A CRITERIA-BASED APPROACH TO THE TRAVELING SALESMAN PROBLEM (TSP)

Business Analytics and Data-Driven Decision Making

Daryl Ono, PhD, Department of Business and Economics, Marymount California University, 30800 Palos Verdes Drive, East Rancho Palos Verdes, California, (310)-377-5501, dono@marymountcalifornia.edu
Jose Rincon, MLS, MBA, Head Librarian, Marymount California University, 30800 Palos Verdes Drive, East Rancho Palos Verdes, California, (310)-377-5501, jrincon@marymountcalifornia.edu
Althea Thomas, MBA, Lead Tutor, Marymount California University, 30800 Palos Verdes Drive, East Rancho Palos Verdes, California, (310)-377-5501, athomas@marymountcalifornia.edu

ABSTRACT

The "traveling salesman problem (TSP)" is a classic minimum cost network flow problem in mathematical programming and graph theory that can be formulated in multiple configurations. The fundamental question, however, is: "what is a cost"? The original "traveling salesman problem (TSP)" defines distance as the cost and the objective is to minimize distance traveled. This paper proposes other "cost" criteria to the original problem and also proposes a maximum revenue network flow as a variant to improve managerial decision-making. The proposed decision table methodology can be applied to problems that involve multiple locations or multiple tasks to complete.

Keywords: traveling salesman problem, NP-complete, profit maximization, greedy algorithm, distance minimization, time minimization

PROBLEM DEFINITION

The "traveling salesman problem (TSP)" is summarized by the following: a list of destinations and the distances between each pair of destinations, determine the shortest path that visits each destination only once and returns to the origin destination. The traveler begins at a given destination, visits each destination only once. The task is to develop an algorithm or heuristic to find the shortest route possible. Businesses, governments and militaries are attempting to determine the solution to this problem to increase the efficiency of their supply chains and to minimize logistic costs. Current algorithms can solve this using combinatorial optimization but the computational complexity is high and therefore requires significant computer resources to determine the optimal solution [28] [29].

The "traveling salesman problem (TSP)" is an obvious application of a minimum cost network flow but this fundamental problem can be applied to other disciplines. Some other applications include the wiring of a computer network, drilling of printed circuit boards, school bus routing, printing press scheduling, and many others [6] [17].

The "traveling salesman problem (TSP)" is NP-complete in terms of computational complexity because an exhaustive search algorithm is used to determine the optimal solution. Heuristics and/or quick solutions to NP-complete are still undeveloped so the "traveling salesman problem (TSP)" becomes computationally complex when there are many destinations.

“NP stands for "Non-Deterministic Polynomial". It represents a type of problem whose solution can be determined using a "non-deterministic" computer. That's a computer where when there's a branch, such as an "if statement", the computer takes both branches at the same time. An NP problem currently can't be solved on a standard computer in polynomial time, although the non-determinism can be simulated. Unfortunately, the simulation takes exponential time, and as the problem size grows, the amount of time it takes to solve the problem grows too large. There are a lot of NP problems that people would like to solve in a reasonable, polynomial amount of time, but right now, no one knows how.” [25]

Basically an NP-complete solution would be to calculate all possible solutions then choose the solution which minimizes the distance based upon all combinations. At this time there is no way to go straight to the optimal solution [12] [16].

The researchers of this paper understand that the “traveling salesman problem (TSP)” is NP-complete and is therefore computationally complex. The researchers cannot propose a solution to an NP-complete problem but they would like to present a method so that the optimal path of the “traveling salesman problem (TSP)” is truly the optimal solution. This involves understanding what the true objective is and to formulate the “traveling salesman problem (TSP)” to determine the true optimal solution based upon the correct objectives or criteria, not just distance minimization.

The “traveling salesman problem (TSP)” is not limited to a minimum cost network flow solution only and can incorporate other objectives or criteria. The proposed audience of this research is those governments, businesses and militaries that have about ten or less clients to service within one day. The researchers do not intend to improve on the “traveling salesman problem (TSP)” algorithms currently in use by the large package delivery services, such as United Parcel Services (UPS), Federal Express (FedEx) or Dalsey, Hillblom and Lynn (DHL). UPS, FedEx or DHL are global powers that are admired throughout the world and they have the financial resources to procure the best computer equipment, programmers, software and computer scientists.

“Federal Express Corporation has used operations research (OR) to help make its major business decisions since its overnight package delivery operations began in 1973. An early failure pointed out the need for scientific analysis. Subsequently, a successful origin-destination model followed by models to simulate operations, finances, engine use, personal assignments, and route structures influenced the conduct of business during periods of substantial growth. There were many false starts between the successes.” [18]

“FedEx is migrating to a dynamic route mapping system that has several objectives, some of the more prominent being balancing on-road time between routes, maximizing efficiency of each route and achieving service commitments. In most cases the system determines the stop order for the route, however, couriers are expected to use their experience and better judgment if they deem a particular stop order to be less-than-optimal……...The system has been rolling out for some time now, with some success, but not all areas are a good fit for its mapping features, especially rural routes, so in many cases the actual route taken will be determined by the courier.” [4]

FedEx understands the computational difficulties of the “traveling salesman problem (TSP)” and chooses to attack this problem using dynamic programming. FedEx also understands that the “traveling salesman problem (TSP)” could involve more than distance minimization. Dynamic programming is a powerful tool but is computationally complex and generally requires significant computer resources to implement the system. The researchers understand the power of dynamic programming but they will skip this potentially viable solution.

