MathWorks Math Modeling Challenge 2021

Nueva School

Team # 14656 Hillsborough, California

Coach: Ted Theodosopoulos

Students: Elliot Chin, Ryan Cheng, Sebastian Deery, Joseph Kraus, Nikhil Thakur

M3 Challenge Technical Computing Award WINNER—$3,000 Team Prize

JUDGES' COMMENTS

Specifically for Team # 14656 —Submitted at the close of Technical Computing Contention Judging:

COMMENT 1: This paper effectively used technical computing for all three parts of the question. Parts 1 and 3 were especially creative. In Part 1, an agent based model was chosen to predict ISP investment in different cities or regions over time, and MATLAB code was used to implement that model. In Part 3, the students came up with a clever approach for loading map data into their program, which used an optimization method based on simulated annealing to place cellular nodes. We liked that the students took advantage of the ability to easily change parameters in their code to run sensitivity analysis where appropriate. Technical computing was also used to create effective and compelling visuals in the report. One area for improvement is the code itself, which lacked sufficient commenting and clear structure. This was partially made up for by effective explanation and justification of the methods used in the text of the report. Overall, a phenomenal paper that the judges were very happy to select as our winner!

Specifically for Team # 14656 —Submitted at the close of Technical Computing Triage Judging:

COMMENT 1: This is a strong paper. The first model and its discussion is very thorough. The section where the MC model is put to test against the regression model is particularly insightful and convincing. The assumptions of the second model are thoughtful, which seems to have helped make the second model produce a good result. More specifics on the third model are needed; without them it becomes difficult to interpret the data presented.

COMMENT 2: There was a good start here (I especially liked the use of Monte Carlo simulations), but you ran out of steam writing this up. The strengths/weaknesses of the model are important. Perhaps you could use boxes or offset equations in the future to make the important results more evident.

COMMENT 3: Your team paper was very clear and well organized and you made very good modelling efforts! Your background research and organization of assumptions was also particularly good.

COMMENT 4: The paper would have benefited from further verification of data and results, if you had more time. This might, for example, have provided further insight into the long term effect of the buffer cost model on the predicted costs in Q1, and allowed you to flag, in the answer to Q3, that 102 hours a week was a lot and therefore led to modelling corrections.

COMMENT 5: The consideration of demographic information and income in Q3 is commendable.

COMMENT 6: Well written.

COMMENT 7: Be mindful that MB is different than Mb.

The Future of Global Interconnection

Executive Summary

Access to the internet spurs economic growth, educational development, and recreational enjoyment, making it a crucial consideration when evaluating and predicting global development, equality, and equity. Connectivity is a progressively important part of a changing world, as high-speed access is now crucial for education, work, and socialization. Furthermore, the nature of connectivity itself is changing: broadband costs are dropping while emerging wireless technologies stake their claim in the market. As new technologies such as 5G emerge, and the coronavirus pandemic irreversibly changes our digital habits, it becomes increasingly important to evaluate the cost, requirements, and distribution of internet access.

We first predicted the price of bandwidth in the United States and in the United Kingdom in dollars per Mbps and pounds per Mbps using an agent based Monte Carlo simulation, in which each ISP is a profit maximizing entity that can either enter or leave a given region. Both the U.S. and the U.K. were split into smaller subregions with different populations, initial prices, and initial competitors, with future prices being calculated as a function of the aforementioned variables. Each internet service provider (ISP) has a chance of entering each new market, which is dependent on the amount of current competitors in the market, the size of the population, and the potential for profits. This random chance varies per ISP, modeling differences such as market capitalization and access to capital. Once an ISP enters a new region, the region has more competition, and the ISPs in it have an incentive to lower the price. In our model, when an ISP enters a new region, the price in that region is lowered to a variable cost plus some fixed buffer cost. This process is repeated across all ISPs, regions, and years. Our model predicts that in 2031, U.K. prices will be $0.2287 per Mbps while regional prices in the U.S. will be $.293 per Mbps on average.

To model the minimum bandwidth required for different individuals, we created a model that creates random profiles with a variety of characteristics given a set of input parameters. Then, given each profile, the model simulates hour by hour internet usage and determines the minimum amount of internet bandwidth required for each profile by iteratively testing different bandwidth speeds. We then applied this model to three specific profiles and found the specific requirements that would cover their needs 90 percent of the time and 99 percent of the time. For the first profile, we found they need 105 Mbps and 179 Mbps for 90 percent and 99 percent, respectively. For the second profile, we found that they need 71 Mbps and 129 Mbps. For the third profile, we found that they need 123 Mbps and 219 Mbps.

To optimize the distribution of cellular towers throughout hypothetical regions, we first calculated the bandwidth needs of each region based on demographic data such as age and socio-economic status. Given this value, we then iteratively tested different cell tower distributions, varying amount, placement, and strength, optimizing with simulated annealing to minimize the number of nodes and net bandwidth unaccounted for in the region. Our model predicts that region A needs a single medium capacity node, region B requires three medium nodes, and region C requires a single high capacity node.

1 Introduction

1.1 Restatement of the Problem The problem that we are asked to model is as follows:

1. Model the cost of bandwidth in dollars or pounds per megabit per second over the next 10 years for consumers in the United States and the United Kingdom.

2. Create a model to predict a given household's need for internet over the course of a year. Apply this model to different households to determine the minimum amount of bandwidth required that would cover their needs for 90 percent of the time and the minimum amount of bandwidth that would cover their needs for 99 percent of the time.

3. Create a model that produces an optimal distribution of cell towers in a given region, taking into account the demographic data of the region and the region's bandwidth needs. Apply this model to three different regions.

2 Global Assumptions

G-1. *There are six types of broadband connections: Digital Subscriber Line (DSL), Cable, Fiber, Wireless (4G/5G), Satellite, and Broadband over Powerlines (BPL).*[1] Bandwidth for consumers traditionally refers to residential internet connection, which includes all types of connection except wireless; published data is consistent with this assumption.[2] Mobile broadband, on the other hand, is wireless.[3]

G-2. *Unknown random variables are normally distributed.* Normal distributions effectively model many phenomena and are often the default distribution statisticians turn to[41]. When possible, we further refine our models and explain the exact assumptions behind a specific distribution. However, this global assumption provides important context for the paper.

3 Part I: The Cost of Connectivity

3.1 Assumptions

3-1. *U.S. bandwidth pricing is defined as the median price divided by the median speed.* Published sources utilize median measurements as a norm for classifying bandwidth prices.[2,4] Dividing median price by median speed is imprecise but still allows an accurate picture of bandwidth pricing, as plan price generally correlates with plan speed.

3-2. *U.K. bandwidth pricing is defined as the most recent Superfast price divided by the average speed.* Superfast connections are available to over 96 percent of the U.K., making them the norm for data pricing.[5]

3-3. *Select cities can be used as a model for nationwide U.S. broadband prices.* Broadband prices vary significantly by location across the U.S. We select a diverse selection of cities approved and studied by the Open Technology Institute.[6] Pilot cities are often used in diverse fields of research.[7,8]

3-4. *The main variable component of bandwidth pricing is internet transit cost.* Transit cost is the cost per megabit per second (Mbps) of a connection to the internet for an ISP.[9] This transit is nonresidential and a cost for ISPs; it is always lower than the cost of connection sold by an ISP. There are other costs associated with bandwidth pricing such as infrastructure and salaries; however, these fluctuate depending partly on the market, are unpredictable over time, and are modeled as such.[13-15]

3-5. *A uniform proportion of households across urban areas are connected to the internet, and households with a greater number of people require greater internet capacity.* Internet access is widely available, and disconnected households are usually disconnected by choice (or in some cases, for financial reasons), rather than because of geography.[17] Individual people, rather than groups of people, access the internet.[19]

3-6. *Total profits in a region are proportional to the total population multiplied by the profit per Mbps sold.* While exact profits are not calculated in this model, relative profits compare possible profit between different region. Total profit can be calculated by per-unit-profit multiplied by expected units sold.[20]

3-7. *Internet Service Providers (ISPs) prioritize profit.* ISPs are public companies and are obligated to produce profits for their shareholders.[21]

3-8. *ISPs have limited capital.* No company has unlimited money; ISPs are hesitant to spend money if they don't expect a return. [22]

3-9. *ISPs continually assess whether to enter into new regions.* Internet service has expanded over the past decade in both expanding coverage as well as improving service to current regions (ie. companies moving into new coverage locations).[23] Additionally, state- and federal-level policy shifts encourage internet expansion.[24-26] Finally, new technology will prompt positive shifts in internet coverage.[27]

3-10. *ISP service area expansion is done primarily by large, nationwide companies.* Smaller companies have less capital and thus less expansion potential. We choose, for the U.S., to focus on ISPs in 10 or more states covering at least 1M people, of which there are 26.[28] For the U.K., we look at ISPs covering 100k+ people, of which there are 10.[29]

3-11. *Transit costs' decline can be modeled by exponential decay.* Historical data shows exponential decline in transit costs, which matches conventional wisdom about technology prices.[30,31]

3-12. *Internet transit costs are the same in the U.S. and the U.K..* Transit costs are dependent on technology, which isn't regionally dependent.[32]

3-13. *Competition drives prices lower, while stagnant regional industries do not experience price disruption.* Without external pressure from competition, oligopolist profit maximizing companies have no reason to change prices when consumers already pay inflated prices.[33]

3-14. *Profits are expected to be equally distributed among all ISPs serving a given region.* In reality, this isn't necessarily true, but it is incredibly difficult to model exact profits, which are not relevant to our model. We simply assume the ISPs looking to disrupt and enter a market usually believe they will capture a suitable market share.

