right)^2\right] - \mathbb{E}\left[\lambda^G\right]^2$ is the variance of λ^G and $\overline{\lambda^G} = \mathbb{E}\left[\lambda^G\right]$ is the mean of λ^G. While this could be mathematically derived, in practice $\mathrm{Var}\left(\lambda^G\right)$ and $\overline{\lambda^G}$ are the sample variance and mean which are empirically found from a set of uniformly distributed sampled points (i.e., a simple random sample of λ^G).

The final log-normal distribution, $\lambda = \lambda(x,\, y), x \in [0,\, X], y \in [0,\, Y]$, is determined by assigning location (μ) and scale (σ) parameters to λ^S according to

$$\lambda(x,\, y) = \exp\left(\sigma\, \lambda^S(x,\, y) + \mu\right). \tag{2.11}$$

Fig. 2.2 shows the resulting SSLT demand density distribution, λ, generated from the λ^G fields displayed in Fig. 2.1 with default location and scale parameters (i.e., $\mu = 0$, $\sigma = 1$). By controlling the maximum angular frequency of the originating Gaussian random field, $\omega_{\max}$, and the log-normal location and scale parameters, the SSLT model can be used to simulate the demand characterization of a hypothetical SP service.

Lee et al. [28, 32] implement their proposed SSLT demand model as a discrete pixelated set of rectangular cells, the value of which indicates the overall demand located within that cell's region. Each demand point located within the SSLT area has an identical amount of demand associated with it. The value of each cell represents the number of homogeneous



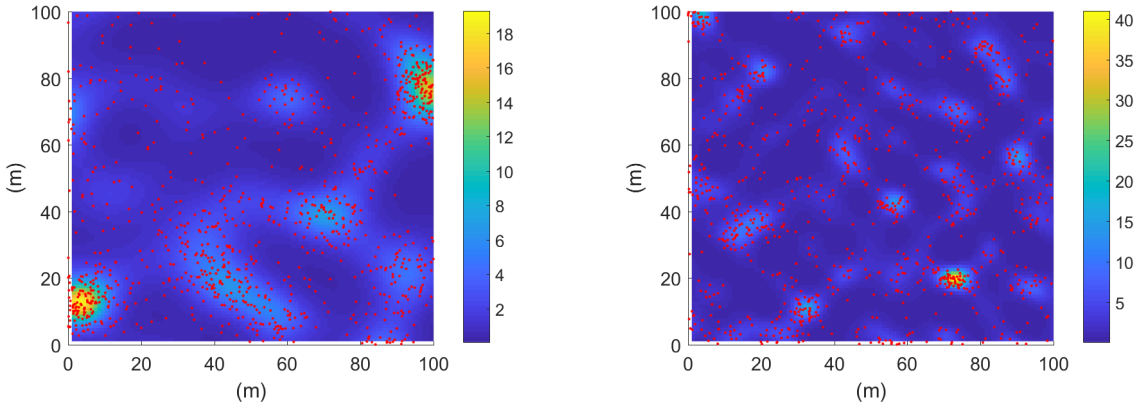
(a) $\omega_{\max} = \frac{2\pi}{30}$ (b) $\omega_{\max} = \frac{4\pi}{30}$

Figure 2.2: Two SSLT demand fields generated from the λ^G demand fields displayed in Fig. 2.1. $\mu = 0$, $\sigma = 1$

demand points located within that cell, and these demand points are uniformly distributed within that cell.

I deviate from their implementation by leaving the SSLT distribution as a continuous function representing the overall demand of the region, and only pixelate it for visualization purposes. To generate discrete demand points, I generate a non-stationary PPP using λ as the spatial intensity function as described by the accept-reject method (eqs. (2.1)–(2.3)) described previously. This allows the specific number of demand points to be controlled and accommodates for the assumption that the SSLT demand model is an overall distribution of demand points rather than each cell operating as independent PPP. Fig. 2.3 shows a realization of demand points distributed as a non-stationary PPP.

This SSLT model is used in **??** to generate SP service demand and end-user demand points while testing the approaches described in Chapter 3.



(a) $\omega_{\max} = \frac{2\pi}{30}$ (b) $\omega_{\max} = \frac{4\pi}{30}$

Figure 2.3: Example demand point realizations distributed according to the described non-stationary PPP with 400 demand points.

2.2 Stochastic Optimization

As presented in Section 2.1, the VNB present within the $\mathcal{R}$ must construct a set of VWNs to satisfy the needs of the services the various SPs seek to provide. Each service provider provides a characterization of its demand, such as the SSLT model presented in Section 2.1.1. With this information, the VNB must select the subset of virtual resources available in $\mathcal{S}$ that minimizes the cost to the VNB, and associate slices of these resources to VWNs that optimally satisfy the SPs' demand.

I formulate the presented problem of resource selection as Problem 1 (eqs. (2.13)–(2.19)), a two-stage stochastic optimization program. Let z_s, $s \in \mathcal{S}$, be a binary decision variable defined as

$$z_s \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if resource } s \text{ is selected to be sliced into a VWN;} \\ 0, & \text{otherwise.} \end{cases} \tag{2.12}$$

which establishes VNB resource selection. δ_{ms}, $m \in \mathcal{M}$, $s \in \mathcal{S}$, which is defined in Section 2.1 to represent the rate of resource s that is allocated to point m, is a second decision variable which establishes the slice of resource s that has been allocated to the VWN service that is associated with demand point m.

To balance the interest of maximizing demand satisfaction against minimizing cost, I introduce the positive real number α as a weighting coefficient between the two stages (eq. (2.15)). The SP indicates the desired amount of demand satisfaction necessary for the service the SP provides; α, as set by the VNB, realizes this degree of demand satisfaction of the constructed VWNs relative to their cost.

This doesn't differentiate between the various SPs of $\mathcal{N}$. Effectively, all SPs share a single α. This could be altered to do so Needs discussion.