“We (UPS) use a proprietary dispatch software. The base dispatch software is not new; we have had it for some time. I do not know exactly when we began using it, but I know it has been around for some time, at least since 2000 or so. ORION is essentially an additional piece of software designed to coexist with the dispatch system which when fed certain perimeters will determine an optimal route. I don't know much about our third part logistics division (UPS Supply Chain Solutions), but they are there to help deal with these types of issues." [26]

"Called ORION, or On-Road Integrated Optimization and Navigation, UPS' data-drenched route optimization tool aims to deliver the best answer yet to the traveling salesman problem, the classic computational conundrum that shows just how hard it is to find the shortest distance among a series of points on a map. The size of the numbers involved means simple arithmetic is out. Instead, ORION depends on heuristics, the field of math and computer science devoted to finding answers that are good enough, and that get better based on past experience. Of course, finding the shortest distance is only one of many variables in play for UPS. Promised delivery times, different types of customers and the types of packages being delivered and picked up are just a few of the additional factors ORION must take into consideration. And Levis is quick to emphasize that UPS doesn't discount the value of driver wisdom accumulated during years on a route. The best system, he says, is one that relies on both human and algorithmic intelligence, not just one or the other." [27]

It seems like FedEx uses dynamic programming to attack this problem whereas UPS uses heuristics in its ORION computer software to solve the "traveling salesman problem (TSP)" in their package delivery services. The difficulty is that their computer software is proprietary so the researchers cannot access their dynamic programming or heuristics and examine how exactly they operate. One must assume, however, that given the global dominance of UPS, FedEx and DHL, that their solutions are highly efficient, if not optimal, and that their computer scientists continuously search for improvements to their algorithms and heuristics.

LITERATURE REVIEW

Currently there are numerous algorithms and heuristics to solve the "traveling salesman problem (TSP)" and the researchers have chosen to review a few in-depth as these seem to be the most dominant for this situation. Again other algorithms and heuristics may be more efficient and could be the subject of further research in the future. [14] focuses on highlighting a number of popular algorithms and heuristics published in the 1960s to late 2000s to solve the traveling salesman problem (TSP) by a variety of practitioners [1] [3] [5] [7] [8] [10] [11] [14] [15] [19] [20] [21].

One significant concept [14] mentioned for the symmetric traveling salesman problem (TSP) is that it is on a direct graph $G = (V , E)$. The V (vertex) represents a city and the E (edge) represents the distance traveled between the two vertex or cities The set $V = \{1,\dots, n\}$ is the vertex set, $E = \{(i, j): I, j \ \ V, i < j\}$ is an edge set [14] [22]. The most efficient algorithm in use today is Concorde and before this method, the simplex method was used to solve integer linear programs. This formulation assisted in solving a 42vertex instance containing cities in the United States [7]. [19] used the Accelerated Euclidean algorithm which is an extension of the method of integer forms published in 1963.

Laporte states in the mid-1970s the author, Miliotis was the first to devise a fully automated algorithm based on constraint relaxation and using either branch-and bound or Ralph Gomory's cuts to reach integrality [7] [14]. A cut-and price algorithm was published in 1979, combining subtour elimination constraints, Gomory cuts and column generation, but no branching [7] [11] [13]. Land's algorithm was able to solve nine Euclidean 100-vertex instances out of 10 [13]. In 1965, Jack Edmonds introduced the 2-matching inequalities, which were then generalized to comb inequalities [5]. Some generalizations of comb inequalities, such as clique tree inequalities and path inequalities turned out to be quite effective [3] [9].

Through the work of several authors from the 1980's to the early 90's polyhedral theory and of branchand-cut was developed [14]. This work allowed Martin Grotchel and William Holland to solve a drilling