3-15. *Regional populations in developed countries remain constant.* In most cities we looked at, population growth is small, and in many cases below 1 percent.[34-35]

3-16. *U.K. ISPs expand at half the rate as U.S. ISPs.* Regions in the U.K. we explored have on average double the amount of ISPs as those we explored from the U.S., and new ISPs are less likely to expand into a region they already occupied by many others.[36-37]

3-17. *Broadband pricing is determined by supply-side competition and can be modeled with an agent-based Monte Carlo approach.* Assumptions 3-4 through 3-16 allow us to simulate the decision making of ISPs and their effect on prices.

3.2 Model Parameters

Time (t): Time, expressed in years.

Transit Cost exponential decay at time t (TCE_t): Exponential decay rate at time t. Because exponential decay is not perfect,

$$TCE_t \sim \mathcal{N}(TCEM, TCES)$$

where $TCEM$ is the mean exponential decay per year and TCES is the standard deviation. An exponential regression of past data shows that $TCEM = .673$, while the historical suggests $TCES = .107$ (3-11).

Transit Cost per Mbps at time t (TC_t): Transit cost is modeled by exponential decay (3-11) with the following equation: TC_t=$(TC_t - 1) * TCE_t$ except when $t < 2021$, where TC_t=$(TC_t - 1) * TCEM$. This is because accessible data only goes to 2015, and developments from 2015 to 2020 are not random.[30]

ISP i (ISP_i): Represents one ISP agent.

Number of ISPs (NI): Dependent on 3-10.

Chance of ISP i exploring expansion at time t ($CI_{i,t}$): Randomly distributed variable which starts in 2021 with the following equation:

$$CI_{i,t} \sim \mathcal{N}(CIM, CISD)$$

where $CIM = .3$ for U.S. and .15 for U.K., and $CISD = .1$ for U.S. and .05 for U.K. (3-16). After ISP_i expands in timestep t, $CI_{i,t+1}$ is set to 0. If an ISP_i does not expand in timestep t, $CI_{i,t+1} = CI_{i,t} + .05$. The chance of ISP_i exploring expansion options in timestep t is equal to $CI_{i,t}$, where if $CI_{i,t}$ is greater than 1 or less than 0 it is reduced to 1 or increased to 0, respectively.

Cost buffer of region j at time t (CB_j): We assume that all costs other than the transit cost are fixed buffer cost (3-4) and also that markets are in a stable equilibrium at each timestep (3-13). Thus, CB_j is the sum of all costs such as infrastructure investment, salaries, and dividends/reinvestment (profit margin, which is here a "cost" as it is a component in the difference between revenue and transit cost). At the beginning,

$$CB_j = (C_j - TC_{2021}) * \mathcal{N}(CBM, CBSD)$$

where CBM is 1 and CBSD is .05, as the cost buffer cannot be exactly pinpointed.

City or Region j Mbps cost at time t ($C_{j,t}$): $C_{j,2021}$ is initially calculated as $\frac{\text{median cost}}{\text{median or average Mbps}}$ (3-1, 3-2). When ISP_i moves into region j because it identifies a new market, a new competitive equilibrium forms. $C_{j,t}$ becomes

$$(CB_j + TC_t) * \mathcal{N}(1, .1)$$

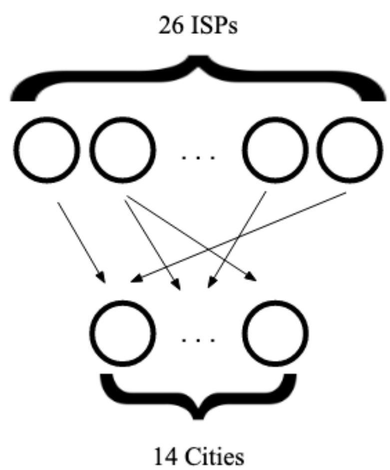
The normal distribution accounts for randomness in competition results between competing firms. Essentially, the new price at region j becomes the new transit cost (variable cost) plus the buffer cost (fixed cost, including profits).

Population of region j (P_j): Constant population of region j (3-15).

Boolean existence of ISP i in region j at time t ($SL_{i,j,t}$): To keep track of whether an ISP has already entered a region, $SL_{i,j,t} = 1$ if and only if ISP_i is in region j.

3.3 Model Development

Here, we present an agent-based Monte Carlo simulation for predicting future broadband Mbps prices in the U.S. and U.K. This model, due to its randomness and multiple independent agents, is complicated. Therefore, we also evaluate traditional regression models to assess our model's effectiveness.
We start with a series of 14 U.S. cities, 5 U.K. regions, 26 U.S. ISPs, and 10 U.K. ISPs (3-3; 3-10). We define these entities in 2.3. Without loss of generality, the U.S. model will be covered here.
We begin with 26 individual agents representing ISPs, with 14 possible areas of expansion (Figure 1).



1.png 1.png

Figure 1: Model Structure

In 2021, we assume that each of these ISPs has not entered any of the regions. This is not exactly true, as each city already has about 3 ISPs,[36-37] but the model is relatively robust to this flaw (3.5).
We then calculate the costs associated with each city, C_j, 2021. From the Open Technology Initiative we find the average cost per Mbps in each city (3-1, Figure 2).

1.png 1.png

Location	Initial Cost ($/Mbps)	Location	Initial Cost ($/Mbps)	Location	Initial Cost (£/Mbps)
Ammon, ID (U.S.)	0.4500	Lafayette, LA (U.S.)	0.1882	Urban 1 (U.K.)	0.3125
Atlanta, GA (U.S.)	0.3250	Los Angeles, CA (U.S.)	0.6552	Urban 2 (U.K.)	0.3125
Chattanooga, TN (U.S.)	0.2320	New York, NY (U.S.)	0.2750	Urban 3 (U.K.)	0.3125
Cleveland, OH (U.S.)	0.5000	San Francisco, CA (U.S.)	0.4995	Urban 4 (U.K.)	0.3125
Fort Collins, CO (U.S.)	0.2667	Seattle, WA (U.S.)	0.2750	Rural 1 (U.K.)	0.3125
Kansas City, KS (U.S.)	0.4000	Washington, DC (U.S.)	0.1916		
Kansas City, MO (U.S.)	0.5495	Wilson, NC (U.S.)	0.4165		

Figure 2: Initial Pricing

Then, we calculate each CB_j, $CI_{i,2021}$, TC_t (we can calculate for all t, as it is an independent variable, shown in Figure 3) as outlined in 3.2. We proceed with the following steps, starting with $t = 2021$. Note that randomness is introduced after every iteration to simulate the real world:

2.png 2.png

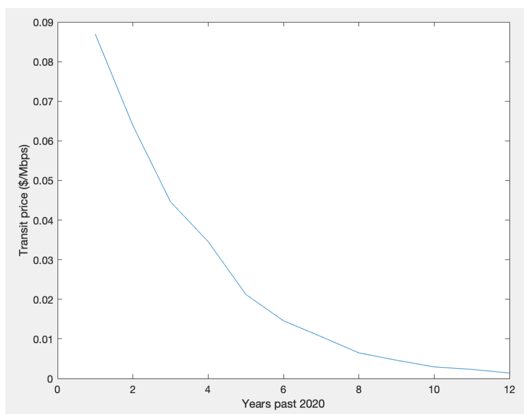


Figure 3: Transit costs per year

The iterative steps are listed below.

1. $t = t + 1$.
2. If $CI_{i,2021} > r$, where r is a random number from 0 to 1, then agent ISP_i evaluates investment options. This can be understood as a random probability of an ISP having the capital to expand.
3. If ISP_i is evaluating investment options, then for each j, ISP_i evaluates potential for profit, $\frac{P_j * (C_{j,t} - CB_j - TC_t)}{\text{number of ISPs invested in region } j}$. This amount can be understood as the population of region j multiplied by the difference between the current price and expected price (variable + fixed cost).
4. If the region with most potential for profit has an expected profit greater than 0, then the agent invests in that region. A new competitive equilibrium is reached in that region, and $C_{j,t}$

is calculated (3.2).

5. Variables are tidied: $SL_{i,j,t}$ and $CI_{i,t}$ are updated (3.2).

This simulation is repeated until $t = 2031$.

3.4 Model Simulation

We run this agent-based Monte Carlo process multiple times and find the following mean expected price per Mbps in each city (Figure 4). U.K. regions are averaged to find a nationwide broadband price. Histograms are displayed for the example regions (Figure 5).

