

## Chapter 1 : Meet Frédéric Janssoone, God's traveling salesman --Aleteia

*He was paid little at first, but soon, thanks to his business acumen and talent for selling, he became a prosperous traveling salesman. His genius for business would always remain one of his gifts."*

Fortunately, humans are pretty good at this, we can easily work out a reasonably good route without needing to do much more than glance at the map. To find a single route, we first have to choose a starting location from the three possible locations on the map. This would mean there are  $3 \times 2 \times 1$  different routes to pick in total. That means, for this example, there are only 6 different routes to pick from. The number of possible routes is a factorial of the number of locations to visit, and trouble with factorials is that they grow in size remarkably quick! For example, the factorial of 10 is , but the factorial of 20 is a gigantic, So going back to our original problem, if we want to find the shortest route for our map of 20 locations we would have to evaluate different routes! Finding a solution Although it may not be practical to find the best solution for a problem like ours, we do have algorithms that let us discover close to optimum solutions such as the nearest neighbor algorithm and swarm optimization. In this tutorial however, we will be using genetic algorithms as our optimization technique. Creating a genetic algorithm for beginners Finding a solution to the travelling salesman problem requires we set up a genetic algorithm in a specialized way. For instance, a valid solution would need to represent a route where every location is included at least once and only once. To ensure the genetic algorithm does indeed meet this requirement special types of mutation and crossover methods are needed. One type of mutation method we could use is swap mutation. With swap mutation two location in the route are selected at random then their positions are simply swapped. For example, if we apply swap mutation to the following list, [1,2,3,4,5] we might end up with, [1,2,5,4,3]. Here, positions 3 and 5 were switched creating a new list with exactly the same values, just a different order. In this crossover method we select a subset from the first parent, and then add that subset to the offspring. Any missing values are then adding to the offspring from the second parent in order that they are found. To make this explanation a little clearer consider the following example: Next, the missing route locations are adding in order from the second parent. This process continues until the offspring has no remaining empty values. Creating the Genetic Algorithm In literature of the traveling salesman problem since locations are typically refereed to as cities, and routes are refereed to as tours, we will adopt the standard naming conventions in our code.

**Chapter 2 : The Descent | Tales of a Travelling Salesman**

*The traveling salesman problem can be solved by a nondeterministic polynomial-time algorithm. It is not known if it can be solved by a deterministic polynomial-time algorithm - but it can also be solved by a deterministic algorithm taking exponential time.*