problem of size n=2392 in 1991 [8]. The [14] article covers some of the significant author's and algorithms which have led to the development of Concorde by Applegate, which is the best solver for symmetric "traveling salesman problem" (TSP). Concorde is a computer program that is based on branch-and-cut – and price. Constraints and variables are initially relaxed and dynamically generated during the solution process. The algorithm uses 2-matching constraints, comb inequalities and certain path inequalities. It makes use of sophisticated separation algorithms to identify violated inequalities [14]. The largest instance now solved optimally by Concorde arises from a VLSI application and constraints 85,900 vertices [14].
Laporte mentions several significant heuristics for the symmetric "traveling salesman problem" (TSP). Constructive heuristics which continues to build until a solution is obtain. Improvement heuristics starts with a feasible solution and look for an improved solution that can be found by making a very small number of changes. The nearest neighbor heuristics starts at a random city and repeatedly visits the nearest city until all have been visited. Strip heuristic is quick and convenient for drilling problems that must be solved in real time [14]. The r-opt heuristic iteratively removes r edges from the current tour, considers all feasible reconnections and implements any improving move to yield the new current tour. Other heuristics mentioned are 2-opt, 3-opt, and OR-opt [14].
The [15] heuristic has been shown to be one of the best available improvement methods for the symmetric TSP. It applies set of 2-opt exchanges which, taken globally, produce a cost reduction. [15] found that in 2000, [11] had a successful implementation of the method. It was quick to produce optimal solutions for both small and large instances. The largest instance solved optimally with his heuristic contains 85,900 vertices [14]. For NP-hard problems, the Helsgaun heuristic is used to produce some optimal solutions. Gaps in these type of large instances are addressed by applying the shortest spanning 1-tree lower bound [10].
Patterson and Harmel (2003) illustrate an algorithm for using linear programming/zero-one programming and Solver in the solution of a seven -city traveling salesman problem [23]. What is holding people back is the lack of readily available and efficient solution algorithms has limited its practicality in many business applications. [23] points out significant Traveling Salesman Problem literature such as an early article titled "On the Hamiltonian Game (a traveling salesman problem)," by J.B. Robinson (1949) and "Solution of a Larger Scale Traveling-Salesman Problem" published in 1954 and considered to be one of the best known traveling salesman problem TSP papers. This article documented the use of a special application of linear programming for solving a 49-city problem [23]. According to Clause, "The standard formulation of the travelling salesman problem on n nodes as an integer program involves use of 2^n subtour elimination constraints. This paper provides a set of n^3 constraints that define the same polytope. This is accomplished through introduction of additional variables. An additional set of n(n-1)/2 constraints were introduced and results in a polytope that is smaller than the subtour elimination polytope. The introduction of further variables as well as constraints results in an even smaller polytope [2]. The program will continuously reduce the results as different factors are implemented such a cost and distance. The article stated the method involved too large a number of variables and constraints to be practical as a heuristic. It may, shed light on the differences between the polytopes defined by travelling salesman solutions and by subtour elimination constraints, in which these constraints yield an integer solution [2].
Another method is the multiple Traveling Salesman Problem (mTSP) in which the number of cities or edges are assigned to different salesmen. The concept extends the traveling salesman problem (TSP) equation as the problem implements multiple salesmen compared to the original with just one person. The method, according to the article; mTSP is defined on a graph G = (V, A), where V represents the set of vertices and A represents the set of edges. Let C = (C_{ij}) be the cost matrix defined on the set of A. If C_{ij}= C_{ji} then the cost matrix is symmetric, otherwise it is asymmetric [22]. The authors found that the trials formed single edges as heuristics were added into the formula, creating subtours of the cities. UPS created

a system called ORION, using a traveling Salesman Problem (TSP) algorithm to determine the shortest and most profitable route. Jack Levis, Senior Director of Process Management at UPS, stated that the system calculates thousands of combinations along with any time changes from customers to find the quickest delivery route. According to the article, the ORION program has been implemented on most of the company's travel routes and saved UPS more than $50 million a year [24]. ORION, along with the judgement of the driver, is able to deliver more packages faster while reducing time and cost to the company. According to [24], a Delivery Information Acquisition Device (DIAD) provides drivers with options to either use ORION or a combination of work rules, procedures and analytic tools that are used to establish the order of package deliveries. This system gives the driver the opportunity to use the most efficient system depending on their route, creating a harmony of human intellect and AI complexity.

METHODOLOGY

The first question to ask is "What specifically do we want to optimize?" That might seem like a trivial question but an optimal solution is a solution that works best for the user. For example, if the user wants to minimize distance as in the traditional "traveling salesman problem (TSP)", the user may spend additional time traveling. The user may spend additional unproductive hours on the road when a longer but less time-consuming route would have better met the need of the user. Will the user be less profitable spending unproductive hours on the road? That has to be another consideration and a potential weakness of the traditional "traveling salesman problem (TSP)" solution.

The researchers suggest the following approach:

- Determine which criteria or objectives are the most important and rank these
- Scenario analysis with algorithms or heuristics based upon the criteria or objectives
- Summarize the results of the various scenarios
- Compare the results of the various scenarios

If the results of the different scenarios are fairly close, choose objectives or criteria that produced the optimal results.

The problem is when scenarios based upon different objectives or criteria give divergent results. For example, the "least-distance" scenario gives a very different solution than the "least-time" scenario. This would imply that although the distance traveled is minimized, the user is spending many unproductive hours sitting in highway traffic. Is this an unacceptable compromise? This is a potential scenario that the user must plan for and apply managerial decision-making to determine the best course of action [12] [29].

The multi-criteria approach, however, comes with a severe handicap; because the "traveling salesman problem (TSP)" is NP-complete, additional criteria mean that the algorithm or heuristic must be run multiple times to determine the optimal route based on multiple criteria, which is a significant use of computer resources that will involve both time and money.

Is there a way to combine multiple objectives or criteria? Any combinations would be subjective and applicable to the user only. Some type of weighting scheme could be applied but again that would be ad hoc to the user only and if the weights are not precise; incorrect conclusions can be inferred. This is why the researchers propose to keep the objectives or criteria separate and run them separately based upon their individual merits only. If the user wants to combine multiple objectives or criteria, dynamic programming may be the best course of action [12].

The researchers plan to use the Solver function in MS Excel to perform the combinatorial optimization but other software such as AMPL, LINDO and MathCad could have been used also. The strength of the researchers' suggested course of action, is that it can be used with any commercial-off-the-shelf (COTS) optimization software and derive the same results.