1 REAL.png 1 REAL.png

Location	Mean Final Cost ($/Mbps)	Location	Mean Final Cost ($/Mbps)	Location	Mean Final Cost (£/Mbps)
Ammon, ID (U.S.)	0.4094	Lafayette, LA (U.S.)	0.1089	U.K.	0.2287
Atlanta, GA (U.S.)	0.2414	Los Angeles, CA (U.S.)	0.5653		
Chattanooga, TN (U.S.)	0.1509	New York, NY (U.S.)	0.1893		
Cleveland, OH (U.S.)	0.4149	San Francisco, CA (U.S.)	0.4135		
Fort Collins, CO (U.S.)	0.1857	Seattle, WA (U.S.)	0.1905		
Kansas City, KS (U.S.)	0.3186	Washington, DC (U.S.)	0.1081		
Kansas City, MO (U.S.)	0.4658	Wilson, NC (U.S.)	0.3392		

Figure 4: Mean Final Cost in 2031

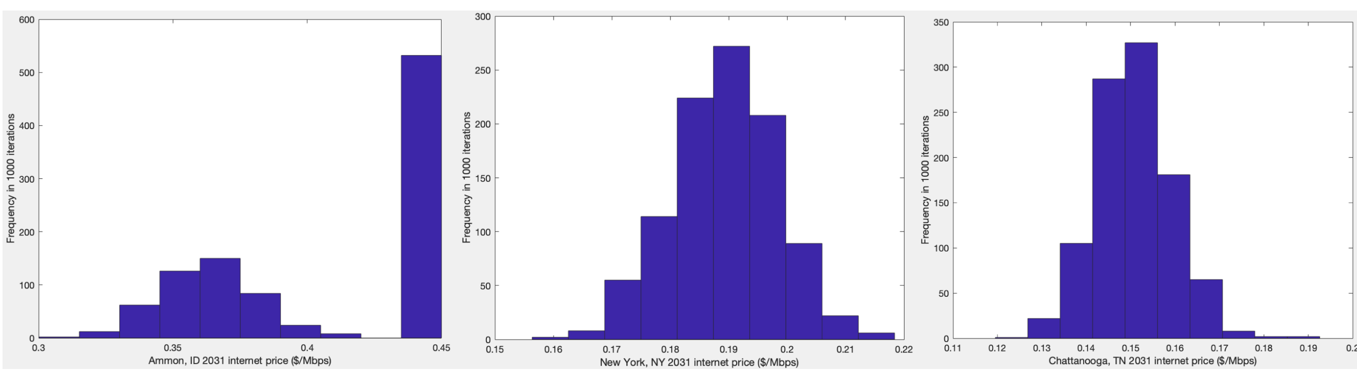


Figure 5: Final Cost by City in 2031

Interesting conclusions are reached. Cities with small populations, such as Ammon, ID, are not priorities for ISPs in most simulations. Therefore, it will be important for state and federal governments to ensure that traditionally overlooked areas are not left behind. The Monte Carlo simulation results in a distribution which looks close to normal, matching our expectations about randomness. Locations that start with lower prices continue to have lower prices, as expected fixed costs are higher (3-4). The average price across all U.S regions simulated is \$.293 per Mbps.

3.5 Sensitivity Analysis

To test how accurate and robust the results of our model are, we varied the two main parameters, initial competitors, by -3 and $+3$, along with the median price/ median Mbps by ± 10 percent. The results are presented below (Figure 6, Figure 7), with the U.S. results averaged over all regions.

Change	**Result** (Pounds/Mbps)	**Percentage Change**
No Change	0.2287	0%
-3 Competitors (Can't go below zero)	0.2285	-0.087%
+3 Competitors	0.2286	-0.043%
+10% Median $/Median Speed	0.2601	13.729%
-10% Median $/Median Speed	0.1975	-13.642%

Figure 6: U.K. Sensitivity Analysis

Change	**Result** (Dollars/Mbps)	**Percentage Change**
No Change	0.2929	0%
-3 Competitors (Can't go below zero)	0.2927	-0.068%
+3 Competitors	0.2927	-0.068%
+10% Median $/Median Speed	0.3297	12.564%
-10% Median $/Median Speed	0.2556	-12.734%

Figure 7: U.S. Sensitivity Analysis

While our model isn't sensitive to changes in the number of initial competitors, it is sensitive to changes to the initial price to speed ratio (cost). This makes sense, since we assume that companies will only lower prices with new competition; some regions will experience no change in prices, leading to a high or lower average respective to the change in the ratio.

3.6 Comparison with Traditional Model

To further test our model, we compared it to more traditional linear regressions. For the U.S., we took five different regressions corresponding to the five U.S. cities with both 2020 and 2012 data.[2] We then used these regressions to predict the 2013 price, which was averaged between the five regressions. For the U.K., Superfast network data from the U.K. Office of Communications was used to produce a single regression.[2] The predictions made by the regressions versus the predictions made by our model are in Figure 8.

The U.S. regressions are incorrect, as they predict a negative value, which wouldn't happen as the lower bound for price is the transit cost since ISPs would break even providing internet at such a cost. The U.K. regression however, predicts a value similar to the agent-based model, but the regression is still flawed. If it continued over more than 10 years, the U.K. regression would eventually predict an impossible negative value, while our model is able to make robust predictions well into the future. We believe that the U.K. regression model is initially consistent because U.K. internet service prices in 2021 are more competitive and close to optimized, meaning they are currently

Country	Regression Value	Our Value
UK	0.224	0.2287
U.S.	-2.2755	0.2929

Figure 8: Model Comparison

decreasing at a slower rate than U.S. prices.[40]

3.7 Strengths and Weaknesses

This agent-based model allows for streamlined modeling of extremely advanced economic concepts and outperforms other simplifications such as regressions. It accounts for ISP expansion over time, population size, competition, and changing variable costs. The model is resilient to changes in agent properties, such as competitor population.

However, the model is not resilient to certain changes in starting parameters, such as initial cost. Thus, while the model has predictive value, it requires accurate starting data, which is not always available.

4 Part II: Bit by Bit

4.1 Assumptions

4-1. *Required bandwidth for an activity is defined as the bandwidth needed to sustain an activity over a period of one hour.* Internet media consumption is measured by average hours per week, so for consistency it is best to measure bandwidth using the same unit of time.

4-2. *People sleep the recommended amount[10], work 9 hours[11], and go to school for 7 hours[12].* For the sake of simplicity the model assumes that people sleep, work, and go to school for the same amount of hours each time. 50 percent of students in school are completely virtual.[42] In-person students are half hybrid-learning and half in-person-learning.

4-3. *HD and SD videos are watched at equivalent rates.* What constitutes high definition is not well defined, so the model assumes people watch videos of varying quality at equal rates.

4-4. *4k Ultra HD TVs are present in 37 percent of households.* In 2018, they were present in 31 percent of households[18]. Fitting a logistic regression, the data suggests that in 2021 now 37 percent of households have these TVs.

4-5. *If people own 4k Ultra HD TVs they will stream on 4k Ultra HD definition.* The model assumes that people will take full advantage of the technology in their household.

4-6. *Despite the increase of internet consumption, the rate of noninternet activities have remained constant.* Internet consumption is up 70 percent, but the model assumes that people still watch TV at the same rate as before the pandemic.[16]

4-7. *12–17 year olds have apporximately equivalent internet consumption compared to 18–34-year-olds.* 12–17-year-olds use TV connected game consoles slightly more and TV connected internet devices slightly less compared to 18–34-year-olds, so the model assumes that they will generally use the internet at the same rate.

4-8. *Family members use internet independently.* In reality, certain families may tend to use the internet heavily or sparsely during certain times.

4-9. *The teacher in example household 1 makes 60,000 dollars a year. In the 2018–2019 school year, the average school teacher made 61,730 dollars.*[39]

4-10. *Traditional television does not use bandwidth.*

4-11. *Children aged 11 and under don't use the internet outside of the television.*

4-12. *If a person is both in school and employed, we automatically assume the job is part time.*

4.2 Generating Profiles

In order to create a flexible model, we rely on a Monte Carlo simulation, which creates plausible "profiles" given certain user inputs. Each profile contains the following properties:

- Whether they're in school (online, hybrid, or in-person)
- Whether they're employed (working form home, not working, or essential worker)
- How many hours they sleep
- Whether they are away from home any days
- How old they are
- *And of utmost importance!* Whether they have a 4k Ultra HD TV (Surprisingly common in households and requires significant bandwidth)

The only required user input for a profile is age, which is used to determine the probability of employment, how long a person sleeps, and other properties. Each of the profile properties can also be entered as an input. Income is an optional profile input, which changes how many average hours people do certain internet activities. The data for average hours per activity by income was outdated, so we normalized the data by converting each entry to percentages of the mean and applied those percentages to our updated data. The percentages are shown in the table above. The final, and most random, property of the profile is the large file download frequency. We used a beta distribution with alpha = 1 and beta = 8 to create a skewed distribution with most values near 0. Large file downloads were chosen to be relatively rare occurrences given that they require the most bandwidth out of all activities by far.