The first stage objective function (eq. (2.13)) minimizes the total cost of the selected network with respect to that formed network's expected ability to satisfy the demand contained within $\mathcal{R}$. It characterizes this demand satisfaction as the expectation of $h(z, u)$, which is the optimal value of the second stage (eq. (2.15)) given a fixed z_s from the first stage. The optimal value of the second stage maximizes demand satisfaction by maximally slicing the BSs comprising the network selected by the first stage to the SPs' demands. The first stage handles the interaction between the RPs and the VNB where the VNB selects resources to use, and the second stage handles the interaction between the VNB and the SPs where the

Two-Stage Stochastic Optimization Program for BS Selection and Adaptive Slicing

$$\underset{\{z_s,\, s\in\mathcal{S}\}}{\text{minimize}} \left\{ \sum_{s\in\mathcal{S}} c_s\, z_s + \mathbb{E}\left[h\left(z,\, u\right)\right] \right\} \tag{2.13}$$

subject to:

$$z_s \in \{0,\, 1\}\,,\ \forall s \in \mathcal{S} \tag{2.14}$$

where $h(z, u)$ is the optimal value of the second-stage problem, which is given by:

$$\underset{\{\delta_{ms},\, m\in\mathcal{M},\, s\in\mathcal{S}\}}{\text{minimize}} \left\{ -\alpha \sum_{m\in\mathcal{M}} \sum_{s\in\mathcal{S}} \delta_{ms}\ \tilde{u}_{ms} \right\} \tag{2.15}$$

subject to:

$$z_s = \left[\sum_{m\in\mathcal{M}} \delta_{ms} > 0 \right],\ \forall s \in \mathcal{S} \tag{2.16}$$

$$\sum_{s\in\mathcal{S}} \delta_{ms}\ \tilde{u}_{ms} \leq d_m,\ \forall m \in \mathcal{M} \tag{2.17}$$

$$\sum_{m\in\mathcal{M}} \delta_{ms} \leq r_s,\ \forall s \in \mathcal{S} \tag{2.18}$$

$$\delta_{ms} \in [0,\, d_m]\,,\ \forall m \in \mathcal{M},\ \forall s \in \mathcal{S} \tag{2.19}$$

VNB slices the selected resources to the SPs' VWNs (i.e., interactions D and B as described in Section 1.2.2, respectively).

Equations (2.14) and (2.19) are constraints that implement the defined ranges of the decision variables z_s and δ_{ms}. Equation (2.17) is a constraint that reinforces eq. (2.19) and

asserts that a demand point is not overallocated by slicing it more resources than it demands. Similarly, eq. (2.18) ensures that a given BS is not over allocated to demand points.

Equation (2.16) ensures that demand is only allocated from available, selected resources. For this constraint, $[*]$ is the Iverson bracket, which is defined by

$$[*] \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if condition } * \text{ is true;} \\ \\ 0, & \text{otherwise.} \end{cases} \tag{2.20}$$

The stochastic optimization program (eqs. (2.13)–(2.19)) models the optimization that the VNB must perform to construct VWNs for and balance the needs of the various SPs. However, this program is not directly solvable, as mixed integer programming optimization tools cannot be used to solve stochastic optimization programs. Chapter 3 poses two approaches for solving this program.

Chapter 3

Approximation Approaches

Chapter 2 established a mathematical model for analyzing the relationship between SPs and a VNB and proposed a two-stage stochastic optimization program (eqs. (2.13)–(2.19)) for constructing VWNs in the context of that relationship. In this chapter, I present two approaches that arrive at a solution for this program. First, I modify the stochastic program to convert it into a deterministic form, the deterministic equivalent program (DEP), and sample it such that it is solvable using typical mixed integer programming optimization tools. From this, I further derive a model to adaptively slice BSs into new VWNs when resources are already selected. I then present a heuristic approach via genetic algorithm (GA) to handle resource selection with lower computational complexity.

3.1 Approach I: The Deterministic Equivalent Program

In order to directly solve the two-stage stochastic optimization program (eqs. (2.13)–(2.19)), it must be converted into a deterministic equivalent program (DEP). The DEP is equivalent to the original stochastic optimization program, but does not contain any stochastic variables (only deterministic variables) [21]. This is accomplished by converting stochastic variables into sets which contain every scenario or realization in the scope of the stochastic variables.

Let Ω be defined as the sample space (i.e., the set of all scenarios) of demand point locations defining $\tilde{u}_{ms}$. The probability a given scenario $\omega \in \Omega$ occurs is denoted by $p^{(\omega)}$, where $\sum_{\omega \in \Omega} p^{(\omega)} = 1$. Variables that are dependent on the scenario are shown with a superscript (ω) with the specific scenario it is dependent on indicated by ω. By making this change, the DEP (eqs. (3.1)–(3.5)) of the two-stage stochastic optimization program modeled in Section 2.2 is established.

The objective function (eq. (3.1)) combines both objective functions of the initial stochastic optimization program (eqs. (2.13) and (2.15)) into a single deterministic objective. The goal modeled by eq. (3.1) is unchanged from that modeled by eqs. (2.13) and (2.15); the first half handles resource selection by finding the minimal cost network relative to the second half, which handles adaptive slicing of those selected BSs by allocating slices of the BSs to demand for maximum demand satisfaction. These competing goals of minimizing VWN cost and maximizing VWN demand satisfaction are controlled by α.

Equation (3.2) is a constraint that ensures that demand is allocated to BSs that it is

Deterministic Equivalent Program (DEP) of Equations (2.13)–(2.19)

$$\underset{\left\{\substack{z_s,\,\delta_{ms}^{(\omega)},\\ m\in\mathcal{M},\,s\in\mathcal{S},\\ \omega\in\Omega}\right\}}{\text{minimize}} \left\{ \sum_{s\in\mathcal{S}} c_s\, z_s - \alpha \sum_{\omega\in\Omega} p^{(\omega)} \left(\sum_{m\in\mathcal{M}} \sum_{s\in\mathcal{S}} \delta_{ms}^{(\omega)}\, u_{ms}^{(\omega)} \right) \right\} \tag{3.1}$$

subject to:

$$\sum_{s\in\mathcal{S}} \delta_{ms}^{(\omega)}\, u_{ms}^{(\omega)} \leq d_m,\ \forall m \in \mathcal{M},\ \forall \omega \in \Omega \tag{3.2}$$

$$\sum_{m\in\mathcal{M}} \delta_{ms}^{(\omega)} \leq r_s\, z_s,\ \forall s \in \mathcal{S},\ \forall \omega \in \Omega \tag{3.3}$$

$$z_s \in \{0,\, 1\},\ \forall s \in \mathcal{S} \tag{3.4}$$

$$\delta_{ms}^{(\omega)} \in [0,\, d_m],\ \forall m \in \mathcal{M},\ \forall s \in \mathcal{S},\ \forall \omega \in \Omega \tag{3.5}$$

within range of and that it is not overallocated resources, adapting eq. (2.17) for all demand scenarios. Equation (3.3) is a constraint that ensures only selected BSs are allocated demand, and that BSs are allocated within capacity, combining and adapting eqs. (2.16) and (2.18) for all demand scenarios. Equations (3.4) and (3.5) define bounds on the decision variables according to eqs. (2.14) and (2.19).

