AN HEURISTIC ALGORITHM TO SOLVE LESS-THAN-TRUCKLOAD VEHICLE ROUTING PROBLEMS

by

Andrea R. Castle

A thesis submitted to the Faculty and Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Mathematical and Computer Sciences).

Golden, Colorado

Date 3/29/95

Signed: Andrea R. Castle
Andrea R. Castle

Approved: Ruth A. Maurer
Dr. Ruth A. Maurer

Golden, Colorado

Date 3/29/95

Graeme Fairweather
Dr. Graeme Fairweather
Professor and Head
Department of Mathematical
and Computer Sciences

TABLE OF CONTENTS

ABSTRACT

Vehicle routing problems belong to a class of problems defined as NP-hard. These are of the most difficult problems in the class NP. Due to this classification, the optimal solutions involve highly computationally complex algorithms that are often uneconomical for industrial use. Since no polynomially optimizing algorithm exists to solve problems in NP, heuristic algorithms have been used to generate good to near optimal solutions in a reasonable amount of time for general industry applications.

This thesis presents an heuristic constructive algorithm to minimize the routing of less-than-truckload shipments from a regional satellite distribution center. The method is constrained by weight and mileage traveled per trip. Furthermore, the method also works to maximize the amount of full truckloads being routed by establishing a minimum weight at which to validate a full truckload. The algorithm was designed for efficiency, flexibility, ease of use, and the ability to generate a solution in a reasonable amount of time for all the regional satellite distribution centers.

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank the members of my thesis committee, Dr. Ruth Maurer, Dr. Robert E.D. Woolsey, Dr. John Golden and Dr. Art Bertanzetti for all of their time and assistance. I would also like to thank Coors Brewing Company's Corporate Transportation and Logistics Group for their support in this project, and Dr. Tom Wilger for the guidance he has lent in this project.

I would especially like to thank Dr. Art Bertanzetti for providing me support and direction as well as for providing me the opportunity to get involved with the Transportation and Logistics group at Coors Brewing Company.

INTRODUCTION

One could argue that, until recently, American corporations have accepted transportation and logistics methods “as is” with few questions asked about efficiencies although the transportation and logistics costs have been quite substantial. Only in the last few years has corporate management started asking questions and looking for ways to create efficiencies in this area.

One such area of transportation and logistics that can yield great improvement is vehicle routing. This area has been the subject of extensive study for the past few decades, see for example [1], [2], [4], [5], [6], [7].

A vehicle routing problem(VRP) is common for many types of industries and government agencies. The aim of most VRPs is to efficiently assign the delivery of goods to customers. These problems are typically constrained by capacity, mileage and/or fleet size.

Basically there are two types of algorithms to solve a vehicle routing problem: optimal and heuristic. Optimal algorithms yield solutions with an optimal or exact solution in a finite number of steps. These algorithms are usually based on a branch and bound approach[8]. The optimization algorithm for a VRP is defined as NP-hard[9]. Therefore, while a solution can be found, the computational time becomes uneconomic as the size of the problem becomes larger. Optimal algorithms are successful for problems with relatively few route locations. At present only relatively small instances of VRP have been solved optimally[1], [3].

The second approach to vehicle routing problems is heuristic algorithms which are capable of finding near-optimal and sometimes optimal answers in a reasonable amount of time. Heuristic algorithms can be categorized into four general types: (1) *Constructive algorithms*; (2) *Two-phase algorithms*; (3) *Incomplete optimization algorithms*; and (4) *Improvement methods.*[1]

Recently the Coors Brewing Company in Golden, Colorado has been employing operations research methodology to make improvements in their logistics and operations. One such topic for study is the routing of less-than-truckload shipments to distributors. Since Coors pays for a full truckload delivery regardless of whether or not the truck is full, they would like to minimize the number of trucks sent out without a full load.

A "drop-shipment" is a term for grouping less-than-truckload shipments into a full truckload shipment and making deliveries to more than one distributors in a single route. In other words, the truck "drops" each order/shipment at each stop along the route. Currently, operators of Coors' satellite distribution centers manually try to "marry up" the less-than-truckload (LTL) shipments with other LTL shipments to create a full or near-full truckload of orders. This is done on a weekly basis. This manual system does not have uniform guidelines and is currently not tracked or monitored. Coors would like to implement system wide parameters and methodology to minimize the routing of less-than-truckloads.