These are simply our x, y coordinates. We want to minimize route distance, so a larger fitness score is better. Based on Rule 2, we need to start and end at the same place, so this extra calculation is accounted for in line 13 of the distance calculation. Create the population We now can make our initial population aka first generation. To do so, we need a way to create a function that produces routes that satisfy our conditions Note: To create an individual, we randomly select the order in which we visit each city: This is as simple as looping through the createRoute function until we have as many routes as we want for our population. Subsequent generations will be produced through breeding and mutation. Determine fitness Next, the evolutionary fun begins. Our output will be an ordered list with the route IDs and each associated fitness score. Fitness proportionate selection the version implemented below: The fitness of each individual relative to the population is used to assign a probability of selection. Think of this as the fitness-weighted probability of being selected. A set number of individuals are randomly selected from the population and the one with the highest fitness in the group is chosen as the first parent. This is repeated to chose the second parent. Another design feature to consider is the use of elitism. With elitism, the best performing individuals from the population will automatically carry over to the next generation, ensuring that the most successful individuals persist. In lines 3â€”5, we set up the roulette wheel by calculating a relative fitness weight for each individual. In line 9, we compare a randomly drawn number to these weights to select our mating pool. Ultimately, the selection function returns a list of route IDs, which we can use to create the mating pool in the matingPool function. Now that we have the IDs of the routes that will make up our mating pool from the selection function, we can create the mating pool. However, the TSP is unique in that we need to include all locations exactly one time. To abide by this rule, we can use a special breeding function called ordered crossover. In ordered crossover, we randomly select a subset of the first parent string see line 12 in breed function below and then fill the remainder of the route with the genes from the second parent in the order in which they appear, without duplicating any genes in the selected subset from the first parent see line 15 in breed function below. Illustration of ordered crossover credit: In line 5, we use elitism to retain the best routes from the current population. Then, in line 8, we use the breed function to fill out the rest of the next generation. Mutate Mutation serves an important function in GA, as it helps to avoid local convergence by introducing novel routes that will allow us to explore other parts of the solution space. Similar to crossover, the TSP has a special consideration when it comes to mutation. Again, if we had a chromosome of 0s and 1s, mutation would simply mean assigning a low probability of a gene changing from 0 to 1, or vice versa to continue the example from before, a stock that was included in the offspring portfolio is now excluded. This means that, with specified low probability, two cities will swap places in our route. Next, we can extend the mutate function to run through the new population. First, we rank the routes in the current generation using rankRoutes. We then determine our potential parents by running the selection function, which allows us to create the mating pool using the matingPool function. Finally, we then create our new generation using the breedPopulation function and then applying mutation using the mutatePopulation function. All we need to do is create the initial population, and then we can loop through as many generations as we desire. Running the genetic algorithm With everything in place, solving the TSP is as easy as two steps: First, we need a list of cities to travel between. Then, running the genetic algorithm is one simple line of code. This is where art meets science; you should see which assumptions work best for you. With a simple tweak to our geneticAlgorithm function, we can store the shortest distance from each generation in a progress list and then plot the results. Run the GA in the same way as before, but now using the newly created geneticAlgorithmPlot function: Sample output from the geneticAlgorithmPlot function Conclusion I hope this was a fun, hands-on way to learn how to build your own GA. Try it for yourself and see how short of a route you can get. Or go

## **DOWNLOAD PDF GODS LITTLE TRAVELLING SALESMAN.**

further and try to implement a GA on another problem set; see how you would change the breed and mutate functions to handle other types of chromosomes. End notes You can find a consolidated notebook here.

Chapter 3 : home | Comedy Central

*I was desperately looking for any advice from other moms/wives on being the wife of a traveling salesman. My husband and I have been together for 3 1/2 years, and married for 7 months. I was a single mom for years, and what I imagined of marriage to a Christian man has been a lot different than what I have gotten.*

You can play around with it to create and solve your own tours at the bottom of this post , and the code is available on GitHub. How does the simulated annealing process work? We start by picking an arbitrary initial tour from the set of all valid tours. There are so many valid toursâ€” 47! But a well-designed annealing process eventually reaches a solution that, if it is not the global optimum, is at least good enough. Start with a random tour through the selected cities. Pick a new candidate tour at random from all neighbors of the existing tour. If the candidate tour is better than the existing tour, accept it as the new tour. If the candidate tour is worse than the existing tour, still maybe accept it, according to some probability. The probability of accepting an inferior tour is a function of how much longer the candidate is compared to the current tour, and the temperature of the annealing process. A higher temperature makes you more likely to accept an inferior tour. Go back to step 2 and repeat many times, lowering the temperature a bit at each iteration, until you get to a low temperature and arrive at your hopefully global, possibly local minimum. The key to the simulated annealing method is in step 4: As we turn the temperature down, we accept fewer longer tours and eventually we converge on the globally optimal tour. Why not do the same process with 0 temperature, i. Histograms of the results for 1, trials of the traveling salesman through the state capitals show that simulated annealing fares significantly better than hill climbing: The results via simulated annealing have a mean of 10, miles with standard deviation of 60 miles, whereas the naive method has mean 11, miles and standard deviation miles. And so, while you might not think that Nikolay Chernyshevsky or Chief Wiggum would be the best people to offer an intuition behind simulated annealing, it turns out that they, along with cliché-spewers everywhere, understand the simple truth behind simulated annealing: Give it a shot below! Bonus points if you recognize where the default list of cities comes fromâ€” The app is hosted at ShinyApps. Posted by Todd Schneider.