As the DEP reformulation is equivalent to the original stochastic optimization program, the solution of the DEP is also a solution of the original. However, the stochastic variable of the original problem, $\tilde{u}_{ms}$, is dependent on the locations of the demand points in $\mathcal{M}$. As these demand point locations are stochastic and capable of assuming any of an uncountably infinite number of locations over $\mathcal{R}$, the set of deterministic scenarios, Ω, has an infinite

cardinality. This renders the DEP as also not directly solvable.

3.1.1 Sampling the DEP; Sample Average Approximation

In order to find a solution to the DEP reformulation, the set of considered scenarios (i.e., the sample space) must be finite. By generating O random scenarios, Ω is sampled into a finite set. Let $\hat{\Omega} \stackrel{\text{def}}{=} \{1, 2, \ldots, O\} \subset \Omega$ be this finite, discrete set containing O sampled scenarios of the sample space. Each scenario in $\hat{\Omega}$ is generated as a single, independent realization of demand points according to the demand intensity field λ_n (i.e., a non-stationary PPP with intensity function λ_n). By making this change, the sampled deterministic equivalent program (sDEP) (eqs. (3.6)–(3.10)), or sample average approximation (SAA), of the original stochastic optimization program is found.

The differences between the DEP and sDEP formulations are seemingly cosmetic. The objective (eq. (3.6)) and constraints (eqs. (3.7)–(3.10)) are effectively unchanged from the equivalent objective (eq. (3.1) and constraints (eqs. (3.2)–(3.5)) of the DEP. The distinction lies in that finding a solution of the sDEP optimizes for the specific scenarios in $\hat{\Omega}$. By sampling Ω to form $\hat{\Omega}$, the sDEP formulation introduces solution error as a tradeoff for manageability. The more accurately $\hat{\Omega}$ estimates Ω, the more accurate the optimal solution of the sDEP estimates the solution of the original stochastic program.

Sampled Deterministic Equivalent Program (sDEP) of Equations (2.13)–(2.19)

$$\underset{\left\{\substack{z_s,\, \delta_{ms}^{(\omega)},\\ m\in\mathcal{M},\, s\in\mathcal{S},\\ \omega\in\hat{\Omega}}\right\}}{\text{minimize}} \left\{ \sum_{s\in\mathcal{S}} c_s\, z_s - \alpha \sum_{\omega\in\hat{\Omega}} p^{(\omega)} \left(\sum_{m\in\mathcal{M}} \sum_{s\in\mathcal{S}} \delta_{ms}^{(\omega)}\, u_{ms}^{(\omega)} \right) \right\} \tag{3.6}$$

subject to:

$$\sum_{s\in\mathcal{S}} \delta_{ms}^{(\omega)}\, u_{ms}^{(\omega)} \leq d_m,\ \forall m \in \mathcal{M},\ \forall \omega \in \hat{\Omega} \tag{3.7}$$

$$\sum_{m\in\mathcal{M}} \delta_{ms}^{(\omega)} \leq r_s\, z_s,\ \forall s \in \mathcal{S},\ \forall \omega \in \hat{\Omega} \tag{3.8}$$

$$z_s \in \{0,\, 1\},\ \forall s \in \mathcal{S}. \tag{3.9}$$

$$\delta_{ms}^{(\omega)} \in [0,\, d_m],\ \forall m \in \mathcal{M},\ \forall s \in \mathcal{S},\ \forall \omega \in \hat{\Omega} \tag{3.10}$$

The Sample Average Approximation Estimator

It is apparent that $\hat{\Omega}$ approaches the whole sample space as O approaches infinity. For a sufficiently large O, $\hat{\Omega}$ contains enough scenarios to represent an arbitrarily tight approximation of Ω. However, as O increases, the computational complexity of the solution increases exponentially, causing the manageability of $\hat{\Omega}$ and the sDEP to decrease. While an arbitrary number of scenarios could be considered within $\hat{\Omega}$ to provide an overly-sufficient approximation, doing so would impose an unnecessary burden on the solution's computational complexity and increase computation time. It is therefore valuable to understand and find the minimum value of O that provides a sufficiently tight approximation of Ω. This can be done via an analysis of the SAA estimator

If I don't finish figuring out the SAA estimator, the following will be edited into the preceeding subsection.

Figure out the SAA estimator. Continue from here and fill out this section.

3.1.2 Adaptive Slicing

After the solution to the sDEP model has been found, the VNB has determined the joint BS selection that supports the VWNs. It further provides a proposed resource slicing of the BSs to the SPs' demand points, which determine the separate VWN constructions for those scenarios considered in $\hat{\Omega}$. As a condition for the sDEP model, $\hat{\Omega}$ is not infinite in scope. For an actual implementation of the sDEP within a VNB, any actual, observed scenario of demand points for which to build VWNs is certain to not be within $\hat{\Omega}$, and the sDEP model would not provide a slicing of the selected resources to construct a VWN for the new scenario.

Due to the computational complexity of the sDEP model, it is infeasible to find a new resource selection as the realization of demand changes. Further, assuming that the underlying demand characteristics do not change (i.e., that demand realizations still correlate to the demand intensity function, λ_n), it is unnecessary to determine a new joint resource selection as it is still optimally selected within a given confidence for the demands. Instead, it is valuable to have a model which assigns an adaptive slicing of the selected resources so the VWNs can adapt according to specific realizations of demand without needing to determine a new joint resource selection. This new adaptive slicing model (eqs. (3.11)–(3.14)) is found by fixing the selected resources, z_s, as a known constant and only considering the single,

specific demand realization $\omega \notin \hat{\Omega}$ to be sliced to and satisfied.

Adaptive Slicing Model

$$\underset{\{\delta_{ms},\, s\in\mathcal{S},\, m\in\mathcal{M}\}}{\text{maximize}} \left\{ \sum_{m\in\mathcal{M}} \sum_{s\in\mathcal{S}} \delta_{ms}\, u_{ms} \right\} \tag{3.11}$$

subject to:

$$\sum_{s\in\mathcal{S}} \delta_{ms}\, u_{ms} \leq d_m,\ \forall m \in \mathcal{M} \tag{3.12}$$

$$\sum_{m\in\mathcal{M}} \delta_{ms} \leq r_s\, z_s,\ \forall s \in \mathcal{S}. \tag{3.13}$$

$$\delta_{ms} \in [0,\, d_m]\,,\ \forall m \in \mathcal{M},\ \forall s \in \mathcal{S} \tag{3.14}$$

The objective function (eq. (3.11)) is a simplified version of the sDEP objective function (eq. (3.6)). As z_s is fixed to be a constant, the first block in the sDEP objective is constant for the adaptive slicing model and is removed as a simplification. Similarly, α no longer impacts the objective and is removed as a simplification. As only a single scenario is being considered, the summation across $\hat{\Omega}$ is trivial and $p^{(1)} = 1$, allowing both to be removed as a simplification. The constraints (eqs. (3.12)–(3.14)) are unchanged from the sDEP (eqs. (3.7), (3.8), and (3.10)), except that z_s is now a constant and without mention of $\hat{\Omega}$.