4.3 Model Simulation

Given at least age, the model uses the given input parameters to generate 100 profiles derived from the input. For each generated profile, the model simulates an entire year of internet consumption by going through hour by hour. For example, the user starts their day off when they wake up after some number of hours of sleep. Then, depending on what the profile says, they may go to school or work. If they're attending in-person school, the simulation will show no internet consumption for 7 hours, and if they attend online school, their internet consumption will be 4 Mbps (required bandwidth for video conferencing) over 7 hours. A similar thing will be done for

Activity	< $25K	$25k-50K	$50k-75K	> $75K
Watching Traditional Television	1.32	1.07	0.90	0.71
TV Connected Game Console	1.45	1.15	0.78	0.62
TV Connected Internet Device**	1.32	1.15	0.90	0.62
Total Internet on a Computer	1.24	1.00	0.94	0.81
Total App/Web on a Smartphone	1.10	1.01	1.01	0.88
Total App/Web on a Tablet	1.00	1.04	1.02	0.94

Figure 9: Consumption Percentage by Income

work. In the person's free time they will have certain preset probabilities, determined by age and income, of doing certain internet activities. Naturally, they will consume the required bandwidth of the corresponding activity.

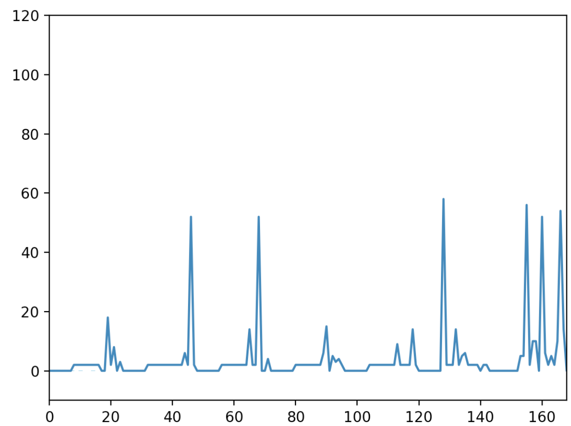


Figure 10: Example Simulation

In the figure above, you can see a randomly generated week of internet consumption for a 30-year-old working at home. As you can see, there are repetitive patterns of internet consumption during the weekday, namely low levels of internet consumption while working. However, during the weekends there is a spike of internet consumption, which makes sense given the increased free time. The model continues this simulation for an entire year and after iterating through each of the 100 profiles compiles all results and outputs a histogram representing the amount of bandwidth used per hour. The model can also take in multiple profiles, which is useful for determining the bandwidth that a household needs. When given multiple profiles the model simply runs the simulation for each profile and aggregates the results together. Finally, after generating all the hourly internet consumption data the model iterates through increasing amounts of bandwidth until the bandwidth can support 90 percent of internet consumption. In order to support internet consumption the Mbps must be 50 percent more than the required amount in order for the internet to be sufficient, because of wireless interference and distance from the router.[2]

4.5 Testing Specific Profiles Since we've defined a framework to input profiles, all we have to do is create profiles for each person in each household. For household 1, we create the following profiles:

- Profile 1: age = 30, income = 60000, isEmployed = True, employedEssential = True, inSchool = False

- Profile 2: age = 30, income = 0, isEmployed = False, inSchool = False
- Profile 3: age = 3, income = 0

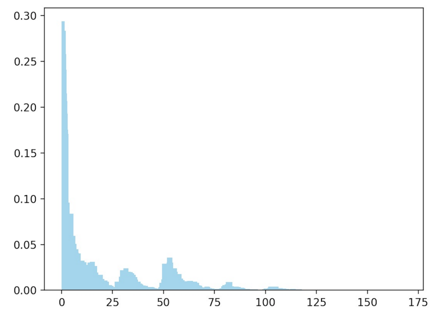


Figure 11: Household 1

As you can see in the histogram, household 1 primarily uses low bandwidth internet, but in order to support 90 percent of the internet consumption the bandwidth must be 105 Mbps, and 179 Mbps for 99 percent. While this number is quite high, it is reasonable given that there are 3 members of the household. These are the profiles for household 2:

- Profile 1: age = 70, income = 0, isEmployed = False, inSchool = False
- Profile 2: age = 7, income = 0, inSchool = True, daysOff = 5
- Profile 3: age = 13, income = 0, inSchool = True, daysOff = 5

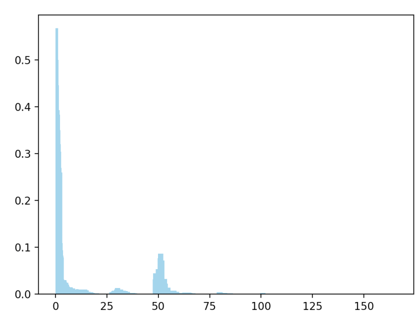


Figure 12: Household 2

71 Mbps is the required bandwidth for this household for 90 percent support, and 129 Mbps for 99 percent support. This result makes intuitive sense compared to household 1 because although the family size is the same, the two children are at the house only twice a week, hence the far lower internet consumption. Finally, we have our last household:

- Profile 1: age = 19, isEmployed = True, inSchool = True
- Profile 2: age = 20, isEmployed = True, inSchool = True
- Profile 3: age = 20, isEmployed = True, inSchool = True

123 Mbps is the required bandwidth for this household for 90 percent support, and 219 Mbps for 99 percent support. Unsurprisingly, this household requires the largest amount of bandwidth.

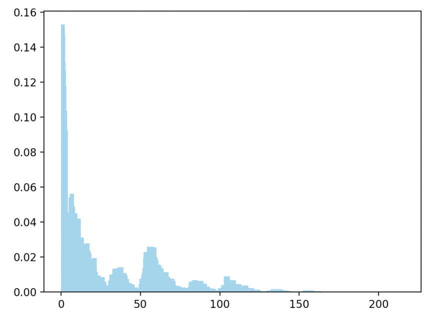


Figure 13: Household 3

4.6 Sensitivity Analysis In order to ascertain how sensitive the model is to certain input parameters we adjusted various parameters for household 1. For example, if we assume that the minimum amount of bandwidth is sufficient, household 1 only needs 53 Mbps for 90 percent support and 88 Mbps for 99 percent support. Furthermore, if we revert back to needing 50 percent more than the minimum required bandwidth and adjust profile 1 to also be unemployed, we see the required amount of bandwidth is 109 Mbps for 90 percent support. This result is only marginally higher than when the person was working, showing that required bandwidth is more dependent on the number of members in the household and their age rather than their jobs. Finally, we adjusted the beta distribution to alpha = 1 beta = 15, meaning that large file downloads were far more unlikely to be downloaded. This change had a significant result dropping the required bandwidth for 90 percent support to 69 Mbps, even with the necessary 50 percent margin. This results indicates that downloading large files is one of the main uses of bandwidth.

4.7 Strengths and Weaknesses The biggest strength of this model is its flexibility. The ability to input very specific base profiles allows the model to create quite accurate representation of internet consumption. Furthermore, the Monte Carlo technique accounts well for the inherent unpredictability of modeling an entire household based on just a few inputs. The biggest weaknesses of the model is the assumptions it makes. For example, the formula for downloading large files is quite subjective and not based in evidence. Furthermore, hours of sleep, work hours, and school hours are all constant, when in reality it would fluctuate a lot. Finally, bandwidth usage would spike and dip at certain times within households, such as dinner time.

5 Part III: Mobilizing Mobile

5.1 Assumptions

5-1. *The population is uniformly distributed with constant population density in each given subregion.*

5-2. *Given a range of potential memory usage per hour (ex. 40 MB/hr–300 MB/hr) for a certain device activity, the average was taken (170 MB/hr in the previous range).*

5-3. *Teens living in households with an annual income of less than 35,000 USD spent 8.5 hours per day on screen media, and teens in households with an annual income of above 100,000 USD spent 6.8 hours per day* [38].

5-4. *Screen time in hours is linear for a fixed age group. This works on the basis that this overpredicts the screen time (based on the tables in D4) so it yields an upper bound for calculations.*

5-5. *The median (both in age and in income) was used as a representative of a subregion. For example, if a subregion had a median age of 25, then the data usage for the entire population was assumed to lie in the same range as 25.*

5.3 Data Extrapolation

The data given in D4 is divided into separate tables: two of them deal with screen time stratification by age and the other features a breakdown based on income. In order for the model to work, the data needed to be combined (i.e., a breakdown that would give an average screen time summary based on age and income range).