The researchers will apply the Simplex Algorithm to perform the mathematical optimization. In standard/canonical form the Simplex Algorithm in matrix notation is:

$$\begin{array}{ll} \text{Maximize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b} \\ \text{and} & \mathbf{x} \geq \mathbf{0} \end{array} \tag{1}$$

A further constraint is that the decision variables must be binary (1, 0) since a fractional result is nonsensical. The user will either travel along the entire distance of the edges or not – no partial travel is allowed. The "traveling salesman problem (TSP)" will be formulated as an integer linear programming problem using an algebraic formulation in the standard/canonical form of:

$$\begin{array}{ll} \min \sum_{i=0}^{n} \sum_{j \neq i, j=0}^{n} c_{ij} x_{ij} & \\ 0 \leq x_{ij} \leq 1 & i, j = 0, \ldots n \\ x_{ij} \text{ integer} & i, j = 0, \ldots n \\ \sum_{i=0, i \neq j}^{n} x_{ij} = 1 & j = 0, \ldots, n \\ \sum_{j=0, j \neq i}^{n} x_{ij} = 1 & i = 1, \ldots, n \\ u_i - u_j + n x_{ij} \leq n - 1 & 1 \leq i \neq j \leq n. \end{array} \tag{2}$$

The researchers chose four landmark locations in the Los Angeles area to demonstrate their methodology. They are: (1) Sears Roebuck, Boyle Heights (the "Gateway to East LA"), (2) Dodgers Stadium, (3) the La Brea Tar Pits and (4) Giant Felix Chevrolet. There will be four nodes and six edges in the graph. This may seem like a small amount but it is large enough to illustrate the validity of the proposed methodology yet manageable to make valid comparisons.

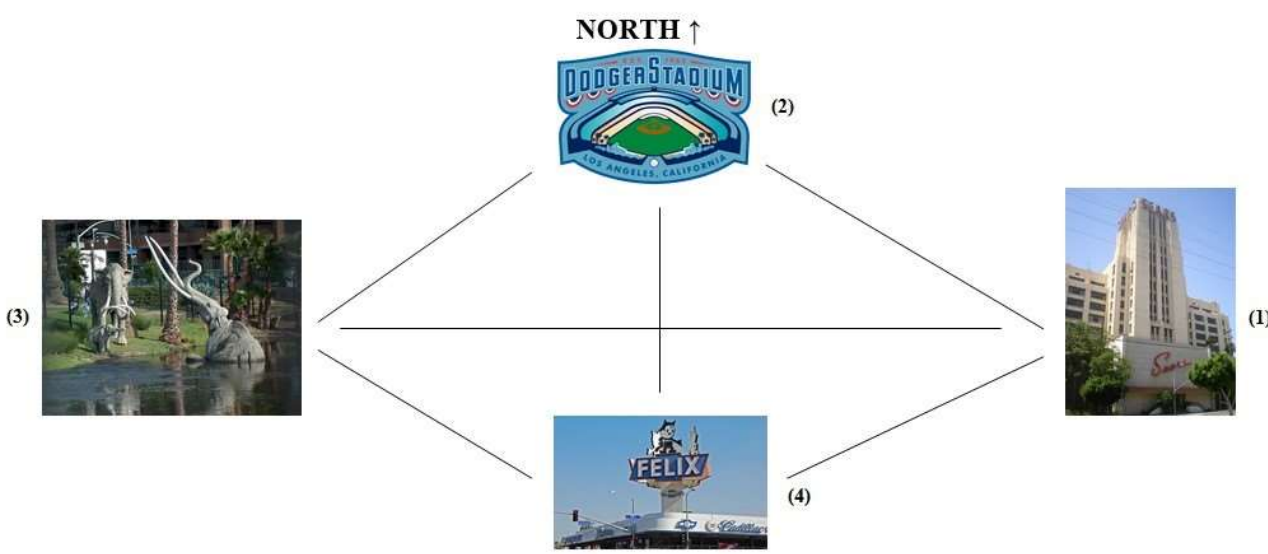

Figure 1. Graph of Locations (Edges)

The premise in this research assumes that the service provider begins the workday at any of the four locations and the purpose then is to determine the optimal path to the remaining three locations. The researchers will not introduce a new heuristic to determine an optimal path. The general methodology loosely follows the dynamic programming methodology but the setup is not strictly a dynamic programming formulation. The focus is on constructing a decision table based on multiple criteria to determine the optimal path.

The researchers propose the following “costs”

- distance, traditional in the “traveling salesman problem (TSP)”
- travel time (in minutes)
- expenses, tolls and “Fast Track” fees
- others

In addition, the researchers propose a profit maximization strategy. The service provider will accommodate all clients on a given trip but will “cherry pick” and concentrate on the most profitable job first, then the second most profitable, then the third most profitable, etc. This seems to be the strategy of cable TV service providers – they will do the profitable installations first, then the repairs at the end of the day since these jobs are purely expenses to the company.

The standard/canonical form of an integer linear programming formulation using matrix algebra to maximize profits is:

$$\begin{aligned} &\text{maximize} && \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ &\text{subject to} && A\mathbf{x} + \mathbf{s} = \mathbf{b}, \\ & && \mathbf{s} \geq \mathbf{0}, \\ & && \mathbf{x} \geq \mathbf{0}, \\ &\text{and} && \mathbf{x} \in \mathbb{Z}^n, \end{aligned} \tag{3}$$

While this is not a cost minimization problem, it is a common objective or criteria and can easily be incorporated into the researchers’ proposed methodology.