This new formulation is far more tractable than the sDEP formulation, as there is only a single decision variable, δ_{ms}, to solve and that the set of scenarios, $\hat{\Omega}$, simplifies to a singleton. Further, as δ_{ms} is not discrete, the adaptive slicing model is a linear programming problem and has better time order classification compared to the sDEP, which is a mixed integer

linear programming problem and is provably NP-Complete.

Effectively, the adaptive slicing model is a self-contained form of the second stage of the original stochastic program. With a given resource selection, this optimization program provides the slicing of the selected resources that optimizes the demand satisfaction in a specific realization of demand. Explicitly, it provides a determination in the VNB for how those resources are allocated to construct the VWNs for the SPs. Since this simplified model can provide the optimization for adaptive slicing and VWN specification, it allows for other approaches to the problem of just resource selection.

3.2 Approach II: The Genetic Algorithm

A major limitation of the approach in Section 3.1 is that the sDEP is increasingly unmanageable as O, S, or M increases. Most importantly, the accuracy of the sampled DEP is directly dependent on the size of $\hat{\Omega}$, O, directly causing a trade off between the accuracy of the sDEP and its computability in a reasonable amount of time. While diminishing returns can be avoided by determining the minimum necessary O for $\hat{\Omega}$ to provide a desireably tight estimation, the manageability of the sDEP is still dependent on the size of S and M. In this section, I reformulate the problem of joint BS selection for the VWN as a genetic algorithm (GA), circumventing the need to discretize demand or to establish $\hat{\Omega}$, thereby simplifying the original problem into a more scalable form. Further, by using the adaptive slicing model of Section 3.1.2, this GA approach can provide an approximate solution of the

original stochastic program while only needing to solve for resource selection.

A genetic algorithm is an iterative metaheuristic algorithm inspired by the concept of natural selection in which an approximate solution to a given optimization problem is arrived at via a series of progressive generations. Each generation contains a number of candidate solutions, called individuals, each of which is defined by a chromosome. During a given generation, a fitness heuristic is assessed for each individual based on its chromosome. Then individuals are *selected* at random to become parents, with more fit individuals being selected with higher probability. Parents are then paired off, and each pair of parents may, with probability p_{xov}, undergo a process called *crossover*, a process similar to genetic recombination, in which the parents' chromosomes are mixed to form two individuals (children) for the next generation. If crossover does not occur, the parents are cloned to be their own children for the next generation. The chromosomes of the resulting children then undergo *mutation*, altering the chromosome slightly. Once enough new children have been generated, this new set of individuals forms the next generation to repeat the process. Figure 3.1 is a block diagram that illustrates this general process as proposed for this approach. Each of these steps is described in further detail below.

3.2.1 The GA Chromosome

Let $\mathcal{G} \stackrel{\text{def}}{=} \{1, 2, \ldots, G\}$ be the set of generations used in the genetic algorithm and $\mathcal{I}_g \stackrel{\text{def}}{=} \{1, 2, \ldots, I\}$, $g \in \mathcal{G}$ be the set of individuals within generation g. Each individual $i \in \mathcal{I}_g$

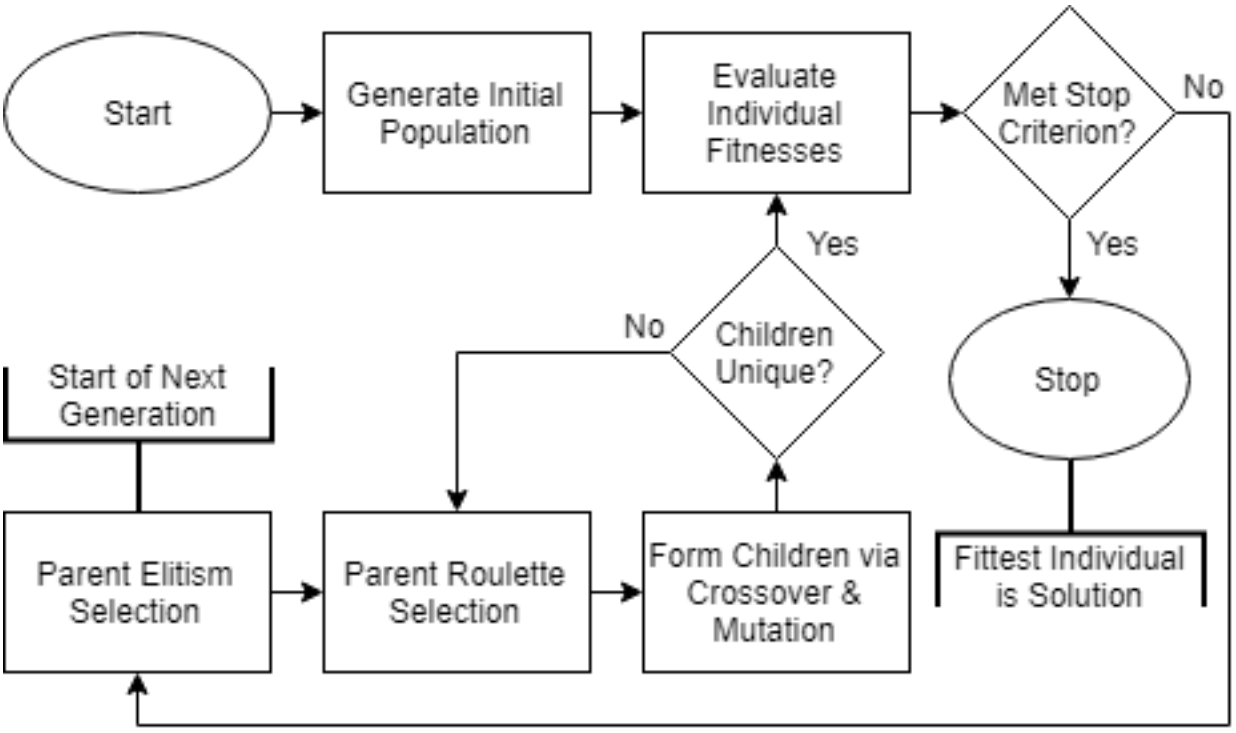


Figure 3.1: Block diagram of the GA approach. This is for a typical genetic algorithm utilizing elitism and enforcing generational uniqueness.

has a binary chromosome $z^{\{ig\}}$ of length S. $z_s^{\{ig\}}$, $s \in \mathcal{S}$, denoting each individual bit of the chromosome, is defined as

$$z_s^{\{ig\}} \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if BS } s \text{ is selected by the VNB for individual } i \text{ in generation } g; \\ 0, & \text{otherwise.} \end{cases} \tag{3.15}$$