Here, assumptions 5-3 and 5-4 are significant. First, we examined the income range of 0-35,000. Given that a teenager (age 12-17) in a family with income $< 35,000$ uses screen media for 8.5 hours/day (59.5 hours/week), we can use proportions to determine the hours per week for the other age ranges.
Since the tables in D4[2] are limited for the age range 12–17 (only contains the first three), we computed the average screen time for the age range 18–34 according to the following expression:

$$(59.5) \cdot \frac{11.35 + 3.63 + 6.95}{7.60 + 4.18 + 4.52} = 80.05$$

For the following age ranges, we computed the average screen time according according to a similar equation:

$$S_{current} = S_{prev} \cdot \frac{\text{sum of screen time of current age group [across all incomes]}}{\text{sum of screen time of previous age group [across all incomes]}}$$

In the above equation, $S_{current}$ and S_{prev} represent the average weekly screen time for this fixed income range of $< 35,000$, for the current and previously analyzed age group. The results for this computation can be summarized in the table below:

age group	predicted screen time
12-17	59.5
18-34	80.05
35-49	94.3
50-64	102.43
65+	102.21

Table 5.1: Predicted screen time (weekly) by age range for families with income in the range $< 35,000$

Now, given these predicted screen time, we can compute the number of megabytes (not megabits) required. Using the following rates (measured in MB/hr), we computed a weighted average that provided us with an estimate of how many MB a person would be used in an hour. 4G/5G already satisfy many of these usage types, and will continue to grow in prevalence over time.

device usage type	data used (MB/hr units)
Traditional Media	1000
TV Connected Game Console	170
TV Connected Internet Device	700
Internet	60
Video	600
Total App/Web on Smartphone	180
Total App/Web on Telephone	205

Table 5.2: Corresponding amount of data used at an hourly rate for a given type of device usage.

Performing a weighted average with the above values gives us an average data consumption rate stratified by age group summarized below:

age range	average data used (MB/hr units)
18-34	393
35-49	486
50-64	601
65+	689

Table 5.3: Average amount of data used at an hourly rate for a specified age ranges.

Now, we can multiply the rows in the above table by the corresponding number of hours predicted in Table 5.1 and convert units from MB/hr to Mbps to get the following table which relates how many megabits per second an individual in the age range is using. (Still examining the income range of $< 35,000$.)

age range	average megabits per second
18-34	0.0939
35-49	0.13687
50-64	0.18387
65+	0.21035

Table 5.4: Average megabits per second for a specified age ranges for income of $< 35,000$.

This prediction method allows us to estimate the data required for the two parameters of age and income, which in turn allows us to model the regions in question. For example, in region A, the median income for each subregion lay in the range of $< 50,000$, so we repeated the process detailed for that income range:

age range	average megabits per second
18-34	0.07951
35-49	0.11582
50-64	0.15559
65+	0.17799

Table 5.5: Average megabits per second for a specified age ranges for income of $35,000 - 50,000$.

Through looking at the median income and age and population of the subregions region A, we computed the required Mbps per region:

subregion	median age	median income	population	required Mbps
1	28.2	27941	690	64.837
2	30.2	30929	1422	133.62
3	40.7	47163	1303	150.914
4	64.3	34273	278	51.117
5	37.8	30425	1243	170.136
6	36.9	46659	1391	161.107

Table 5.6: Summary of each subregion in region A detailing the median age, median income, population and required Mbps.

The same methods were used for regions B and C with the following results:

subregion	median age	median income	population	required Mbps
1	47.7	99652	3873	426.244
2	48.2	134375	2114	232.657
3	59	173188	1253	185.252
4	36.3	112306	1129	124.252
5	55.1	108056	2493	368.582
6	45.5	147500	2398	263.91
7	53.8	132045	1794	265.237

Table 5.7: Summary of each subregion in region B detailing the median age, median income, population and required Mbps.

subregion	median age	median income	population	required Mbps
1	41.67	214125	1468	161.561
2	24.9	96209	1624	122.700
3	47.1	190729	1012	111.376
4	44.7	139261	1295	142.521
5	32.7	152500	1309	98.900
6	47.9	206875	1008	110.935
7	40.7	151731	1789	196.889

Table 5.8: Summary of each subregion in region C detailing the median age, median income, population and required Mbps.

5.4 Model Development

We implement a simulated annealing based algorithm to place the circular areas of low, medium, and high frequency broadband nodes in the optimum position such that a regions' network needs are entirely met (regions being the three A,B,C regions provided).

From our processed data reflecting the network needs for the respective subregions for our larger regions A,B,C we aim to find global optimums for our node configurations. For each region, our algorithm assigns the subregions a color though image processing which is later converted to a gray-scale color mapping related to their predetermined Mbps broadband usage.

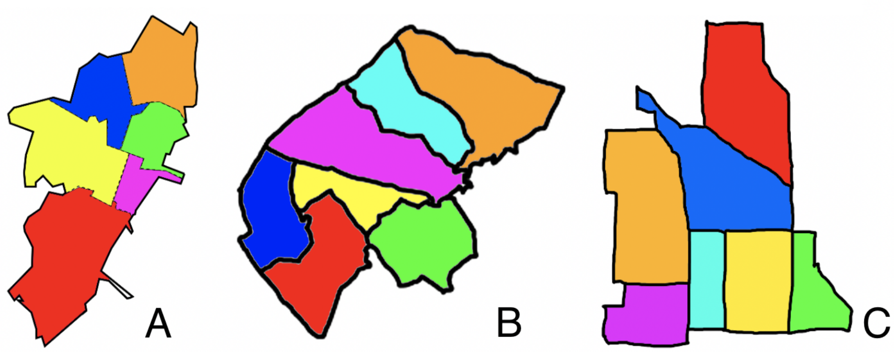


Figure 14: Regions A, B, and C - Image Processed Color Coding

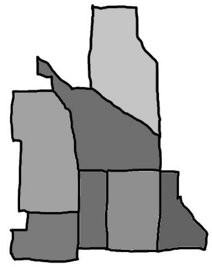


Figure 15: Region C - Bandwidth Correlated Gray Scale Adjustment

Each placed node has a specific range of coverage and strength of coverage, both of which reduce the gray-scale value of all pixels under the node's area. Eventually, a pixel reaches a value 0, indicating coverage has been fully satisfied for that pixel (mind that overlapping node area capacities are combined).

To expand on the rationale for using simulated annealing for our algorithm. The optimization technique is integral when attempted to discern a state's global optimum from possible local optima when naively altering the state (or in our case moving the nodes and changing their capacity levels). Simulated annealing achieves this by searching for a local optimum alongside a probability temperature scheme which "cools" (meaning it is less likely to move) at each iteration to determine whether to randomly move to a nearby, but random state. Eventually the temperature algorithm cools sufficiently to no longer justify moving to a new state.

We begin with optimizing the position and low/med/high capacity of a single node on the region's map. Upon assessing the net neglected coverage (determined by averaging all pixels remaining network coverage needs), the algorithm chooses to add a new node if the net neglected coverage is greater than 0.

This process resumes with 2 nodes, optimizing their positions and approaching a global maximum considering their variables of position and low/med/high capacity. If net neglected coverage is 0, the algorithm exits; otherwise the process is repeated.

5.5 Results

Upon observing the results for all three simulations, we see that there should be some sort of stratification based on area.

Region A requires 731 Mbps and has overall area of 6.83 square miles. Since the medium broadband covers between 4π and 9π area ($6.83 < 4\pi$) and has a max download of speed of 900 Mbps, it is optimal to place a medium broadband at the center of the region.

For Region B, since the total area is 33.64 square miles and the region 1866 Mbps, it would be best served using three medium broadbands, because the max download speed of $3 \cdot 900 = 2700$

Mbps exceeds the required download speed. Additionally, the circles can be configured to cover the necessary area.

Finally, for Region C, which has area 1.64 square miles and requires 945 Mbps, the optimal strategy is to place a high broadband, which covers both the download speed and area.

Upon observing the results for all three simulations, it is clear that meeting the needs of the region in its entirety (as for as our three regions A, B, and C are concerned) is a matter of having a sufficient number of nodes to cover the area while the node strength is of secondary importance. This makes sense as the low capacity nodes have a radius which eclipses those of the medium and high capacity nodes, yet the medium and high nodes have a significant strength increase. Although maybe relative intuitive, we noted region A requires a single medium node, B requires no more than three medium capacity nodes, while Region C requires 1 high capacity nodes placed consideration.

It is worth mentioning that when using one node less then the optimum which addresses each subregion's network needs we reach a similar conclusion that one need only use low capacity nodes to address the region's needs.

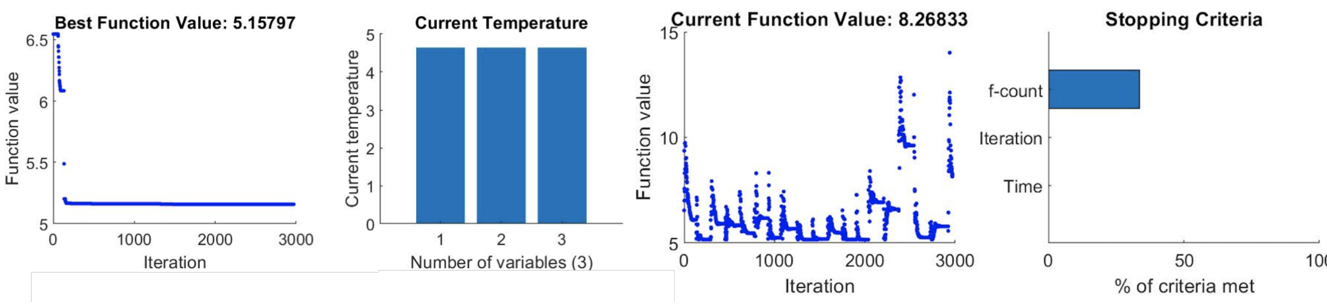


Figure 16: Temperature Visualization - Optimizing Region B with 3 Nodes

Above is a visualization of the annealing process performed on region B with three nodes (note: three nodes is one node short of the needed amount to cover the full area entirely producing a net neglected coverage of 5Mbps).