An example of expense minimization could be tolls or “Fast Track” traffic system charges which requires vehicles to have a transponder to access the system. The costs of these types of tolls could be broken-down into fixed and variable components. The fixed cost would be the entry fee to begin the system (including the cost of the transponder) and the variable costs would be the miles traveled on the “Fast Track” traffic system. The driver is trading one cost for another – the driver is willing to pay a toll to decrease travel time by accessing the toll road.

The expense minimization scenario will not be computed because relevant data is difficult to obtain. The “Fast Track” system in the greater Los Angeles area is a “time of use (TOU)” configuration and the “Fast Track” administrators do not make the “time of use (TOU)” rates readily available. Finally, there are no “Fast Track” or toll roads in the graph above.

Distances and travel times were retrieved from “Mapquest”, the on-line mapping service. The travel times were obtained at about 7:30 AM on a weekday morning, which is “rush hour” in the greater Los Angeles area. One of the researchers has found “Mapquest” to be fairly accurate in reporting the travel times and uses “Mapquest” each business day.

The tables assume symmetry. The distance between Location A and Location B is the same as the distance between Location B and Location A. This assumption weakens in regards to travel times, especially in terms of rush hours in the Los Angeles area but it won’t make much difference based upon the destinations in this study. Distances (in miles) between the locations are shown in the Table 1 below:

Table 1. Distance in Miles Between the Locations (Edges)

Los Angeles Locations	Sears Roebuck Boyle Height	La Brea Tar Pits	Giant Felix Chevrolet
Dodgers Stadium	4.9	8.2	5.4
Giant Felix Chevrolet	5.5	9.0	0
La Brea Tar Pits	11.5	0	0

Note. "Mapquest" was the source of these estimates.

Travel times (in minutes) between the locations are shown in the Table 2 below:

Table 2. Travel Times in Minutes Between the Locations (Edges)

Los Angeles Locations	Sears Roebuck Boyle Height	La Brea Tar Pits	Giant Felix Chevrolet
Dodgers Stadium	27	21	15 (seems to low)
Giant Felix Chevrolet	38	28	0
La Brea Tar Pits	28	0	0

Note. Travel times are estimated at about 7:30 AM, peak traffic rush hour in Los Angeles.
If there was a choice in routes, a conservative method was employed and the greatest travel time was chosen.
The researchers feel that the travel time in Table B between Giant Felix Chevrolet and Dodgers Stadium provided by "Mapquest" is inaccurate, especially during rush hour. The route between Giant Felix Chevrolet and Dodgers Stadium will take one through the infamous "LA Slot" in the Los Angeles freeway system. The "LA Slot" is the section of the 110 Freeway between the "Four Level Interchange" and the intersection of the Harbor Freeway and the Santa Monica Freeway. The "LA Slot" will take one past the heart of Downtown LA, where the Staples Center and the large office buildings are located. Every business day on the "LA Slot" is dreadful and it is even worse with rain or a traffic accident. While the discussion above about the "LA Slot" may seem excessive, it illustrates another difficulty in the "traveling salesman problem (TSP)" – inputs into the formulation must be accurate; otherwise, the final solution will be inaccurate and incorrect decisions will be made.
Table 3 were figures created by the researchers for the purposes of this study and shows the profit between the locations. Table 3 was constructed to facilitate a maximization problem and to incorporate these results into the final decision. The figures shown in Table 3 did not come from a credible source such as "Mapquest"; however, these are important inputs to illustrate the viability of the researchers' approach:

Table 3. Profits Between the Locations (Edges)

Los Angeles Locations	Sears Roebuck Boyle Height	La Brea Tar Pits	Giant Felix Chevrolet
Dodgers Stadium	$25	$300	$300
Giant Felix Chevrolet	$300	$25	0
La Brea Tar Pits	$25	0	0

Note. Profits are for illustration purposes only.

DECISION-MAKING BASED ON THE INPUT DATA AND ITS RESULTS

Three integer linear programming were run as part of the multi-criteria approach to the “traveling salesman problem (TSP)”. These scenarios were:

- distance minimization
- travel time minimization
- profit maximization

A rough approximation of the graph with its labels is shown below:

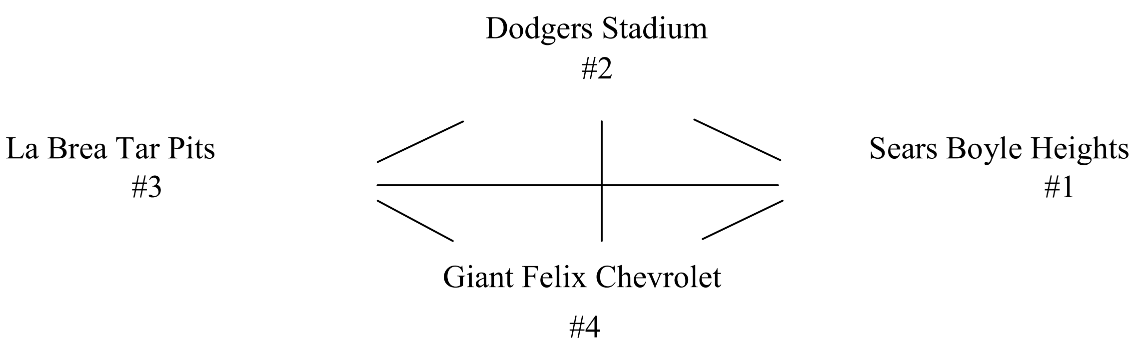

Figure 2. Graph of Edges with Labels

The numbers represent nodes while the lines indicate the edges. For example, X12 represents the edges between Sears Boyle Heights to Dodgers Stadium.