**Chapter 4 : A Forest in the Appalachians | Tales of a Travelling Salesman**

*Delaney Davidson, whose album release tour starts today, talks to Henry Oliver about storytelling, collaborating and his new album Shining Day.. Delaney Davidson sees himself as a travelling salesman.*

In what order should a salesman visit his prospects? The traveling salesman problem may appear simple but it has engaged some of the greatest mathematical minds and today engages some of the fastest computers. This makes new findings, that bees compute their route before pollinating flowers, all the more remarkable. Colburn was a nineteenth century American child prodigy. As a young boy he could multiply large numbers in an instant, compute square roots of large numbers, and determine the number of hours or seconds between distant dates. But the local talent was not always a walk-over. While in Dublin, Colburn went up against the twelve year old Hamilton. While Colburn mostly was victorious, the older Irish boy was sometimes the faster of the two. The matchup was probably the first time that Hamilton, who knew a dozen languages by that time, had been shown up in anything intellectual and it may have motivated him. Hamilton went on to become one of the greatest mathematicians in history. He is probably most famous for his reformulation of Newtonian mechanics and today the Hamiltonian operator is well known to physicists everywhere. Hamilton also discovered quaternions while on a Sunday stroll with his wife. The fundamental relationships came to him in a flash and he immediately carved them into the Brougham bridge so as not to forget them. Though quaternions have rather narrow applications today, where used they are quite valuable in providing faster and more robust computations. For instance, from molecular dynamics to computer graphics to spacecraft navigation, coordinate transformation algorithms are required. Quaternion-based transformations both minimize the number of multiplications and avoid singularities. Hamilton also was interested in early forms of the traveling salesman problem. The problem was well known to salesmen of the day, but had not yet been formalized in mathematics. Hamilton invented a mathematical puzzle called the icosian game in which the objective was to create a path that visits twenty points without doubling back on itself. The traveling salesman problem is interesting, and important, because the seemingly obvious strategy of simply moving to the next closest location after each visit does not generally work. The cost may be directly proportional to the distance traveled, or it may be more complicated. For instance, the cost of travel may vary by location and direction of travel going uphill versus downhill, for instance. In computer science the problem is known to be what is called NP-hard. All of this makes new findings, that bees regularly solve their own routing problem, rather remarkable. As one report explained: Bees can solve complex mathematical problems which keep computers busy for days, research has shown. The insects learn to fly the shortest route between flowers discovered in random order. Bees manage to reach the same solution using a brain the size of a grass seed. They visit flowers at multiple locations and, because bees use lots of energy to fly, they find a route which keeps flying to a minimum. After exploring the location of the flowers, the bees quickly learned to fly the best route for saving time and energy. How does the bee, with a brain the size of a grass seed, optimize its route? But the bee does solve its routing problem. Perhaps, but this certainly is not a fact as evolutionists insist it is. Religion drives science, and it matters.

**Chapter 5 : FastGames - The Travelling Salesman**

*Comedy Central Jokes - Traveling Salesman - A traveling salesman approached an old farmhouse and noticed the strange behavior of the couple inside. The woman was running the lawn mower over the carpet and the man had one hand dipped in a fish bowl and was playing with himself with the other.*