6 Works Cited

[1] https://www.fcc.gov/general/types-broadband-connections

[2] Defeating the Digital Divide Data, MathWorks Math Modeling Challenge 2021, https://m3challenge.siam.org/node/523

[3] https://arxiv.org/pdf/1405.2957.pdf

[4] https://www.telecompetitor.com/report-u-s-median-broadband-price-is-80-monthly/

[5] https://www.ispreview.co.uk/index.php/2020/05/ofcom-2020-study-average-uk-broadband-isp-speeds-hit-64mb.html

[6] https://www.newamerica.org/oti/reports/cost-connectivity-2020/

[7] https://www.nature.com/articles/gim200662

[8] https://escholarship.org/content/qt3jh9d5xd/qt3jh9d5xd.pdf

[9] https://drpeering.net/core/ch2-Transit.html

[10] https://www.sleepfoundation.org/how-sleep-works/how-much-sleep-do-we-really-need

[11] https://www.cnbc.com/2017/05/03/how-the-8-hour-workday-changed-how-americans-work.html

[12] https://nces.ed.gov/surveys/sass/tables/sass0708035s1s.asp

[13] https://www.thebalancecareers.com/how-does-a-salary-range-work-1918256

[14] https://www.spglobal.com/ratings/en/research/articles/200506-economic-research-infrastructure-what-once-was-lost-can-now-be-found-the-productivity-boost-11355758

[15] John Y. Campbell, Andrew W. Lo, and A. Craig MacKinlay, The Econometrics of Financial Markets, 2012.

[16] https://www.forbes.com/sites/markbeech/2020/03/25/covid-19-pushes-up-internet-use-70-streaming-more-than-12-first-figures-reveal/?sh=4dec7df3104e

[17] https://www.washingtonpost.com/technology/2019/09/13/cities-not-rural-areas-are-real-internet-deserts/

[18] https://www.statista.com/statistics/736142/4k-ultra-hdtv-us-household-penetration/

[19] https://www.sciencedirect.com/science/article/pii/S0198971519300808

[20] https://www.omnicalculator.com/finance/profit

[21] https://www.economicshelp.org/blog/3201/economics/profit-maximisation/

[22] https://www.businessinsider.com/internet-isps-competition-net-neutrality-ajit-pai-fcc-2017-4#-6

[23] https://www.fcc.gov/reports-research/reports/broadband-progress-reports/eighth-broadband-progress-report

[24] https://www.kwtx.com/2021/01/19/internet-expansion-plans-up-for-debate-in-texas-legislature/

[25] https://www.nga.org/center/publications/expand-affordable-broadband/

[26] https://thecounter.org/fcc-20-billion-dollar-rural-broadband-auction/

[27] https://www.cnbc.com/2020/10/27/spacex-starlink-service-priced-at-99-a-month-public-beta-test-begins.html

[28] https://broadbandnow.com/All-Providers/page/2

[29] https://www.ispreview.co.uk/review/top10.php

[30] https://drpeering.net/white-papers/Internet-Transit-Pricing-Historical-And-Projected.php

[31] https://www.investopedia.com/terms/m/mooreslaw.asp

[32] https://www2.telegeography.com/hubfs/2017/presentations/telegeography-ptc17-pricing.pdf

[33] https://www.economicshelp.org/blog/5597/economics/price-stability-in-oligopoly/

[34] https://datausa.io/profile/geo/chattanooga-tn/

[35] https://www.macrotrends.net/cities/23052/los-angeles/population

[36] https://labs.thinkbroadband.com/local/broadband-map#6/49.732/-1.181

[37] https://broadbandnow.com/national-broadband-map

[38] statista.com/statistics/1099629/hours-screen-time-teens-income

[39] https://www.businessinsider.com/teacher-salary-in-every-state-2018-4

[40] https://arstechnica.com/tech-policy/2010/03/uk-regulators-officially-mock-us-over-isp-competition/

[41] https://onlinestatbook.com/2/normaldistribution/intro.html

[42] https://www.edweek.org/teaching-learning/districts-retreat-to-remote-learning-even-as-biden-calls-for-reopening-schools/2021/01