The formulation of the “distance minimization” scenario is shown below:

Minimize $4.9X12 + 11.5X13 + 5.5X14 + 8.2X23 + 5.4X24 + 9X34$

Subject to $X12 + X13 + X14 =$ 2 (to prevent cycling) [Sears constraint] $X23 + X24$
$+ X12 =$ 2 (to prevent cycling) [Dodgers constraint]

$X34 + X23 + X13 =$ 1 (to prevent cycling) [La Brea constraint]

$X14 + X24 + X34 =$ 1 (to prevent cycling) [Felix constraint]

$X12, X13, X14, X23, X24, X34$ binary (must be 0 or 1)

$X12, X13, X14, X23, X24, X34$ non-negativity

$X12 + X13 + X14 + X23 + X24 + X34 =$ 3 (“total # of visits” constraint)

Computer output is shown below:

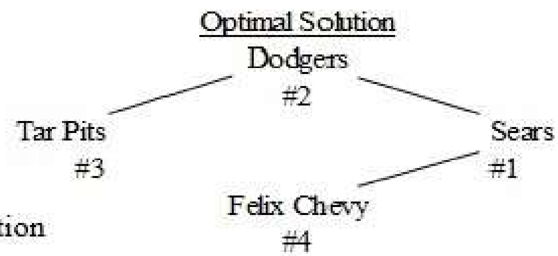

Figure 3. Distance Minimization

Linear Programming

	Label	C12	X12	C13	X13	C14	X14	Objective
Sears Roebuk	1	4.9	1	11.5	0	5.5	1	18.6
		C23	X23	C24	X24			
Dodgers Stadium	2	8.2	1	5.4	-2E-07			
		C34	X34					
La Brea Tar Pits	3	9	5E-07					
Giant Felix Chevrolet	4							

Constraint Table	
Sears Constraint	2
Dodgers Constraint	2
La Brea Constraint	1
Felix Constraint	1
X12	1
X13	0
X14	1
X23	1
X24	-2E-07
X34	5E-07
Total	3

Solutions Based on the "Greedy" Algorithm	
X12, X23, X14	18.6 Optimal
X12, X24, X34	19.3
X14, X24, X23	19.1
X14, X34, X23	22.7
X13, X12, X24	21.8
X13, X14, X24	22.4

This is an example of NP-Complete: Calculate the solution for every possible combination, then choose the minimum or maximum answer.

The formulation of the "travel time" minimization scenario is shown below:

Minimize 27X12 + 28X13 + 38X14 + 21X23 + 15X24 + 28X34

Subject to X12 + X13 + X14 = 2 (to prevent cycling) [Sears constraint] X23 + X24 + X12 = 2 (to prevent cycling) [Dodgers constraint]

X34 + X23 + X13 = 1 (to prevent cycling) [La Brea constraint]

X14 + X24 + X34 = 1 (to prevent cycling) [Felix constraint]

X12,X13,X14,X23,X24,X34 binary (must be 0 or 1)

X12,X13,X14,X23,X24,X34 non-negativity

X12 + X13 + X14 + X23 + X24 + X34 = 3 ("total # of visits" constraint)

Computer output is shown below:

There are two optimal solutions: one is shown as a solid line while the other is a dashed line.

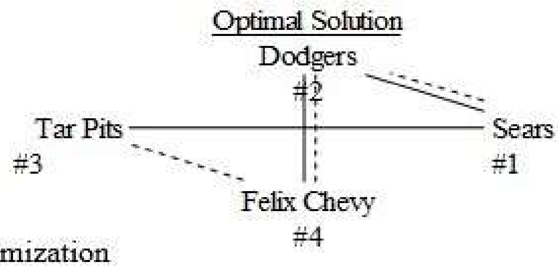

Figure 4. Travel Time Minimization

Linear Programming Formulation	Label	C12	X12	C13	X13	C14	X14	Objective
Sears Roebuk	1	27	1	28	1	38	0	70
		C23	X23	C24	X24			
Dodgers Stadium	2	21	0	15	1			
		C34	X34					
La Brea Tar Pits	3	28	0					
Giant Felix Chevrolet	4							

Constraint Table	
Sears Constraint	2
Dodgers Constraint	2
La Brea Constraint	1
Felix Constraint	1
X12	1
X13	1
X14	0
X23	0
X24	1
X34	0
Total	3

Solutions Based on the "Greedy" Algorithm	
X12, X23, X14	86
X12, X24, X34	70 Optimal
X14, X24, X23	74
X14, X34, X23	87
X13, X12, X24	70 Optimal

This is an example of NP-Complete: Calculate the solution for every possible combination, then choose the minimum or maximum answer.