The ground was much colder than Robert remembered, and harder too. The air hurt to breathe for some reason. He looked around at the park. It was not the park. Robert James was just asleep against a stone that jutted out of the side of a hill, or was it a mountain? A real fear that shot through his body, from his toes to his fingertips. What is going on? Robert whirled around to get a look at the voice, higher up than him on the hill behind him. It was a mountain of a man, bearded and dressed for the weather: How did I get here? The man hopped down, one leg at a time landing steadily where he bounded. His brown boots made distinct thuds with each landing. He stood in front of R. I told you man: How else could I be here? Might as well enjoy it while it lasts. Robert laughed a bit, nervously. Snow laced between leaves on the ground and floated slowly from above like dandelions. It all looked very real. The cold made his bones ache with each step. Maybe it was the one that he kept on having, and he was finally lucid enough to really experience it. He was right about one thing: He would always, always remember. The cold really made his lungs hurt, and multiplied the exhaustion that was setting in. Mountains had a way of making the sun vanish, he noticed grudgingly. This better not turn into some damned nightmare. Without warning, the running was over. The man, still with his strange boyish grin, looked on as Robert bent over with his hands on his knees gulping in air like a fish out of water. He was looking for something, perhaps? Robert paid hardly any mind. I work for a living! It was modest, wooden everything of course. Dirty shoes by the door, all of the windows had shutters. Strikingly thick and sturdy looking shutters, actually. The man bolted the door shut with several different locks, and a bar over the door. Bears must be an issue here, deep in these woods. Rather large teeth, Robert observed. Not sharp or anything, but Robert was noticing now that his mouth was actually quite big. This is a dream, after all! A hot bowl of stew was in his hand, and he felt really comfortable in the big couch. There was a fire roaring surrounded by a big grey stone hearth. The man looked nervous. He walked down the hall past the kitchen and out of sight. Some sounds of slamming, suddenly. Loud, followed by sounds of metal moving and clicking into place. The man came out of the back with a large hunting rifle. HE gets to enjoy whatever show his mind has conjured up for him now. Finally some real amusement! The door slammed behind the man, and R. He then unlatched one window to the right of the door, and peeked outside. The fireplace was at his back and he could feel its warmth radiating behind him. The snow was falling harder outside now. He could see the man with his flashlight running off into the darkness like he knew where he was going. But he was also moving differently than before. After about 5 minutes the man was making his way back, carrying some strange shape in his hands. The light from the window reached them now, and the man noticed and looked up in anger, bellowing: The child was missing an arm, and he was bleeding everywhere. It was all over the man and now the floor, and they slid in it a little as the man laid his son on the couch, taking his hat off to try and stop the bleeding. The man looked over his left shoulder at him, the eye Robert could see was glassed over from fear. The mountain of a man was trembling with terror. Robert saw now something he failed to notice before. Robert James went quickly now to shut the window, legs moving without hesitation. He glanced outside as the shutter moved to close, and he could see a pair of eyes reflecting the light leaking out the window, low to the ground. Hanging in the darkness like glow flies, they vanished. He latched the window as quickly as he could and backed away in horror. Unknowingly he reached for the rifle, which the man had tossed on the ground. Itâ€¦ Saw meâ€¦ Itâ€¦â€¦â€¦ Did this? His mind was numbed, he had never felt fear like this. Starting high, then dropping into a lower tone. Then one more began, then another repeatedly until a cacophony of alien cries echoed off of the falling snow and the trees. The moon watches quietly, with the sadness of the elderly. Like it had watched these wretched children of nature grow, and yet did nothing. Like it had lived with regret. There were scratches at the door now. This sound he heard was unlike anything he had heard before. He cocked the rifle and flicked the safety off. There was work to do.

**Chapter 6 : The traveling salesman and the farmer's wife : Jokes**

*Traveling Salesman Problem The Travelling Salesman Problem (TSP) is the most known computer science optimization problem in a modern world. In simple words, it is a problem of finding optimal route between nodes in the graph.*