The algorithm to solve the VRP for less-than-truckload shipments will need to consider the following: (1) all LTL orders routed must originate from the same satellite distribution center; (2) LTL orders routed are limited to a maximum mileage per day; (3) LTL orders grouped together must be less than a maximum weight; and (4) certain distributors will be ineligible to be grouped in drop-shipments. The algorithm will also restrict delivery of shipments less than 75% full, measured by weight, except for those subject to special state laws.

This thesis presents a heuristic constructive algorithm that will be used to solve the vehicle routing problem to minimize the routing of less-than-truckload shipments for Coors Brewing Company at the Columbus satellite distribution center. This algorithm will be compared to other algorithms used to solve similar vehicle routing problems. Additionally, actual examples will be presented to prove its effectiveness.

OVERVIEW OF DISTRIBUTION

Coors utilizes a four-tier distribution system. This essentially means that Coors products visit four stops before they reach the consumer. Products are produced and packaged at one of the three breweries/packaging facilities. After the products have been packaged, they are shipped to a Coors Satellite Distribution Center (SDC). Currently, Coors has 21 SDCs. From the SDCs products are delivered to distributors. There are approximately 650 distributors in the United States. The distributors then deliver the products to retailers.

Coors Brewing Co. considers the distributors as their customers. The satellite distribution centers maintain inventories and are responsible for servicing the customers in its area.

Coors pays for the transportation from the brewery/packaging facility to the distributor. This cost is incorporated into the cost of the product. Thus, if Coors can reduce their cost of transportation, they will ultimately be able to recoup more profit.

Approximately 10% of Coors distributors sell about 50% of their sales volume. The other 90% sell the remainder. Coors is able to take advantage of large volume discounts in transportation costs to these high sales volume distributors. The reduction of transportation costs results in the form of direct shipments from the breweries (bypassing the SDCs) and from the use of shipping orders in full loads. Direct shipments are often shipped by rail or intermodal/piggyback[1].

Since the remainder of the distributors do not receive such large volumes of orders, their orders are supplied by the satellite distribution centers. These orders are usually shipped in full truckloads; however, some distributors receive less-than-truckloads. A less-than-truckload (LTL) is an order that does not constitute a full load. A full load is 45,000 pounds. Less-than-truckload orders are usually delivered via drop shipments.

The routing of drop shipments is a vehicle routing problem. By using a VRP algorithm one can improve on the transportation efficiencies of shipping LTLs.

[1] A truck-rail service. A highway trailer is loaded by a shipper and is driven to a rail terminal, where it is loaded on a rail flatcar; the trailer-on-flatcar combination is moved to the destination terminal by the railroad, where the trailer is offloaded and delivered to the consignee. [14]

PROJECT INTEGRATION

In the past few years Coors Brewing Company has devoted a considerable effort to the development of their "Computer Integrated Logistics" system (CIL). A critical task of CIL is the development of the Load Configurator/Order Picker. The Load Configurator and Order Picker will determine optimal vehicle loads based on current production schedules, current inventory and actual vehicle characteristics[13].

> "The primary goal of the project is to use computer and Operations Research technology to improve load configuration efficiency over the current system. Load efficiency is being measured by annual transportation cost and warehouse productivity."[13]

The Load Configurator's focus will be on orders combined into full loads. Distributors receiving drop shipments will not be evaluated. The question of how these drop shipments will be configured is separate because it requires different information from the standard Load Configurator problem[13].

Meanwhile, it is understood that the problem of drop shipments must be addressed. The intent of this thesis is to address this problem and provide an answer for Coors Brewing Company to implement within the Computer Integrated Logistics system. Figure 1 shows a flowchart of how the drop shipment project will fit into the Satellite Load Configurator/Order Picker.