The formulation of the "profit" <u>max</u>imization scenario is shown below:

Maximize $25X12 + 25X13 + 300X14 + 300X23 + 300X24 + 25X34$

Subject to $X12 + X13 + X14 =$ 1 (to prevent cycling) [Sears constraint] $X23 + X24$
$+ X12 =$ 2 (to prevent cycling) [Dodgers constraint]
$X34 + X23 + X13 =$ 1 (to prevent cycling) [La Brea constraint]
$X14 + X24 + X34 =$ 2 (to prevent cycling) [Felix constraint]
X12,X13,X14,X23,X24,X34 binary (must be 0 or 1)
X12,X13,X14,X23,X24,X34 non-negativity
$X12 + X13 + X14 + X23 + X24 + X34 =$ 3 ("total # of visits" constraint)

Computer output is shown below:

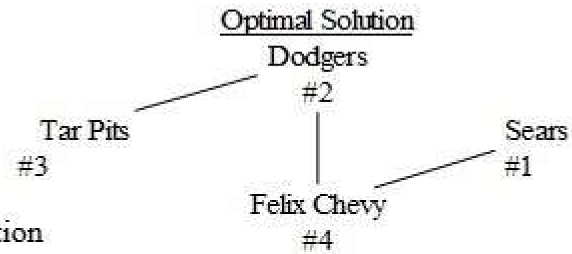

Figure 5. Profit Maximization

Linear Programming

	Label	C12	X12	C13	X13	C14	X14	Objective
Sears Roebuk	1	25	0	25	0	300	1	900
		C23	X23	C24	X24			
Dodgers Stadium	2	300	1	300	1			
		C34	X34					
La Brea Tar Pits	3	25	0					
Giant Felix Chevrolet	4							

Constraint Table

Sears Constraint	1
Dodgers Constraint	2
La Brea Constraint	1
Felix Constraint	2
X12	0
X13	0
X14	1
X23	1
X24	1
X34	0
Total	3

Solutions Based on the "Greedy" Algorithm

X12, X23, X14	625
X12, X24, X34	350
X14, X24, X23	900 Optimal
X14, X34, X23	625
X13, X12, X24	350
X13, X14, X24	625

This is an example of NP-Complete: Calculate the solution for every possible combination, then choose the minimum or maximum answer.

Again, the SOLVER function in MS Excel was used to run each scenario. SOLVER was specifically used because mathematical programming software such as AMPL or LINDO are powerful and user-friendly, but they may require a separate compiler and they are not inexpensive, to the detriment of smaller businesses. Most businesses already use MS Excel for other functions and therefore have ready access to SOLVER to deal with these types of problems.

The following decision table summarizes the optimal solutions from each of our scenarios. This decision table shows the impact of the optimal scenario on the other objectives or criteria. For example, the optimal distance minimization is shown, then the travel time and profit for this particular optimum is shown, so that a holistic comparison of all three scenarios simultaneously can be conducted.

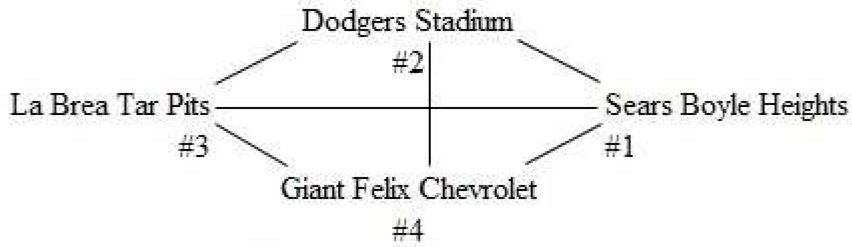

Figure 6. Summary of Decision Analysis
Note . Numbers are the vertices, lines are the edges

DECISION TABLE

Optimal	Network	Distance	Travel	Profit
Distance	X12, X23, X14	18.6	86	625
Travel Time	X13, X12, X24	21.8	70	350
Profit Maximized	X14, X24, X23	19.1	74	900

Let's examine the "profit maximized" optimum, which is network X14→X24→X23. This path gives us the greatest profit. If this path is chosen, the distance traveled will increase by ½ mile over the optimum while the extra time on the road will be four minutes over the optimum. Given that profits will increase by $275 at a minimum, it makes sense to travel a slightly longer distance and spend about 5 more minutes on the road to capture this additional profit.
Decision tables such as the "Summary of Decision Analysis" shown above indicate the value of (1) multi-criteria scenario analysis and (2) a presentation form which facilitates quick yet complete analysis. These decision tables are straightforward to construct and can be configured in a format that best fits the managerial decision structure of a given company.
Finally, additional criteria can be introduced, formulated, ran and analyzed to increase the precision of the decision-making of the "traveling salesman problem (TSP)".

CONCLUSION

The "Criteria-Based Approach to the Traveling Salesman Problem (TSP)" examines a common business problem and proposes an approach that introduces additional solutions to further improve managerial decision-making. Keep the objectives and criteria separate when performing the analysis so that the full effect of each objective or criteria can be measured on its own merit. Graphics and tables can enhance the analysis. While the proposed approach requires additional computer resources, the incremental benefits can outweigh the marginal costs.