In the previous chapter our hero met his inevitable and much deserved fate at the sword-arm of honorable Uthgerd. And they all lived happily ever after. Next chapter I closed my eyes and prepared for the mind-shattering pain, followed by eternal darkness. But the pain did not come, and my mind remained in one piece. Her footsteps punctuated the soft patter of blood dripping into the chamberpot under my bed. There was no response. I opened my eyes and craned my neck as much as the sword sticking out of my chest would allow. Uthgerd was standing in the doorway, facing a man I could not see. But I could smell him, and I realized I could also hear his heart pounding through the wall. In fact, from the cacophony of heartbeats there had to be several people lined up in the staircase and the common room below. I had not come this far to end up as the village idiot. That position was taken, anyway. Uthgerd collected her thoughts, and answered the man. The man at the top of the staircase stepped into the room, and I recognized him as the randy bard I had knocked down a peg when I first came to Whiterun. He looked at me and winced. Still not giving you a tip. I could smell fear wafting up through the cracks in the floor. The bard reached for his dagger, but Uthgerd stopped him with a determined look on her face. He belongs to me! I only mean to atone for my foolishness! I was the one who let him defile the shrine of Stendarr, he must die by my hand! The bard turned towards Uthgerd and pulled his dagger. I smiled at the prospect of watching him get a clobbering, but she only stepped back and raised her hands. He stabbed at her neck, but she dodged it. I put my hands on the hilt of the sword, and pushed upwards. It gave way with almost no effort, and I continued hauling it up by the flat of the blade until it plopped out of my chest and fell down next to me. The bleeding had stopped and I could feel how my flesh was already knitting itself back together. My stomach pains were gone, in fact I felt fit as a fiddle and had no aches at all. As soon as I woke up I would have to roll an industrial-sized barrel down to that glowing tree; this substance had the potential to make me wealthy beyond my dreams! No doubt it could be diluted into a finished product without losing too much potency, though it would take some experimenting to get the dosage right. I would also need to procure an industrial-sized barrel. Uthgerd dodged another feeble stab, and threw a worried glance at my bed. She noticed me sitting upright and cried out in anger, but I was ready. I focused my thoughts on the invisibility spell I had purchased from Farengar, and to my surprise it worked on the first try. Uthgerd kicked the bard away, drew her sword and lunged towards me, but I rolled sideways and darted out on the balcony overlooking the common room. I grabbed hold of one of the ceiling rafters in mid-air and pulled myself up with ease. Uthgerd was not so lucky. She got up on the railing to jump after me, but it collapsed under her weight and she tumbled over the edge, landing hard on the floor below. From my vantage point I could see the crowd turn against her, and I sensed their confusion and fear refocusing itself into anger. The bard rushed down the stairs and joined in the fight. I felt no joy in watching this. I much prefer a due process involving bribable judges and underpaid soldiers. However, if I were to reveal myself they would surely gang up on me instead, and I was curious to see what else the magical tree god had in store for me. I lowered myself down from the rafter and landed on the floor without a sound, then scuttled invisibly through the room and up the stairs. Back in my quarters I grabbed the travelling clothes I had laid out the night before and started putting them on. Interacting with any object will break the illusion! But I did notice that my invisibility had indeed worn off, and wondered if the voice was just a manifestation of my subconscious. Still, nobody who buys a leisure trip wants a backseat driver along for the ride. I decided to leave my weapons and armor behind, and grabbed only my pickaxe and beloved coinpurse before I recast the invisibility spell. To my great annoyance it was nowhere near as easy this time around, and the effect occasionally flickered like a candle in a draft, but it would have to do. I crept down the stairs until I could see what was going on in the common room. Thankfully my invisibility held up as I made my way down the stairs and exited through the kitchen door. Outside it was pitch dark, but my eyes adjusted themselves astonishingly quick, and within seconds I could see as if it was bright daylight. Opening the door had made me visible again, so I figured I