Chromosome Fitness

To evaluate the fitness of each chromosome in the GA approach, I assume that all demand over $\mathcal{R}$ is allocated to the closest resource relative to that resource's coverage radius. That is, the demand allocated to a BS $s \in \mathcal{S}$ is all the demand located in the region V_s such that

$$V_s = \left\{ p \in \mathcal{R} \mid \frac{d\left(p,\, s\right)}{b_s} \leq \frac{d\left(p,\, t\right)}{b_t} \; \forall s \neq t \right\} \tag{3.16}$$

where $d\left(p,\, s\right)$ is the euclidean distance between an arbitrary point p and BS s. Let $\mathcal{S}' \subseteq \mathcal{S}$ be the set of resources that have been selected by an arbitrary chromosome (i.e., $z_s^{\{ig\}} = 1$, $\forall s \in \mathcal{S}'$ and $z_s^{\{ig\}} = 0$, $\forall s \notin \mathcal{S}'$ for an arbitrary chromosome $\{ig\}$). By making this assumption, $\mathcal{S}'$ partitions $\mathcal{R}$ into a multiplicatively weighted Voronoi tessellation using the point locations of the selected BSs as the defining sites and BS coverage radii, b_s, $s \in \mathcal{S}'$, as weights; if all BSs in $\mathcal{S}'$ have homogeneous coverage radii (i.e., $b_s = b$, $\forall s \in \mathcal{S}'$), this partition is a standard unweighted Voronoi tessellation.

The region within the coverage radius of BS s is denoted by B_s. It is desired that V_s be wholly contained within B_s (i.e., $V_s \subseteq B_s$), but this is not guaranteed by the Voronoi tessellation assumption. If V_s is not wholly contained within B_s, BS s is considered to be *overcoverage*. The total demand allocated to a selected resource $s \in \mathcal{S}' \subseteq \mathcal{S}$ is $\sum_{i=1}^{N} \iint_{V_s} \lambda_i\left(x,\, y\right)\, dx\, dy$. If the total demand allocated to BS s exceeds B_s, BS s is considered to be *overcapacity*.

The fitness heuristic of each individual chromosome, $z^{\{ig\}}$, is assessed as the reciprocal

of the chromosome's cost

$$\text{fitness}\left(z^{\{ig\}}\right) = \frac{1}{\text{cost}\left(z^{\{ig\}}\right)} \tag{3.17}$$

The cost of a given chromosome is the sum of the direct (OpEx) costs of the individual selected BSs and their associated overcoverage (c_{cov}) and overcapacity (c_{cap}) costs

$$\begin{aligned}\text{cost}\left(z^{\{ig\}}\right) = \sum_{s\in\mathcal{S}}\Bigg(& c_s\ z_s^{\{ig\}} + c_{\text{cov}}\ \left[V_s \not\subseteq B_s\right] + \\ & \left(c_{\text{cap}}^{g} - 1\right)\ \max\left(0, \sum_{i=1}^{N}\left(\iint_{V_s}\lambda_i\left(x,\ y\right)\ dx\,dy\right) - r_s\right)\Bigg)\end{aligned} \tag{3.18}$$

where $[*]$ is the Iverson bracket (eq. (2.20)).

Resources being overcoverage is not desired; solutions that assume some demand is unallocatable by being out of range should be consider less-than-feasible or completely infeasible from the outset. In contrast, the overcapacity cost starts at zero and grows exponentially with each successive generation. For early generations, this allows for inefficient or infeasible solutions to temporarily exist to seed later generations and improve generational diversity. This increases the probability that the final solution (a local maxima) is the global maximum or closer to the global maximum of the fitness function.

3.2.2 Forming the Next Generation

Once the fitness calculation of all of the individuals of the current generation has been performed, the next generation begins to form.

Elitism and Selection

Generating the next generation begins with selection. First, I use elitism selection. Under elitism selection, a defined number of the fittest individuals from each generation is selected as that generation's elitist members. These elitist members, the n fittest individuals of the generation, are cloned into the next generation without undergoing the genetic operators of crossover or mutation. Through elitism, I ensure that the fittest individual of a given generation is not less fit than the previous generation's fittest individual. The only exception comes from if the fittest individual's fitness decreases from inefficiency cost, but these are eventually removed as the overcapcity cost increases.

After elitism, standard selection occurs using roulette-wheel selection. In roulette-wheel selection, the fitness of each individual to normalized to the total fitness of the generation and organized into a list. From this list, a new list of the cumulative fitnesses is generated such that the cumulative fitness of individual i is $\sum_{j=1}^{i} \text{fitness}\left(z^{\{jg\}}\right)$; the cumulative fitness of individual I (i.e., the last individual in the generation) is thereby 1. A random number, $P \sim \mathcal{U}(0,\, 1)$, is then generated. The selected individual (or parent) is the first individual in the list that has a cumulative fitness value larger than P.

Under roulette-wheel selection, parents are each randomly selected with a probability given by

$$\mathrm{P}\left(\text{individual } i \text{ of gen. } g \text{ is selected for gen. } g+1\right) = \frac{\text{fitness}\left(z^{\{ig\}}\right)}{\sum_{i\in\mathcal{I}} \text{fitness}\left(z^{\{ig\}}\right)} \tag{3.19}$$

Crossover

Once selection has occurred, the individuals selected via roulette-wheel selection are paired off. With a probability of p_{xov}, the pair undergoes crossover to form their two children which go on to the next generation; otherwise, the parents are cloned as their own children. During crossover, the chromosomes of each pair of parents are mixed. Typical forms of crossover include single-point, k-point, and uniform crossover, though there are other more exotic forms that are application specific.

I use uniform crossover as it has been suggested [33] to be one of the best crossover operators in terms of general performance. With uniform crossover, each bit in the chromosomes of the children are independently chosen from the parents' genomes with equal probability (i.e., 50%). Compared to single-point crossover, uniform crossover is also suggested to increase exploration of the search space as it severs the positional bias of the bits; bits with possible spatial significance to one another no longer undergo crossover in segments like they would in single or k-point crossover.

Needs citation?

Mutation

Once each child has been generated, each bit within the child's genome has an independent chance, p_{mut}, to mutate. When a bit mutates, the bit flips from 1 to 0, or from 0 to 1. Typically, the chance of each bit to mutate is $p_{\text{mut}} = \frac{1}{l}$, where l is the length of the chromosome, which provides an average of one bit mutation per child. For $z^{\{ig\}}$, $p_{\text{mut}} = \frac{1}{S}$.