REFERENCES

[1] Chvátal, V. (1973). Edmonds polytopes and weakly Hamiltonian graphs. Mathematical Programming, 5(1), 29-40.doi:10.1007/BF01580109
[2] Claus, A. (1984). A new formulation for the travelling salesman problem. SIAM Journal on Matrix Analysis and Applications, 5(1), 21-5. doi:http://dx.doi.org/10.1137/0605004
[3] Cornuéjols, G., Fonlupt, J., & Naddef, D. (1985). The traveling salesman problem on a graph and some

related integer polyhedra. Mathematical Programming, 33(1), 1-27. doi:10.1007/BF01582008
[4] DiNitto, T., and Enright, A. (2015, October 6). How are FedEx routes planned? [Discussion Board Posting]. Retrieved from https://www.quora.com/How-are-FedEx-routes-planned
[5] Edmonds, J. (1965). Maximum matching and a polyhedron with 0,1-vertices. Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics, 69b (1 And 2), 125-125.doi:10.6028/jres.069B.013
[6] Fleming, C., & Von Halle, B. (1989). Handbook of relational database design. Reading, Mass.: Addison-Wesley.
[7] Gomory R.E (1963). An algorithm for integer solutions to linear programs. In: Graves RL and Wolfe P (eds). Recent Advances in Mathematical Programming. McGraw-Hill: New York,pp 269–302 [8]
[8] Grötschel, M., & Holland, O. (1991). Solution of large-scale symmetric travelling salesman problems. Mathematical Programming, 51(1-3), 141-202. doi:10.1007/BF01586932
[9] Grötschel, M., & Pulleyblank, W. (1986). Clique tree inequalities and the symmetric travelling salesman problem. Mathematics of Operations Research, 11(4), 537-569. doi:10.1287/moor.11.4.537
[10] Held, M., & Karp, R. (1971). The traveling-salesman problem and minimum spanning trees: Part II. Mathematical Programming, 1(1), 6-25.doi:10.1007/BF01584070
[11] Helsgaun, K. (2000). An effective implementation of the Lin-Kernighan traveling salesman heuristic. European Journal of Operational Research, 126(1),106-130. doi:10.1016/S03772217(99)00284-2
[12] Hillier, F., & Lieberman, G. (2001). Introduction to operations research (7th ed., Mcgraw-hill series in industrial engineering and management science). Boston: McGraw-Hill.
[13] Land, A.H. (1979). The solution of some 100-city traveling salesman problems. Technical report, London School of Economics, London.
[14] Laporte, G. (2010). A concise guide to the traveling salesman problem. The Journal of the Operational Research Society, 61(1), 35-40. doi:http://dx.doi.org/10.1057/jors.2009.76
[15] Lin, S., & Kernighan, B. (1973). An effective heuristic algorithm for the traveling-salesman problem. Operations Research, 21(2), 498-516.
[16] Lee, G. M., & Kim, Y. E. (2016, January 12). Introduction to NP Completeness. Retrieved from https://www.slideshare.net/GeneMooLee/introduction-to-np-completeness
[17] Maier, D. (1983). The theory of relational databases (Computer software engineering series). Rockville, Md.: Computer Science Press.
[18] Mason, R. O., McKenney, J. L., Carlson, W., & Copeland, D. (1997). Absolutely, Positively Operations Research: The Federal Express Story. Interfaces, 27(2), 17–36. https://doi.org/10.1287/inte.27.2.17
[19] Martin G T (1966). Solving the traveling salesman problem by integer programming. Working Paper, CEIR, New York.
[20] Miliotis, P. (1976). Integer programming approaches to the travelling salesman problem. Mathematical Programming: A Publication of the Mathematical Optimization Society,10(1), 367-378. doi:10.1007/BF01580682
[21] Miliotis, P. (1978). Using cutting planes to solve the symmetric travelling salesman problem. Mathematical Programming, 15(1), 177-188. doi:10.1007/BF01609016
[22] Nuriyeva, F., & Kizilates, G. (2017). A New Heuristic Algorithm for Multiple Traveling Salesman Problem. TWMS Journal of Applied and Engineering Mathematics, 7(1), 101-109
[23] Patterson, M. C., & Harmel, B. (2003). An algorithm for using excel solver) for the traveling salesman problem. Journal of Education for Business, 78(6), 341-346.
[24] Rosenbush, S., & Stevens, L. (2015). At ups, the algorithm is the driver. Wall Street Journal - Eastern Edition, 265(38).

[25] What is NP-hard, NP-complete problem? [Discussion Board Posting] (2009). Retrieved from https://answers.yahoo.com/question/index?qid=1006051102995
[26] What software do drivers use to plan routes [Discussion Board Posting] (2016). Retrieved From https://www.reddit.com/r/UPS/comments/4kb8ue/what_software_do_drivers_use_to_pla n_routes/
[27] Wohlsen, M. (2013, June 13). The astronomical math behind UPS' new tool to deliver packages faster. Wired, Retrieved from https://www.wired.com/2013/06/ups-astronomical-math/
[28] Wolsey, L., & Nemhauser, G. (1988). Integer and combinatorial optimization (Wiley series in discrete mathematics and optimization). Hoboken: Wiley.
[29] Winston, W., & Albright, S. (2016). Practical management science (5th ed.). Australia: Cengage Learning.