might as well stroll down to the gates as if nothing had happened. Probably the night watch would be half asleep as usual, and for all they knew I was just a drunken taverngoer headed outside for a piss. This proved to be wishful thinking. The guards were a lot more plentiful and alert than I had ever seen them before, and they all carried torches. I pulled up my hood and hid my face as best I could, but to no avail - one of them spotted me, and he immediately blew his horn. A mass of armored boots came clattering down from the upper city, and I could hear the door to the inn open behind me. I have no idea what the city watch puts in the rations to give them such a superhuman aim, but I suddenly regretted breaking my vow not to piss them off. I could feel the tip scratching the inside of my skull, and the shaft stuck out several inches in front, making my head look like a morbid glazed apple on a stick. The mob charged as one, and I bolted down the main street and out the gates, dodging more crossbow shots and even an occasional fireball along the way. Farengar must have joined them, and I cursed myself for selling him that staff, but obviously I had done it with the best intention. Specifically, the intention of taking his money. I was too fast for any of them to have a chance at catching up, and they soon dropped the chase, but one of the guards kept pursuing me into the wilderness. He was in full armor though, and before long he had collapsed, leaning on his knees as he looked around with a frustrated expression. As I moved closer I saw that it was the same guard who had shot me, and I figured he probably wanted his bolt back, so I pulled it out with a sickening squelch and threw it at him. It missed by a mile, but the eyeball slid off in flight and landed on his boots with a wet plop. Come over here and fight me, you accursed fiend! I was about to start running, when the voice spoke again. However, this voice was starting to make me seriously angry, so I yelled back at it. I left my lute and bouquet of flowers back at the inn! Use those powers and dominate his mind so you can feed on him! Assuming you are able to think without moving your lips. The effect was immediate. And I shot you and everything! You poor fellow, you must have been so scared, what with your frail physique and total lack of combat skills. And us chasing you out of town and all, oh what a lot of beasts you must think us! I patted him awkwardly on the back. I hurried away from the man and continued down the road. Since I was in no actual danger at the moment, I decided I might as well scout out the road to Riften. Of course I would only be seeing what my imagination could conjure up, since I had never actually been there, but I was itching to stretch my legs after three days in bed. Besides, my dreams of becoming the kingpin of Skyrim had to wait until I could muster the resources to deal with those pesky giants that guarded the tree. Uthgerd had made them out to be damn near invincible, but Uthgerd was dead. Surely they could be enticed to leave, or failing that, forced away by other means. But those means were no longer available to me in Whiterun, dreaming or not. Riften, on the other hand, seemed like an ideal cradle for a narcotics empire. As long as I made sure to weed it for thieves and other trash, there my seed would grow into a mighty tree of commerce, from which my purple juice would flow until everyone was swimming in it. My name is Revialcantar, and you may have heard of me. Might I inquire as to how-" "Yeah, yeah, put a sock in it. The reason you exist is that three days ago I drank a bottle of magical tree sap, which I found to be an effective remedy for the mother of all stomach bugs, but also turned out to have potent, late-onset mind altering effects. Ah, you mean from the Sleeping Tree tended by the Giants! I can assure you, as wondrous as its properties are, it is no hallucinogenic. You are indeed walking the firm earth of Mundus right now. Regular spot of fun, really. Perhaps I could saw my own head off a few times and make a living as a juggler. Hopefully it would shut him up for a while. The torches turned out to belong to a group of heavily armed knights afoot, chasing a half-naked woman through the wilderness. He swung his sword at her, but she dodged out of the way with impressive alacrity.

**Chapter 7 : Evolution of a salesman: A complete genetic algorithm tutorial for Python**

*Traveling Salesman Problem is an extremely important problem in operational research. We first define the problem and then we study the methods and algorithms to solve the TSP.*

The trees silently watched as more creatures jumped through their canopies toward the cabin, a skittering flood of claws and teeth. The moon broke through the clouds, shining sporadically onto grey skin with patchy white fur. Forlorn, forgotten by nature. Maybe it was because they were hungry. The mountain man cowered on the floor, covering his ears and weeping openly. The door rocked on its hinges, straining against the wooden bar holding it in place. Filling the air was the sound of horror, and an overwhelming sensation of inevitable death flooded the cabin. It was his dream, of course, and he could be as heroic here as he was pitiful in the real world. The only thing holding him back was his fear of failure. His bones creaked and his back ached terribly, but he ignored it all because he was the hero now. Robert James, hero-man for hire. His blood was electric in his veins, pulsing with a heat that shielded R. The door was rhythmically rocking in and out now, and each window was shattered one by one while Robert was thinking how pleased he was with himself. The heavy wooden shutters now strained against their metal latches with the force of the wild itself pushing inward. This wilderness " the darkness " was hungry for Robert. The thick wooden bar over the door was cracking, bit by bit. Any minute now they would be coming through, and Robert raised the rifle with his elbows in tight, just like his father taught him in the forest as a boy. Look down the sight, breathe in, and fire as you exhale, son. Get something to protect yourself and the boy. His pale face with his ice blue eyes stared blankly at the ceiling, the hole where his arm was only slowly dripping now. Something that kept him alive had doomed him. The blood from the slashes on his back steadily flowed into it, turning it dark. He began to shuffle slowly toward it. Get something heavy to swing! We can beat them. And besides, the ground is too hard this time of year to bury anything. What was I trying to prove? The same eyes that his son had. The door finally tore like wet paper, and the creatures clamored into the cabin. Large round eyes, like orbs. Round head hung low and looking at him, it snarled and showed a large mouth that curved around its head, and layers of teeth. Its backbone stretched its skin in jagged growths. The distinct smell of pennies blended with the putrid odor of the other creatures that darted into the cabin. What kind of dream is this?