[43] https://news.stanford.edu/2020/06/29/snapshot-new-working-home-economy/

[44] https://www.statista.com/statistics/217899/us-employment-rate-by-age/

7 Code

7.1 Question 1, 2, and 3 Code

```
function [prices] = q1sim()

numprov = 26;
enter_prob_mean = .3;
enter_prob_std = .1;

providers = zeros(26,1);

for i = 1:numprov
    providers(i,1) = normrnd(enter_prob_mean,enter_prob_std);
    i = i+1;
end

locations = table2array(readtable('q1us.xlsx')); %price and population for each location

transitprice=zeros(12,1);
transitprice(1,1)=0.08697929247;
pricechangemean = .673;
pricechangestd = .107;
for i = 2:12
    transitprice(i,1)=transitprice(i-1,1)*exp(normrnd(pricechangemean,pricechangestd)-1);
end

plot(transitprice);

costbuffermean = 1;
costbufferstd = .05;
costbuffer = zeros(length(locations(:,1)),1);
for i = 1:length(locations(:,1))
    costbuffer(i,1) = (locations(i,1)-transitprice(1,1))*normrnd(costbuffermean,costbufferstd);
    i=i+1;
```

```
end
allresults = zeros(length(locations(:,1)),11);%log
allresults(:,1)=locations(:,1);
servlocations = zeros(length(locations),numprov);
for year = 2021:2031
    for i = 1:numprov
        if providers(i,1) > rand
            done = 0;
            temparray = locations(:,:);
            while done == 0
                [o,place]=max(temparray(:,3).*(temparray(:,1)-costbuffer-transitprice(year-2019)));
                if servlocations(place,i)==0
                    done = 1;
                    if max(locations(:,3).*(locations(:,1)-costbuffer-transitprice(year-2019))./(locations(:,2)+1)) > 0
                        [o,place]=max(locations(:,3).*(locations(:,1)-costbuffer-transitprice(year-2019)));
                        %['during ',num2str(year),' provider #',num2str(i),' moved into ',num2str(place)]
                        %['old price was ',num2str(locations(place,1)),' and the new price is ',num2str(costbuffer(place
                        locations(place,1)=normrnd(1,.01)*(costbuffer(place,1)+transitprice(year-2019));
                        locations(place,2)=locations(place,2)+1;
                        %locations(:,2)
                        providers(i,1)=0;
                        servlocations(place,i)=1;
                    end
                else
                    temparray(place,1)=-1000;
                    if max(temparray(:,1))==-1000
                        done = 1;
                    end
                end
            end
        end
```

```
        i = i+1;
    end
    providers(:,1)=providers(:,1)+.05;
    allresults(:,year-2020)=locations(:,1);
    year = year+1;
end

prices = locations(:,1);
```

```python
import numpy as np
import matplotlib.pyplot as plt
import random

# Average hours per week on the following activites (Source 2):
# Watching Traditional Television
# TV Connected Game Console
# TV Connected Internet Device
# Internet on a Computer (not including video)
# Video on a Computer
# Total App/Web on a Smartphone
#     Video Focused App/Web on a Smartphone
#     Steaming Audio on a Smartphone
# Total App/Web on a Tablet
#     Video Focused App/Web on a Smartphone
#     Steaming Audio on a Smartphone

age_2_11 = [12.00, 1.7 * 2.72, 1.7 * 7.72, 0, 0, 0, 0]
age_12_17 = [
    7.60,
    1.7 * 4.18,
    1.12 * 4.52,
    1.7 * 3.97,
    1.7 * 1.73,
    [1.7 * 24.83, 1.7 * 3.03, 1.12 * 1.62],
    [1.7 * 3.89, 1.7 * 0.98, 1.12 * 0.20],
]
age_18_34 = [
    11.35,
    1.7 * 3.63,
    1.12 * 6.95,
    1.7 * 3.97,
    1.7 * 1.73,
    [1.7 * 24.83, 1.7 * 3.03, 1.12 * 1.62],
    [1.7 * 3.89, 1.7 * 0.98, 1.12 * 0.20],
]
age_35_49 = [
    23.80,
    1.7 * 1.73,
    1.12 * 6.87,
    1.7 * 4.77,
    1.7 * 1.28,
    [1.7 * 26.99, 1.7 * 2.28, 1.12 * 0.85],
    [1.7 * 5.84, 1.7 * 1.07, 1.12 * 0.22],
]
age_50_64 = [
    40.02,
    1.7 * 0.47,
    1.12 * 4.95,
    1.7 * 5.03,
    1.7 * 1.07,
    [1.7 * 23.25, 1.7 * 1.38, 1.12 * 0.60],
    [1.7 * 6.25, 1.7 * 0.85, 1.12 * 0.17],
]
age_65_plus = [
    50.60,
    1.7 * 0.17,
    1.12 * 3.20,
    1.7 * 3.17,
    1.7 * 0.50,
    [1.12 * 17.73, 1.12 * 0.97, 1.7 * 0.42],
    [1.12 * 7.22, 1.12 * 0.65, 1.7 * 0.10],
]
```

```python

# Required bandwidth for each activity.
required_bandwith = [0, 3, 4, 1, 4, 1, 1]
# 40 weeks, which is the standard amount, is allocated per year for school.
school_on = False

fig = plt.figure()

# returns value if int, returns sum if list
def custom_sum(value):
    if isinstance(value, float):
        return value
    elif isinstance(value, int):
        return value
    else:
        total = 0
        for entry in value:
            total += custom_sum(entry)
        return total


# Profile class for setting user inputs and running Monte Carlo Simulation.
class Profile:
    def __init__(self, age):
        # Mandatory input
        self.age = age

    # Generate random profiles for monte carlo. Takes input parameters.
    def generate_profile(
        self,
        inSchool=None,
        virtualLearning=None,
        hybridLearning=None,
        inPersonLearning=None,
        isEmployed=None,
        employedAtHome=None,
        employedNotWorking=None,
        employedEssential=None,
        hoursOfSleep=None,
        ultraHdTv=None,
        income=None,
        daysOff=0,
    ):
        self.in_school = False
        self.virtual_learning = False
        self.hybrid_learning = False
        self.in_person_learning = False
        self.employed = False
        self.employed_at_home = False
        self.employed_not_working = False
        self.employed_essential = False
        self.ultra_hd_tv = False
        self.income = income
        self.days_off = daysOff

        if self.age >= 12:
            self.large_file_frequency = np.random.beta(1, 8) * 100
        else:
            self.large_file_frequency = 0

        # Source 43
        if self.age <= 2:
            in_school_probability = 0
```

```
        elif self.age <= 4:
            in_school_probability = 54.5
        elif self.age <= 6:
            in_school_probability = 93.4
        elif self.age <= 13:
            in_school_probability = 97.6
        elif self.age <= 17:
            in_school_probability = 95.4
        elif self.age <= 19:
            in_school_probability = 68.4
        elif self.age <= 24:
            in_school_probability = 38
        elif self.age <= 29:
            in_school_probability = 13.1
        else:
            in_school_probability = 6.4

        num = random.uniform(0, 100)
        if num <= in_school_probability:
            self.in_school = True

        #
        if self.age < 16:
            employed_probability = 0
        elif self.age <= 24:
            employed_probability = 45.9
        elif self.age <= 29:
            employed_probability = 73.5
        elif self.age <= 34:
            employed_probability = 75.6
        elif self.age <= 44:
            employed_probability = 77
        elif self.age <= 54:
            employed_probability = 75.5
        else:
            employed_probability = 36.4

        num = random.uniform(0, 100)
        if num <= employed_probability:
            self.employed = True

        if self.age < 1:
            self.hours_of_sleep = 14
        elif self.age <= 2:
            self.hours_of_sleep = 12
        elif self.age <= 5:
            self.hours_of_sleep = 11
        elif self.age <= 13:
            self.hours_of_sleep = 10
        elif self.age <= 17:
            self.hours_of_sleep = 9
        elif self.age <= 25:
            self.hours_of_sleep = 8
        elif self.age <= 64:
            self.hours_of_sleep = 8
        else:
            self.hours_of_sleep = 8

        # Source 42
        if self.in_school:
            num = random.uniform(0, 100)
            if num <= 50:
                self.virtual_learning = True
```

```
            elif num <= 75:
                self.hybrid_learning = True
            else:
                self.in_person_learning = True

        # Source 44
        if self.employed:
            num = random.uniform(0, 100)
            if num <= 42:
                self.employed_at_home = True
            elif num <= 75:
                self.employed_not_working = True
            else:
                self.employed_essential = True

        # Source 18
        num = random.uniform(0, 100)
        if num <= 37:
            self.ultra_hd_tv = True

        if inSchool != None:
            self.in_school = inSchool
        if virtualLearning != None:
            self.virtual_learning = virtualLearning
        if hybridLearning != None:
            self.hybrid_learning = hybridLearning
        if inPersonLearning != None:
            self.in_person_learning = inPersonLearning
        if isEmployed != None:
            self.employed = isEmployed
        if employedAtHome != None:
            self.employed_at_home = employedAtHome
        if employedNotWorking != None:
            self.employed_not_working = employedNotWorking
        if employedEssential != None:
            self.employed_essential = employedEssential
        if hoursOfSleep != None:
            self.hours_of_sleep = hoursOfSleep
        if ultraHdTv != None:
            self.ultra_hd_tv = ultraHdTv


# run multiple profiles
def simulate_profiles(profiles):
    total_data_x = np.zeros(52 * 168)
    total_data_y = np.zeros(52 * 168)
    for profile in profiles:
        x, y = run_simulation(profile)
        total_data_x += x
        total_data_y += y

    return total_data_x, total_data_y


# run simulation for one profile
def run_simulation(profile):

    total_data_y = np.array([])
    for i in range(52):
        if i < 40:
            school_on = True
        else:
            school_on = False
```

```
        y = simulate_week(profile)
        total_data_y = np.append(total_data_y, y)

    total_data_x = np.arange(52 * 168)

    return total_data_x, total_data_y


# Simulate internet consumption for a week based on activity probabilities and p
def simulate_week(profile):

    # Calculate amount of time sleeping, at school, or at work.
    busy_time = profile.hours_of_sleep * 7
    if profile.in_school and (profile.employed_at_home or profile.employed_essen
        busy_time += 11 * 5
    elif profile.in_school:
        busy_time += 7 * 5
    elif profile.employed_at_home or profile.employed_essential:
        busy_time += 9 * 5

    if profile.age <= 11:
        probabilities = age_2_11[:]
        remaining_hours = 168 - busy_time
        probabilities.append(remaining_hours)
    elif profile.age <= 17:
        probabilities = age_12_17[:]
        remaining_hours = 168 - busy_time
        probabilities.append(remaining_hours)
    elif profile.age <= 34:
        probabilities = age_18_34[:]
        remaining_hours = 168 - busy_time
        probabilities.append(remaining_hours)
    elif profile.age <= 49:
        probabilities = age_35_49[:]
        remaining_hours = 168 - busy_time
        probabilities.append(remaining_hours)
    elif profile.age <= 64:
        probabilities = age_50_64[:]
        remaining_hours = 168 - busy_time
        probabilities.append(remaining_hours)
    else:
        probabilities = age_65_plus[:]
        remaining_hours = 168 - busy_time
        probabilities.append(remaining_hours)

    # Source 2
    if profile.income != None:
        if profile.income <= 25000:
            1.32 * probabilities[0]
            1.45 * probabilities[1]
            1.32 * probabilities[2]
            1.24 * probabilities[3]
            1.24 * probabilities[4]
            1.10 * probabilities[5][0]
            1.10 * probabilities[5][1]
            1.10 * probabilities[5][2]
            1.00 * probabilities[6][0]
            1.00 * probabilities[6][1]
            1.00 * probabilities[6][2]
        elif profile.income <= 50000:
            1.07 * probabilities[0]
            1.15 * probabilities[1]
            1.15 * probabilities[2]
```

```
            1.00 * probabilities[3]
            1.00 * probabilities[4]
            1.01 * probabilities[5][0]
            1.01 * probabilities[5][1]
            1.01 * probabilities[5][2]
            1.04 * probabilities[6][0]
            1.04 * probabilities[6][1]
            1.04 * probabilities[6][2]
        elif profile.income <= 75000:
            0.90 * probabilities[0]
            0.78 * probabilities[1]
            0.90 * probabilities[2]
            0.94 * probabilities[3]
            0.94 * probabilities[4]
            1.01 * probabilities[5][0]
            1.01 * probabilities[5][1]
            1.01 * probabilities[5][2]
            1.02 * probabilities[6][0]
            1.02 * probabilities[6][1]
            1.02 * probabilities[6][2]
        else:
            0.71 * probabilities[0]
            0.62 * probabilities[1]
            0.62 * probabilities[2]
            0.81 * probabilities[3]
            0.81 * probabilities[4]
            0.88 * probabilities[5][0]
            0.88 * probabilities[5][1]
            0.88 * probabilities[5][2]
            0.94 * probabilities[6][0]
            0.94 * probabilities[6][1]
            0.94 * probabilities[6][2]

    data = np.zeros(168)
    for i in range(7):
        # ignore days off
        if i < profile.days_off:
            continue
        data = simulate_day(profile, data, i, probabilities)

    return data


# Simulate one day of internet consumption based on the profile
def simulate_day(profile, data, day, probabilities):
    for i in range(24):
        if i < profile.hours_of_sleep:
            bandwith_needed = 0
        else:
            rand = random.uniform(0, 100)

            # if weekday
            if day <= 4 and (profile.in_school or profile.employed):
                # if employed and in school assume the job is part time
                if (profile.in_school and school_on) and profile.employed:
                    if i < profile.hours_of_sleep + 7:
                        if profile.virtual_learning:
                            bandwith_needed = 4
                        elif profile.hybrid_learning:
                            bandwith_needed = 2
                        else:
                            bandwith_needed = 0
                    elif i < profile.hours_of_sleep + 11:
```

```
                        if profile.employed_at_home:
                            bandwith_needed = 2
                        elif profile.employed_not_working:
                            bandwith_needed = run_iteration(profile, probabiliti
                        else:
                            bandwith_needed = 0
                    else:
                        bandwith_needed = run_iteration(profile, probabilities)
                elif profile.in_school and school_on:
                    if i < hours_of_sleep + 7:
                        if virtual_learning:
                            bandwith_needed = 4
                        elif hybrid_learning:
                            bandwith_needed = 2
                        else:
                            bandwith_needed = 0
                    else:
                        bandwith_needed = run_iteration(profile, probabilities)
                elif profile.in_school and not school_on:
                    bandwith_needed = run_iteration(profile, probabilities)
                else:
                    if i < profile.hours_of_sleep + 9:
                        if profile.employed_at_home:
                            bandwith_needed = 2
                        elif profile.employed_not_working:
                            bandwith_needed = run_iteration(profile, probabiliti

                        else:
                            bandwith_needed = 0
                    else:
                        bandwith_needed = run_iteration(profile, probabilities)
            else:
                bandwith_needed = run_iteration(profile, probabilities)

        data[24 * day + i] = bandwith_needed

    return data


# Determine bandwidth based off profile and probabilites
def run_iteration(profile, probabilities):

    bandwith_needed = 0
    for i in range(len(probabilities) - 1):
        num = random.uniform(0, probabilities[-1])
        if num <= custom_sum(probabilities[i]):
            if i == 2:
                if profile.ultra_hd_tv:
                    bandwith_needed += 25
                else:
                    rand = random.uniform(0, 100)
                    if rand <= 50:
                        bandwith_needed += 8
                    else:
                        bandwith_needed += 4
            # smartphone use is split between general, video, and audio
            if i == 5 or i == 6:
                general_probability = probabilities[i][0]
                video_probability = probabilities[i][1]
                audio_probability = probabilities[i][2]
                rand = random.uniform(0, sum(probabilities[i]))
                if rand <= general_probability:
                    bandwith_needed += 1
```

```
                elif rand <= general_probability + video_probability:
                    bandwith_needed += 5
                else:
                    bandwith_needed += 2
            bandwith_needed += required_bandwith[i]

    num = random.uniform(0, 100)
    if num <= profile.large_file_frequency:
        bandwith_needed += 50
    return bandwith_needed


data = np.zeros((100, 168 * 52))
for i in range(100):
    # Can take infinite profiles
    input_profile = Profile(30)
    # Can pass optional profile parameters
    input_profile.generate_profile()
    x, y = simulate_profiles([input_profile])
    data[i] = y

    plt.hist(y, color="skyblue", density=True, bins=50)


for i in range(300):
    sufficient_bandwith_counter = 0
    total = 0
    for j in range(100):
        # 0.5 gives a 50 percent margin to required bandwidth
        sufficient_bandwith_counter += sum((entry < i * 0.5) for entry in data[j])
        total += sum(entry != None for entry in data[j])

    print(sufficient_bandwith_counter / total)
    if sufficient_bandwith_counter / total >= 0.90:
        print("90 percent threshold") + str(i)
    if sufficient_bandwith_counter / total >= 0.99:
        print("99 percent threshold") + str(i)
        break

plt.show()
```

```
% Read the image and binarize
RGB = imread('regionA.png');
I = rgb2gray(RGB);

% gray scale adjustments for each region
I(I==171)=64.837;
I(I==57)=133.6;
I(I==76)=161.1070742;
I(I==221)=51.11779174;
I(I==153)=150.9148222;
I(I==104)=170.13649;

%empty space Fix
I(I==255)=0;

[A, B, C]=ndgrid(1:570,1:350,1:3);
d=[A(:),B(:),C(:)];

q = 1;

x0 = [1];
lb = [1];
ub = [598500];
fun = @(x)annealsolver(I,q,d,x(1))
[sim_mind,var] = simulannealbnd(fun,x0,lb,ub)
```