SATELLITE LOAD CONFIGURATOR / ORDER PICKER

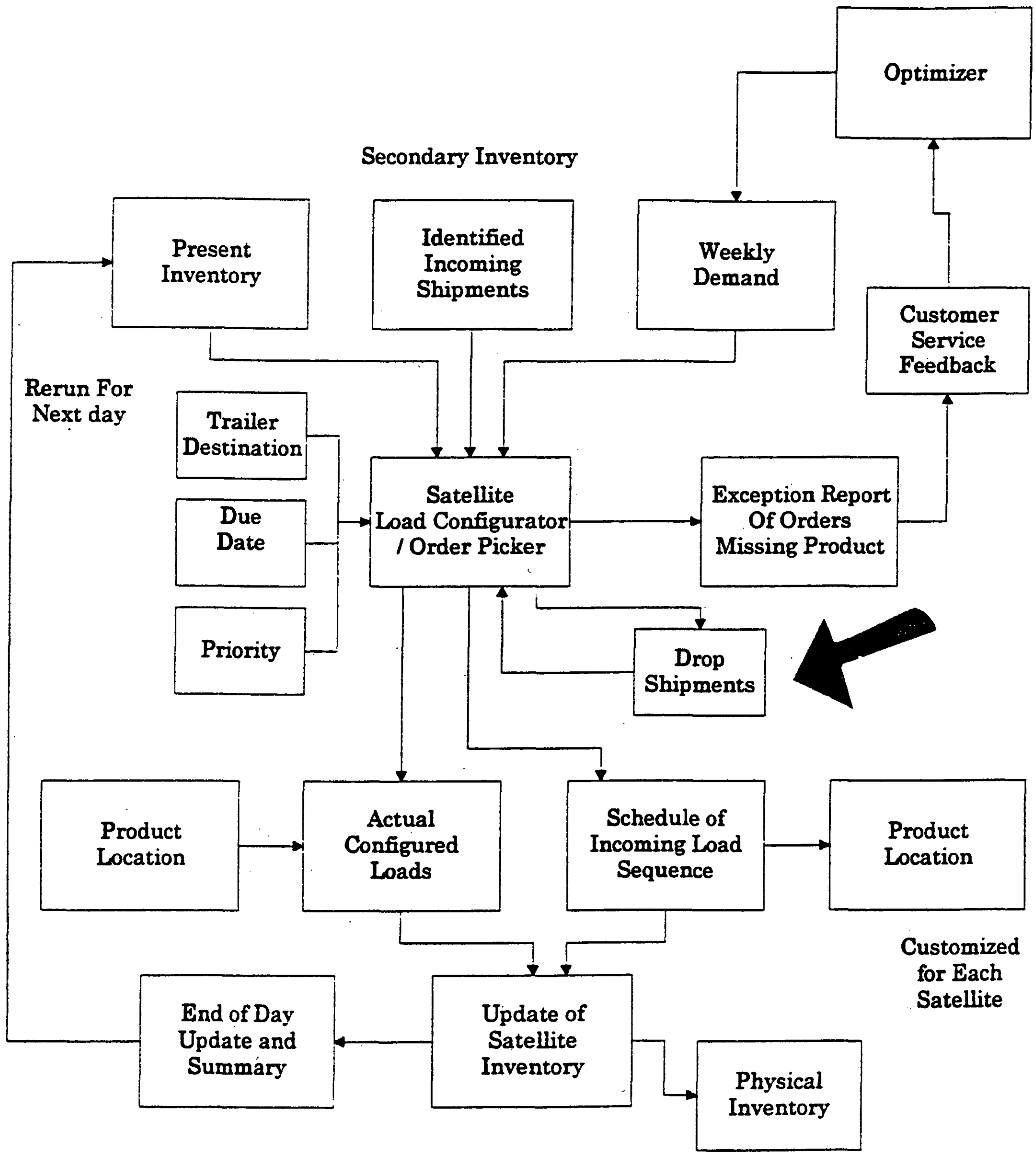

Figure 1 - Satellite Load Configurator/Order Picker Flowchart

VEHICLE ROUTING PROBLEM

The type of vehicle routing problem this thesis will explore deals with routing to multiple demand points with multiple vehicles from a single supply site for several supply sites. Additionally, it will explore combining less than full loads into a full 'vehicle-load'. The objective is to minimize the miles traveled on each route while maximizing the number of full truckloads (or vehicles) sent to the demand points.

A vehicle routing problem is classified as a combinatorial optimization problem meaning the problem may be subdivided into certain subproblems[3]. The vehicle routing problem involves an assignment problem in which loads are assigned to certain vehicles and a traveling salesman problem in which the vehicles are routed to minimize the distances.

Problems which have a known polynomial algorithm such as assignment problems are said to be in class 'P'[9]. Problems such as the traveling salesman problem (TSP) known for their inherent difficulty are in the class: 'NP' - non-deterministic polynomial[9].