Uniqueness

In a traditional genetic algorithm, once mutation has been completed, the next generation of children has been completely generated. However, it is possible for several children to be identical, reducing the overall genetic diversity of the generation and decreasing the algorithm's exploration of the search space. To counter this possibility I enforce the uniqueness condition on each generation. That is the generation must be comprised of unique individuals. If there are any individuals that are identical to others in the generation, all but one of the identical individuals are discarded from the generation. The vacancies are then filled by a new set of generated children

3.2.3 Stopping the GA

The GA iterates for a number of generations G. If the GA settles on a single individual as the fittest individual for a number of continuous generations, G_{ind}, it will halt and present that individual's chromosome as the final approximate solution for z_s. Further, if the fitness

of the fittest individual of a number of continuous generations, $G_{\text{fit}} \leq G_{\text{ind}}$, the GA also halts and presents that fittest individual's chromosome as the final approximate solution for z_s. For both of these halting conditions, a minimum number of generations, $G_{\min}$, must first pass. Otherwise, the chromosome of the fittest individual of generation G determines z_s.

In each case, the chromosome of the fittest individual when the GA halts is an approximate solution for the VNB's resource selection, z_s. As the GA only handles the resource selection, it only represents a solution of the first stage (eqs. (2.13) and (2.14)) of the original stochastic optimization program (eqs. (2.13)–(2.19)). That is, while joint resource selection has occurred, and that the process was informed by the demand to be satisfied (i.e., λ_n), the selected resources still need to be sliced to determine the allocations of those resources to specific VWNs. This process of solving the second stage of the original optimization program can be adaptively handled to a hypothetical or real set of demand points using the adaptive slicing model (eqs. (3.11)–(3.14)) of Section 3.1.2.

Chapter 4

Simulations and Results

4.1 Setup

In this section, we evaluate the sampled DEP and genetic algorithm approaches as approximations of Problem 1. We will compare the cost, demand satisfaction, and time to generate of the resultant networks.

Unless stated otherwise, we use the default parameter values shown in Table 4.1. BS locations are determined as a stationary PPP. Demand point locations are generated independently for each scenario as a non-stationary PPP using $\rho(x, y)$, $x \in [0, X]$, $y \in [0, Y]$, as the spatial intensity function, as described in Section **??**. Fig. **??** provides a visualization of the simulation network area. (**??**) shows the BS locations of $\mathcal{S}$ with the associated Voronoi Tessellation showing the coverage areas of the BSs when all are active with respect to the

Table 4.1: Numerical Values of Relevant Parameters

Width, Height of Geographic Area (X)	2 km x 2 km
Number of BSs (S)	60
Number of Demand Points (M)	75
Number of Sampled Scenarios (O)	$\{5, 10, \ldots, 50\}$
BS cost ($c_s, \forall s \in \mathcal{S}$)	1
BS capacity ($r_s, \forall s \in \mathcal{S}$)	1.50 Mbps
BS range ($b_s, \forall s \in \mathcal{S}$)	500 m
Point Traffic Demand ($d_m, \forall m \in \mathcal{M}$)	0.178 Mbps
Two-Stage Model Weight (α)	20
SSLT Approximation Depth (L)	50
SSLT Maximum Angular Frequency ($\omega_{\max}$)	$\frac{2\,\pi}{30}$
SSLT Location Parameter (σ)	0
SSLT Scale Parameter (μ)	1
Pixel Grid Size	100 x 100, 20 m
Maximum Number of Generations (G)	3000
Minimum Number of Generations	300
Unchanged Generations Before Halt (G_{halt})	150
Number of Individuals per Generation (I)	80
Number of Elite Individuals per Generation	4
Probability of Crossover (p_{xov})	0.7
Probability of Mutation per bit (p_{mut})	$\frac{1}{S} = 0.0167$
Overcoverage Cost (c_{cov})	3
Overcapacity Cost (c_{cap})	1.015

genetic algorithm. (**??**) shows the SSLT demand density field with one example scenario of demand points, which acts as a single sample of the demand density field. To compute $u_{ms}^{(\omega)}$, it is assumed that there is perfect propagation between the demand points and BSs. Unlike as described in Section **??**, $u_{ms}^{(\omega)} = 1$ if the distance between demand point $m \in \mathcal{M}$ of scenario $\omega \in \hat{\Omega}$ and BS $s \in \mathcal{S}$ is less than b_s, and 0 otherwise. To compute the integral of the fitness function (3.18), $\rho\,(x,\, y)$ is discretized into a grid of congruent pixels, and the demands of all pixels within a Voronoi cell of interest are summed together.

We ran our simulations on an Intel Core i7-4790K 4.00 GHz 4 real/8 virtual core CPU with 16 GB of DDR3 RAM. We used CPLEX [34] to solve the sampled DEP optimization problems and we used MATLAB to simulate the genetic algorithm and to generate the demand field and stochastic data (i.e., $\rho\,(x,\, y)$ and $u_{ms}^{(\omega)}$). During the simulations, extraneous processes were culled to allow maximal use of computer resources. Average values for the performance of the genetic algorithm are provided from 50 independent runs using the identical data set except for the set of initial individuals. The sampled DEP solutions were solved across multiple values of α as cost, time, and demand satisfaction are directly dependent on α.

In Fig. **??** is a comparison of the approach run times. It shows the run times of both the DEP and genetic algorithm in terms of the speedup ratio and in absolute number of seconds. As the number of scenarios sampled increases, the CPU run time of the sampled DEP increases exponentionally, and fails to converge to a final solution with 50 scenarios within the time limit of 15 minutes. Except for the 5 scenario case, the genetic algorithm converges to a solution in less CPU time (125.8 seconds) than the sampled DEP. When

20 scenarios are sampled, the DEP takes approximately 440% of the time as the genetic algorithm; at 45 scenarios, this has increased to 4,630%. CPLEX is capable of parallelizing across the 8 CPU cores, allowing for the wall clock run time to be, at minimum, one-eighth the CPU run time. In terms of the adjusted approximate "wall clock" time, the genetic algorithm converges in less wall clock time than the sampled DEP for 25 or more sampled scenarios.

The trade off for the genetic algorithm's improved run time is that the solution provided is less optimal than the sampled DEP, as indicated by an increased cost for the VNB to build the VWN. Fig. **??** compares the increasing cost of the sampled DEP as α increases with the cost of the various genetic algorithm solutions. On average, the genetic algorithm incurs a 36% increased cost in selecting the BSs for the VWN. At minimum, the incurred cost is only 20% than the sampled DEP, which implies the genetic algorithm might be terminating early, and a tighter solution might be found by increasing G_{halt}. It should also be noted that one unit of cost is one additional BS being selected for the VWN, and the sampled DEP selections for $\alpha \geq 30$ have a cost of only 10 BSs. Any variance that incurs one additional BS for the genetic algorithm incurs 10% increased cost. Increasing the number of BSs required to comprise the VWN would introduce additional granularity in $\mathcal{S}$ that might decrease the inefficiency of the genetic algorithm. This was not done as this data set was chosen specifically so the sampled DEP would terminate within 15 minutes (i.e., in a reasonable amount of time); increasing the number of BSs available to the VNB drastically increases the time it would take the sampled DEP to converge to a solution.