problem3_A.m problem3_B.m problem3_C.m annealsolver.m annealsolver_2.m annealsolver_3.m

```matlab
I = rgb2gray(RGB);
I = 255*im2double(I);

%empty Fix
I(I==255)=0;

%region fix
I(I==104)=368.5825861;
I(I==178)=263.9129888;
I(I==224)=124.2526123;
I(I==150)=185.2522986;
I(I==76)=232.6572387;
I(I==33)=426.2447896;
I(I==171)=265.2375289;


I = im2double(I);

[A, B, C]=ndgrid(1:772,1:858,1:3);
d=[A(:),B(:),C(:)];

q = 3;

x0 = [500000 750000 1000000];
lb = [1 1 1];
ub = [1987128 1987128 1987128];
fun = @(x)annealsolver_2(I,q,d,x(1),x(2),x(3))
options = optimoptions(@simulannealbnd, ...
                     'PlotFcn',{@saplotbestf,@saplottemperature,@saplotf,@saplotstopping});
[sim_mind,var] = simulannealbnd(fun,x0,lb,ub,options)
```

problem3_A.m problem3_B.m problem3_C.m annealsolver.m annealsolver_2.m annealsolver_3.m

```matlab
% Read the image and binarize
RGB = imread('regionC.png');
I = rgb2gray(RGB);
I = 255*im2double(I);

% empty Fix
I(I==255)=0;

% gray scale adjustments for each region
I(I==76)=196.8892147;
I(I==89)=110.9359019;
I(I==176)=111.3761237;
I(I==182)=161.5614127;
I(I==214)=142.5218184;
I(I==99)=122.7005092;
I(I==153)=98.90084145;


I = im2double(I);

[A, B, C]=ndgrid(1:495,1:388,1:3);
d=[A(:),B(:),C(:)];

q = 1;

x0 = [1];
lb = [1];
ub = [1987128];
fun = @(x)annealsolver_3(I,q,d,x(1))
[sim_mind,var] = simulannealbnd(fun,x0,lb,ub)
```

problem3_A.m | problem3_B.m | problem3_C.m | annealsolver.m | annealsolver_2.m | annealsolver_3.m

```
function x = annealsolver(I,q,test,a)
    a = round(a);
    pixelmiles = ((sum(I(:) ~= 255))/6.83)^0.5;
    A = test([a],:);
    low = [140 15];
    med = [500 2.5];
    high = [2000 0.75];
    for h = 1:q
        choice = A(h,3);
        if choice == 1
            for i = 1:570
                for j = 1:350
                    if ((A(h,1)-i).^2 + (A(h,2)-j).^2).^0.5 <= pixelmiles*low(2)
                        I(i,j) = (I(i,j) - (low(1)));
                    end
                end
            end
        end
        if choice == 2
            for i = 1:570
                for j = 1:350...
            end
        end
        if choice == 3
            for i = 1:570...
        end
    end
    I(I<=0)=0;
    x = mean(I,'all');
```

problem3_A.m | problem3_B.m | problem3_C.m | annealsolver.m | annealsolver_2.m | annealsolver_3.m

```
function x = annealsolver(I,q,test,a,b,c)
    a = round(a);
    b = round(b);
    c = round(c);
    pixelmiles = (275295/33.64)^0.5;
    A = test([a,b,c],:);
    low = [140 15];
    med = [500 2.5];
    high = [2000 0.75];
    for h = 1:q
        choice = A(h,3);
        if choice == 1
            for i = 1:772
                for j = 1:858
                    if ((A(h,1)-i).^2 + (A(h,2)-j).^2).^0.5 <= pixelmiles*low(2)
                        I(i,j) = (I(i,j) - (low(1)));
                    end
                end
            end
        end
        if choice == 2
            for i = 1:772
                for j = 1:858...
            end
        end

        if choice == 3
            for i = 1:772
                for j = 1:858...
            end
        end
```

problem3_A.m problem3_B.m problem3_C.m annealsolver.m annealsolver_2.m annealsolver_3.m

```
%sim annealing algorithm
function x = annealsolver(I,q,test,a)
    a = round(a);
    pixelmiles = (105422/33.64)^0.5;
    A = test([a],:);
    low = [140 15];
    med = [500 2.5];
    high = [2000 0.75];
    for h = 1:q
        choice = A(h,3);
        if choice == 1
            for i = 1:495
                for j = 1:388
                    if ((A(h,1)-i).^2 + (A(h,2)-j).^2).^0.5 <= pixelmiles*low(2)
                        I(i,j) = (I(i,j) - (low(1)));
                    end
                end
            end
        end
        if choice == 2
            for i = 1:495
                for j = 1:388...
            end
        end

        if choice == 3
            for i = 1:495
                for j = 1:388...
            end
        end
    end
```