**Chapter 8 : The Forsaken Westerns – “ The Traveling Salesman –” tv show – “ Westerns TV**

*Now Jim was traveling the back woods of Iowa, looking for someone needing a vacuum. Now late that afternoon, as it was about to happen, his car broke down, leaving him stranded. Now, luckily Farmer Jones came along on his old John Deere, and Jim explained what had happened.*

We generally only support the major desktop browsers like Chrome, Firefox, Safari, and Edge. Use this one at your own risk! The Travelling Salesman Problem Oct 23, The Travelling Salesman Problem The Travelling Salesman Problem can be solved by determining the quickest route that visits each provided plot and returns to where it started. For example, here are five plots with the optimal route and one of many suboptimal routes: Visually, it is easy to tell what the optimal route is for this field of plots; I can solve this thing with my eyes! Programmatically, this cannot be truly solved without using permutation in an algorithm. Exact algorithms use different optimizations to reduce permutations, but no one has actually solved it. There are many heuristic algorithms that calculate suboptimal paths much quicker than exact algorithms, but cannot prove they are the most efficient path and are often far from it. I have chosen to play with the Travelling Salesman Problem in my spare time over the past two years and continue to have fun with it today. This post is not going to present anything more than my currently incomplete thoughts. If I waited until I solved the problem before blogging about it, there would be an infinitesimally small chance I would ever blog about it at all. Unknown things have few rules. It is like shooting a ball into a hoop without the game of basketball being invented yet. But if Michael Faraday reinforced any ideology, it is that amazing things can come from thinking outside of convention. Some might see this process as sisyphean –” relentlessly unending and repetitive. I see it is as an opportunity to think for myself. How would I go about solving this problem? It is an opportunity to explore uncharted land. Perhaps I will land somewhere interesting and discover a new heuristic or optimization At this point, neither outcome matters much to me. The one thing I do know is that throwing yourself into something for the sake of exploration is never a bad thing thing if you are passionate about it –”and I am passionate about the TSP. One of many reasons why the TSP is so difficult is because of a complexity that is observable in an extremely simple case. Here we have an optimal path. We have two paths, one without and one with plot brown. When we add brown in the upper right, purplebluegreen becomes purplegreenblue. But look what happens if we remove the bottom right yellow: All this from removing yellow. So how the hell do you determine whether or not you should go to green from plot purple? This is a simple example of something that only gets more complicated the more plots you have. If there is a center, you can determine what is on the perimeter, what is central, and what is in-between. This is crucial, because the behavior of a path clearly changes based on where plots lie in relation to it. In a way, outer plots take priority over inner plots. Here I have added a white point that is in the center, or centroid of the set. Any future visualization in this post will visualize the centroid this way. This is the order of plots, furthest from centroid to the closest: My solution is the generated topmost image in that pen and the preexisting optimal path is the last image. Not to mention it uses JavaScript in a browser. Take that, super computer. When I returned to the problem, I started thinking about the problem differently. These permutations would get more and more computationally expensive the more plots in the set. That is when I realized that if I could group plots based on their location, my permutations would be significantly reduced. But how would I group them? After some time it finally clicked –” I could round plot positions. For example, lets say I have four plots with x and y coordinates between 0 and 1. Clearly, the two groups are redpurple and bluegreen. But how do we express that in a program? We simply round the positions at a scale. In this case, we will round each plot coordinate to the closest 0. The same is true for [0, 0]. That is why they are in group I took a TSP set of plots and animated this idea in a pen. I group at different scales in sequence from a very small group size to a very large one eg. All plots move towards whatever group they are being rounded into. See the Pen Travelling Salesman Sketches: Use the slider to change the quantity of plots: The cool thing about rounding everything to a grid is that you have known coordinates and known distances. For example, if I moved every plot its closest corner of the square field, I would have at most four groups, know the distance between the corners, and easily path around