There is a direct correlation between the number of BSs in the VWN and its capability for satisfying demand. As the cost - and, therefore, the number of BSs - increases for a given solution, the average demand satisfaction trends towards 100%, as shown in Fig. **??**. Because of this overallocation of resources, the genetic algorithm solutions have a very high demand satisfaction, averaging 99.9% satisfaction and reaching 100% for some solutions. The worst performing genetic algorithm solution exceeded 99.1% demand satisfaction. The most expensive 10-BS sampled DEP solutions reach 99.2% when slicing to the same set of demand point scenarios, Ω, that determined the sampled DEP BS selection.

When the set of demand points change to a scenario no longer in Ω, the sampled DEP performs very similarly. Fig. **??** shows the demand satisfaction for both the sampled DEP and genetic algorithm BS selections when sliced to a new set of scenarios as evaluated with Problem 3. Here, the number of demand points increases to 200 points per scenario, each with 66.8 kbps rate demand, across 50 independent scenarios. The demand satisfaction trend of the sampled DEP BS selection follows very closely to the original set of scenarios but hits a maximum of 99.0% demand satisfaction with 10 BSs. In comparison, the SP demand point scenarios are far more beneficial to the genetic algorithm, which reaches greater than 99.99% demand satisfaction for all generated VWNs. This is expected as a side effect of the increased number of points and scenarios more accurately describing a sampling of the original SSLT demand density field of the SP, ρ.

In this chapter I will be introducing four different cases to test the provided approximation

Start this chapter ASAP. Start running Case I data now.

approaches. The first will be the test case used in my conference paper (one SP, with homogeneous resources). The second will be an expansion of the test case used in my conference paper, but with heterogeneous resources. The third will extend to service multiple similar cellular SPs. The fourth will extend to a case with multiple SPs with various, specialized demands.

4.2 VWN Construction for a Single SP

Lead into the first two cases, which test the approaches while using a single SP.

4.2.1 Case I: Homogeneous Urban Cellular Network

Start this!

Basically as presented in my conference paper. One SP, homogeneous resources within the RPs. Might need to use a new data set, though, with a larger data set.

4.2.2 Case II: Impact of Heterogeneous Resources

Same as Case I, but with heterogeneous resources within the RPs. Need to understand how this changes the approaches.

4.3 VWN Construction for Multiple SPs

Lead into the second two cases (should I have more?), which test using multiple SPs to satisfy from the same set of resources.

4.3.1 Case III: Two Similar Urban Cellular Networks

First consider a case with two SPs with similar demands. Overlapping cellular networks. Could see how the approaches behave while two SPs partially overlap.

Homogeneous Resources

If it appears that the difference between Case I and Case II (sections 4.2.1 and 4.2.2) is worth further consideration, then analyze here with homogeneous resources. Otherwise, a single comparison should be sufficient.

Heterogeneous Resources

As for the previous subsubsection (4.3.1), but consider with heterogeneous resources.

4.3.2 Case IV: SPs with Specialized Demands

This is the major case that is the extension of my work. Case I (4.2.1) analyzed what happens with a single SP, Case II (4.2.2) expanded that to heterogeneous resources, and

Case III (4.3.1) added an additional similar SP, but Case IV considers when there are several SPs and with their own considerations and unique demands. Need to consider what these SPs look like. One would be a cellular network like in Case I (moderate to high number of users, moderate demand). Another could be a streaming service (few users, high individual demand). Another an emergency service (very low number of users and demand, but requiring virtually 100% demand satisfaction - see note below). What other SPs should I consider?

***Note**: I need to consider how to accurately label demand satisfaction within the approaches. In effect, this would be controlled by α for the (sampled) DEP and controlled by β or some such for the genetic algorithm. I should investigate this at some point of the thesis, probably within their appropriate sections in chapter 3 (DEP: 3.1 and GA: 3.2).*

Homogeneous Resources

As for Case III (4.3.1), if a considerable difference was detected between Cases I and II (4.2.1 and 4.2.2), consider analyzing the case with homogeneous resources and

Heterogeneous Resources

also with heterogeneous resources.

Chapter 5

Conclusions

Consider conclusions of my work. I don't think this chapter would be long, but condense my findings into some coherent thoughts, and redirect to what they are. Also expound on some of the further work that my research could be expanded to (e.g., further use cases investigating my approaches, use of (meta)heuristics other than a genetic algorithm to approximate the optimization problem, improve the basic capacity function used in my optimization model).

Old work Update and Remove!

5.1 Considerations for Future Work

Filled as ideas come to mind during writing

– Other traffic demand models (log-normal mixtures, α-stable)[1]

See note. Integrate.

[1] In general, network traffic and the resource deployments that they are meant to model tend to exhibit characteristics of heavy-tailed spatial distributions. It has been shown that traffic distributions in cellular networks can be more accurately approximated as a mixture of log-normal distributions [28, 29]; Lee et

– Rate normalization that varies with distance (i.e., non-binary $\tilde{u}_{ms}$)

– Other heuristic approaches (e.g., particle swarm optimization, simulated annealing, deep learning approaches)

– Altering α to vary according to the demand points of the various SPs, allowing for each SP to independently control the desired demand satisfaction of their network.

β already works this way for the GA implementation, as it applies a constant to $\lambda(x, y)$.

– Handling varying values of b_s within the GA

– Improve the GA implementation such that it uses some form of integration of $\lambda(x, y)$

al. [28] describe that nonmixtures capture the distribution for a single moment while mixtures capture the distribution as it changes with time and place. Similarly, Zhou et al. [30] have shown that distributions of deployed BSs can be accurately approximated as a α-stable distribution. The latter case might be explained through the central limit theorem, as the sum of power-law distributed (Paretian tailed) random variables, such as those describing traffic in telephone networks [35], will tend towards an α-stable distribution. I use neither in this thesis as α-stable distributions do not generally have a closed-form probability density function (PDF), rendering use non-trivial, and log-normal mixtures require tuning multiple parameters, adding additional complexity.

Bibliography

[1] Cisco, "Cisco visual networking index: Forecast and methodology, 2016-2021," June 2017, white paper at Cisco.com. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf

[2] C. Beckman and G. Smith, "Shared networks: making wireless communication affordable," *IEEE Wireless Communications Magazine*, vol. 12, no. 2, pp. 78–85, Apr 2005.