In addition, the traveling salesman problem (TSP) belongs to a class of problems within all of NP which are known as the most difficult problems[9]. Therefore, if a polynomial algorithm were found to solve the TSP it would solve all NP problems. The reason this can be deduced is found in the concept of "transformability"[9].

The concept of transformability can best be explained as follows by an excerpt from Colin Reeves book, Modern Heuristic Techniques for Combinatorial Problems,

> Suppose you have a problem 'A' which can be solved by an algorithm 'A_1'. If one can transform every part of another problem 'B" into 'A' in polynomial time, then one can use algorithm 'A_1' to solve 'B'. The class of problems of 'A' is at least as large (and probably larger than) the class of transformed problems of 'B'. The class of 'B' are in a sense a special case of the problems of 'A'. Thus, it is reasonable to regard 'A' as being at least as hard (and probably harder than) 'B'. [9]

This concept is exhibited in Figure 2.

If a problem 'A' is such that every problem in NP is polynomially transformable to 'A', then 'A' is called NP-hard[9]. Additionally, if the problem 'A' belongs to the NP class, it is then said to be NP-complete[9]. Hence, these are the most difficult of all problems in NP. Since the vehicle routing problem can be transformed into a traveling salesman problem and the TSP is NP-hard, the VRP can be classified as NP-hard.

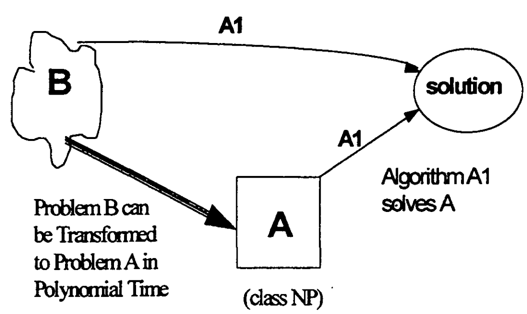

Figure 2 - Transformation in Polynomial Time

THE CASE FOR HEURISTICS

By definition a heuristic "is a technique which seeks good (or near-optimal) solutions at a reasonable computational cost without being able to guarantee either feasibility or optimality". [9] Depending on the heuristic the solution can fall within a range of feasible (an acceptable solution) to near-optimal (an extremely good solution). In contrast to a heuristic is complete enumeration in which every possible solution is evaluated to find the optimal solution.

Consider the traveling salesman problem classified as NP-hard which demonstrates the exponential growth in computing time as the size of the problem grows. The goal of the TSP is to minimize the distance traveled to N cities by a salesman. The salesman may start at any of the N cities but must visit each of the N cities one time and return to the original city. Since the starting city is arbitrary there are (N-1)! possible solutions (or (N-1)/2 if the distance between every pair of cities is the same regardless of the direction.) As the size of the problem grows larger, using the complete enumeration approach to find an optimal solution becomes uneconomical as it increases computing time exponentially to find a solution for (N-1)! routes.

The early emphasis of operations researchers was to find an exact solution for a problem. From this era several exact algorithms were developed to find solutions more efficiently than complete enumeration[9].

An example of such is the branch and bound method.[8] While these first algorithms were capable of solving some small problems, they were less capable of solving larger problems because they were computationally inefficient. Since then, researchers have been looking for a "polynomial optimizing algorithm" to solve problems such as the TSP.[9] As no such algorithm has yet been found, the case for an heuristic algorithm is strong.

INCOMPLETE OPTIMIZATION ALGORITHM METHOD

One heuristic approach for vehicle routing is called incomplete optimization or combinatorial optimization. Charles E. Noon, John Mittenthal and Rekha Pillai in "A TSSP + 1 Decomposition Strategy for the Vehicle Routing Problem"[3] introduce a new decomposition strategy for the VRP which separates the assignment of the vehicles to demand points from the determination of the individual routes. They solve a VRP with a combinatorial optimization problem using Lagrangian relaxation.

The authors' approach to the vehicle routing problem is a two step process. First, the vehicle dispatcher must assign "reward values" to each customer such that each customer will be visited by exactly one vehicle[3]. Next, based on the assigned reward values and the travel costs, the drivers have the responsibility to choose which customers to visit and then determine the route. This approach deviates from the standard vehicle routing problem in that the driver (or salesman) is not required to visit every customer and the driver gets to choose which customers to visit. This is called a "TSSP + 1" problem because the problem has been subdivided into two problems[3].