them. If I applied the same logic inside each group at the next smallest scale, I could do this recursively until there were no more groups. I have yet to use this to build a solid heuristic, but frequently use the principles of it in my sketches. The same is true with groups of two points  $ab$ ,  $ba$  or one point  $a$ . I know this is mind-blowing. If only there were only three or less cities in the world. This pen groups plots in threes, then those groups in threes, then those groups in threes and so on. The thinking being that you would connect each group and solve the problem. It is an incomplete idea, but definitely has some room for exploration. Where do we start drawing the path? Westerners may choose the top left red simply because of how they read words, others may start at the top right purple, contrarians may choose the bottom right blue, anarchists might flip two coins. No one would be wrong starting from any plot. Exact algorithms cannot afford free will. We need to establish a rule to break ties. This led to my handy concept of Radial Bias. I would assign each point a radial value based on its relative position to the centroid. If it shared proximity from the center with another plot, preference would be determined by this radial position. Two plots can live at the same distance and they can share the same radial degree, but they cannot share both. As soon as I started programming with this bias, I discovered it gave me a starting point. Once you establish a starting point and a bias, you create the idea of direction. When you have a start and a direction, you have a future. For example, our square above is easily solved by starting at a point then rotating either clockwise or counter clockwise just choose one. That chosen direction of rotation is your bias. If there is a solution for the TSP, I am fairly confident it requires a radial bias and will work in either clock direction. Using this radial bias, you can easily create a complete path for any set. You just connect each plot after sorting by radial position from the centroid. With small amounts of plots, you get some fairly efficient paths. With large amounts of data however, it totally falls apart. Drag the slider to change the quantity of randomly generated plots: Inside of a program, future ends up being a fairly evasive concept. I hardly know what I am talking about, but I will attempt to convey here, starting with this very crude drawing whose eraser marks showcase my ignorance: I am drowning in the ocean and there is a lifeguard who is deciding the quickest way to get to me. The lifeguard can run on sand quicker than they can swim in water. Path A is the direct route, which would be the optimal path if this were all on land. Path C is the optimal path as it reduces time in the water as much as it can without adding so much distance that it will take more time. In terms of sheer distance, shortest path would be A, the least swimming time would be B, but the quickest path to get to me the one that saves my life is C. I am not drowning at the end of a light beam. Does a light beam have a destination? Does it choose where it lands? Apparently ants of all things know the answer. Quite literally, ant colonies are smart enough to solve math problems without understanding math. Ok, so we use ant logic to solve math problems. I started fiddling around, and the pieces started to fit together.

*This article is about a little-known piece of FME functionality that lets us deal with a classic GIS task: the Travelling Salesman Problem.*

When I am asked what it is that my husband does for a living, I usually answer by saying that he sells something that has to do with lights. A co-worker of his came for dinner over the weekend and I got to see a different side of my husband. Sometimes I forget that he has a life outside of the walls of our house. For the most part, he knows who I know and he sees where and how I work. The weeks that he is away on business are long. And often times I get irritated or bummed when I find out that he has to go away again. However, these weeks usually serve as lessons in contentment in my life. The Lord teaches me this lesson a lot. Maybe because I never quite "get it! I usually fail the test, which is why I am taught the lesson time and time again. When I think about the early years of marriage, I can see just how far we have come. We always had food, we always made rent, had cars that ran, tithed and had jobs. We were also blessed with the generosity of family who let me tag along on trips to Disney or taking us on trips to the beach. And then there was my husband, who knew I loved my job which paid diddly squat and never forced me to find something that paid more. A husband who delivered light bulbs during the week, worked at a gym in the evening and washed cars on the weekend so that I could do what I love. It taught me about hard work and sacrifice, what going without is all about, all so I could do life with my friend. Six and a half years later and my husband is still busting it for me I will forever be thankful to my husband for all of his hard work so I can stay home with our babies. Life has gotten a little more comfortable over the years but I know it could all go away in a blink too. I hope that this post serves as a "cheat sheet" for ME when I find myself in the middle of one of my contentment lessons again.