[3] M. J. Abdel-Rahman, K. Cardoso, A. B. MacKenzie, and L. A. DaSilva, "Dimensioning virtualized wireless access networks from a common pool of resources," in *Proceedings of the IEEE CCNC Conference*, Jan 2016, pp. 1049–1054.

[4] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[5] V. Chandrasekhar, J. G. Andrews, and A. Gatherer, "Femtocell networks: a survey," *IEEE Communications Magazine*, vol. 46, no. 9, pp. 59–67, Sept 2008.

[6] H. S. Dhillon, R. K. Ganti, F. Baccelli, and J. G. Andrews, "Modeling and analysis of k-tier downlink heterogeneous cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 3, pp. 550–560, Apr 2012.

[7] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590–3600, Nov 2010.

[8] 3GPP TS 23.251, "Network sharing; architecture and functional description," v. 14.1.0, Sept 2017.

[9] X. Costa-Perez, J. Swetina, T. Guo, R. Mahindra, and S. Rangarajan, "Radio access network virtualization for future mobile carrier networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 27–35, July 2013.

[10] "Mobile network sharing report 2010-2015 - developments, analysis & forecasts," Visiongain, Tech. Rep., 2010.

[11] J. S. Panchal, R. D. Yates, and M. M. Buddhikot, "Mobile network resource sharing options: Performance comparisons," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4470–4482, Sept 2013.

[12] L. Doyle, J. Kibida, T. K. Forde, and L. DaSilva, "Spectrum without bounds, networks without borders," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 351–365, Mar 2014.

[13] K. V. Cardoso, M. J. Abdel-Rahman, A. B. MacKenzie, and L. A. DaSilva, "Virtualization and programmability in mobile wireless networks: Architecture and resource management," in *Proceedings of the Workshop on Mobile Edge Communications (MECOMM'17)*, 2017, pp. 1–6.

[14] M. Gomez, M. B. Weiss, G. McHenry, and L. Doyle, "Matching markets for spectrum sharing," in *Telecommunications Policy Research Conference*, Sept 2017. [Online]. Available: http://d-scholarship.pitt.edu/33631/

[15] M. Gomez, "Secondary spectrum markets: from "naked" spectrum to virtualized commodities," Ph.D. dissertation, Sept 2017. [Online]. Available: http://d-scholarship.pitt.edu/33130/

[16] M. J. Abdel-Rahman and M. Krunz, "Stochastic guard-band-aware channel assignment with bonding and aggregation for DSA networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 7, pp. 3888–3898, July 2015.

[17] M. J. Abdel-Rahman, M. AbdelRaheem, A. B. MacKenzie, K. Cardoso, and M. Krunz, "On the orchestration of robust virtual LTE-U networks from hybrid half/full-duplex Wi-Fi APs," in *Proceedings of the IEEE WCNC Conference*, Apr 2016.

[18] M. J. Abdel-Rahman, M. AbdelRaheem, and A. B. MacKenzie, "Stochastic resource allocation in opportunistic LTE-A networks with heterogeneous self-interference cancellation capabilities," in *Proceedings of the IEEE DySPAN Conference*, Sept/Oct 2015, pp. 200–208.

[19] N. Y. Soltani, S. J. Kim, and G. B. Giannakis, "Chance-constrained optimization of OFDMA cognitive radio uplinks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 3, pp. 1098–1107, Mar 2013.

[20] R. Atawia, H. Abou-zeid, H. S. Hassanein, and A. Noureldin, "Joint chance-constrained predictive resource allocation for energy-efficient video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1389–1404, May 2016.

[21] P. Kall and S. W. Wallace, *Stochastic Programming*. John Wiley and Sons, 1994.

[22] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, Eds., 1972, pp. 85–103. [Online]. Available: https://doi.org/10.1007/978-1-4684-2001-2_9

[23] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 3rd ed. Springer Publishing Company, Incorporated, 2000.

[24] T. Cui, R. Bai, A. J. Parkes, F. He, R. Qu, and J. Li, "A hybrid genetic algorithm for a two-stage stochastic portfolio optimization with uncertain asset prices," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, May 2015, pp. 2518–2525.

[25] Department of Mining and Mineral Engineering, "Poisson: A program for spatial point generation using poisson processes," Sept 2002. [Online]. Available: http://www.leeds.ac.uk/StochasticRockFractures/Download/ReportsSeminars/Report1_POISSON_Prog.pdf

[26] F. Aurenhammer, "Voronoi diagrams — a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, Sep. 1991. [Online]. Available: http://doi.acm.org/10.1145/116873.116880

[27] U. Gotzner and R. Rathgeber, "Spatial traffic distribution in cellular networks," in *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, vol. 3, May 1998, pp. 1994–1998.

[28] D. Lee, S. Zhou, X. Zhong, Z. Niu, X. Zhou, and H. Zhang, "Spatial modeling of the traffic density in cellular networks," *IEEE Wireless Communications*, vol. 21, no. 1, pp. 80–88, Feb 2014.

[29] M. Michalopoulou, J. Riihijrvi, and P. Mhnen, "Towards characterizing primary usage in cellular networks: A traffic-based study," in *Proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, May 2011, pp. 652–655.

[30] Y. Zhou, R. Li, Z. Zhao, X. Zhou, and H. Zhang, "On the α-stable distribution of base stations in cellular networks," *IEEE Communications Letters*, vol. 19, no. 10, pp. 1750–1753, Oct 2015.

[31] J. Reades, F. Calabrese, and C. Ratti, "Eigenplaces: analysing cities using the space - time structure of the mobile phone network," *Environment and Planning B: Planning and Design*, vol. 36, pp. 824–836, 2009.

[32] D. Lee, S. Zhou, and Z. Niu, "Spatial modeling of scalable spatially-correlated log-normal distributed traffic inhomogeneity and energy-efficient network planning," in *Pro-*

ceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Apr 2013, pp. 1285–1290.

[33] S. Picek, M. Golub, and D. Jakobovic, "Evaluation of crossover operator performance in genetic algorithms with binary representation," in *Proceedings of the 7th International Conference on Intelligent Computing: Bio-inspired Computing and Applications*, ser. ICIC'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 223–230. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-24553-4_31

[34] IBM, "Optimization model development toolkit for mathematical and constraint programming (CPLEX)," 2012. [Online]. Available: http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud

[35] Y. Xia, C. K. Tse, W. M. Tam, F. C. M. Lau, and M. Small, "Scale-free user-network approach to telephone network traffic analysis," *Phys. Rev. E*, vol. 72, p. 026116, Aug 2005. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.72.026116