In order to maintain the Noon, et. al. approach, a 'TSSP + 1' problem must be solved for each vehicle. In the special case that the authors examined all the 'TSSP + 1' problems were identical for each vehicle. In other words, they only needed to solve one subproblem in order to solve the whole problem. For this case of identical vehicles the linear programming relaxation of a set partitioning representation of the problem was capable of providing a feasible solution[3].

However, this differs from the Coors problem because each subproblem is not identical. This is a highly complex version of the problem addressed by Noon, et.al. Since the authors use a set partitioning approach with linear programming relaxation, as the number of customers increases, the problem representation becomes infeasible due to size to solve even for medium-sized problems.[3] Noon, et.al., conclude with the comment that "these issues (the larger problems) will be taken up in future work."[3] Therefore, this approach is not an alternative for the Coors problem.

IMPROVEMENT METHODS

Improvement methods are search schemes that examine successive 'neighbors' of a solution. The objective function is allowed to deteriorate in order to avoid a local minimum for the solution[1]. "As a rule, these algorithms are designed to be open ended and their running time, which can sometimes be quite large, is not a polynomial function of the size of input."[1]

Two such improvement methods are the metaheuristic methods of simulated annealing and tabu search. The basis to both of these heuristic methods is the neighborhood search. A neighborhood search is the process of locating a better solution within the current solution's neighborhood. A neighborhood of a current solution is the set of new solutions that can be reached by making a simple operation. A simple operation is the removal of an object from a solution or possibly an interchange between two objects in a solution.

Tabu search was proposed by Glover [11] and uses the neighborhood search technique while maintaining a history of the instances encountered during the search process. Certain instances are considered 'tabu' for a time while the search process examines other areas of the solution space[10]. This is done to overcome possible local minima in the solution set.

Tabu search is based on the following: (1) the use of flexible memory structures designed to allow for evaluation criteria and historical search information; (2) an associated mechanism of control; and (3) the incorporation of memory functions of different time spans to implement strategies for intensifying and diversifying the search[10]. While tabu search can be used and has been used to solve vehicle routing problems, a recent article by Gendreau, Hertz and Laporte describes a specialized method of tabu search developed specifically for the VRP. M. Gendreau, A. Hertz and G. Laporte in "A Tabu Search Heuristic for the Vehicle Routing Problem" (*Management Science, 1994)* describe a new version of a tabu search heuristic to solve VRPs called TABUROUTE. The algorithm works to develop a set of least cost vehicle routes considering that (1) every stop must be visited at least once; (2) every stop is associated with a non-negative demand; (3) every stop includes a service time that is included as part of the actual route time; (4) every route starts and ends at the depot; and (5) other side constraints - not defined in the paper[1].

This paper's approach is to repeatedly use a generalized insertion procedure to generate neighbor solutions until the best solution is found. This approach does solve the problem; however, the computational time required is generally longer than classical heuristics and may prove to be unsuitable[1].

Additionally, set-up time to work the problem through tabu search is significant. The tabu search algorithm must be set up individually for each problem. Each problem must have a defined proper neighborhood and target analysis, as well as tabu and aspiration candidates[10]. If a problem is subject to frequent redefinition, tabu search can be a timely means to solve it. Since Coors' market and distribution strategies are likely subjects for change, it is likely that this method may not be appropriate.

Simulated annealing is another metaheuristic classified as an improvement method. The approach for simulated annealing is to perform a random search of a solution's neighborhood for a better solution. Simulated annealing continuously tries to improve the solution. However, as in tabu search methodology, some inferior solutions are accepted in order to avoid local minima. The main difference between simulated annealing and tabu search is that simulated annealing has no memory.

The idea of simulated annealing was first published in 1953 as an algorithm to simulate the cooling of material in a heat bath[9]. "The algorithm simulated the change in energy of the system when subjected to a cooling process, until it converged into a steady state.[9]" Thirty years later, its use was explored to find feasible solutions for optimization problems[9]. Today, simulated annealing is widely used as an optimization technique[12].

In Lester Ingber's article "Simulated Annealing: Practice versus Theory" he discusses some of the negative features of simulated annealing. The primary criticism is that it is too slow. "Another criticism is that it is 'overkill' for many of the problems in which it is used."[12] The author also mentions that it is difficult to fine tune specific problems relative to other techniques. Finally, the author states that he believes simulated annealing suffers from "over-hype" that has lead to misinterpretation of results and misuse by transformation of problems into other suboptimal techniques.

With the above comments in mind, this approach does not seem appropriate for finding an effective solution to the Coors vehicle routing problem.

TWO-PHASE ALGORITHM METHOD

An example of a two-phase algorithm is described by B. Gillett and L. R. Miller in "A Heuristic Algorithm for the Vehicle-Dispatch Problem" (*Operations Research, 1974*). They discuss a method called the sweep algorithm to solve medium to large vehicle-dispatch problems. The algorithm aims to minimize total distance traveled in supplying all demand subject to load and distance constraints.

Many vehicle routing algorithms attempt to solve the vehicle dispatch as a one big problem. The sweep algorithm, however, divides the locations into a number of routes and then operates on individual routes by performing a forward sweep algorithm and a backward sweep algorithm until an optimum or near-optimum solution is discovered[4]. While this method is successful in solving the problem, it is inefficient as the computation time increases quadratically with the average number of locations per route if the total number of locations remains relatively constant[4].

This would be an ineffective method for the Coors vehicle routing problem because the total number of locations (or supply points) will remain relatively constant and the average number of locations per route will change each week.

CONSTRUCTIVE ALGORITHM METHODS

In G. Clarke and J. W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points" *(Operations Research, 1964)*, their approach is concerned with the optimum routing of a fleet of vehicles of varying capacities from a central depot to a large number of delivery points[2]. Clarke and Wright's method aims to allocate loads to trucks such that all merchandise is assigned and total mileage is minimized.

This differs slightly from the problem considered here in that Coors does not require that all merchandise (or orders of products) is assigned unless state law prevails that all orders must be delivered in the week they are issued. Additionally, while minimizing total mileage is a priority, sending full truckloads is also a goal. Another slight difference is that each capacity will not vary from each "central depot".

Consider an example of how this method is applied. Suppose a supplier has orders for six different customers to deliver in a week. The supplier has two vehicles to make these deliveries. Each vehicle has the capacity to carry 15 units. Each customer must receive their order in the week.

Of course, the supplier could send the vehicles out individually to each customer and return before heading onto the next. However, the supplier would like to minimize the distance traveled during the week without sacrificing deliveries to any customer. Figure 2 shows a diagram of this example of a single supply point network.

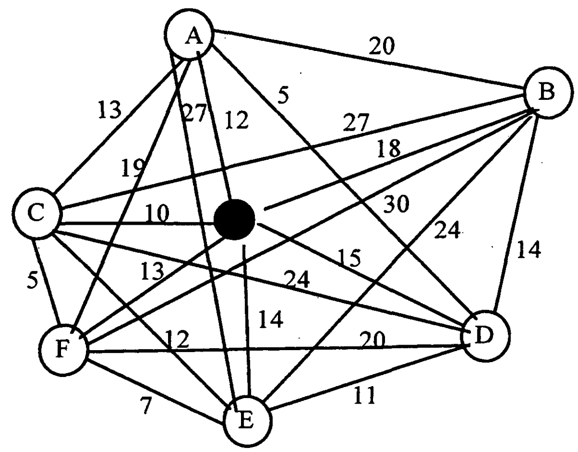

Figure 3 - Clarke & Wright Single Supply Point Network

The black filled node represents the supply point (or the central depot). The six hollow nodes labeled A through E represent the demand points (or customers). The central depot is connected to each customer by a line which indicates the distance to each point. Likewise, each hollow node (or customer) is also connected to the other hollow nodes (or customers) by a line indicating the distance between each pair.

Furthermore, each customer will receive a certain shipment accounted for in units for their weekly demand. Table 1 below lists the demand (in units) for each customer.

	A	B	C	D	E	F
Demand	2	3	10	4	7	4

Table 1 - Demand for Single Supply Site Network Example

After identifying the demand for each customer, the vehicle availability, vehicle capacity constraints, and the distances between each site, the next step is to build a matrix from the distances between the nodes.

The size of the matrix is directly related to the number of supply and demand points. For a single supply point problem, the matrix will be (N + 1) x (N + 1) where 'N' is equal to the number of demand points. Thus, the example will set up a 7x7 matrix. The distance from node 'A' to node "B" will be the AB^{th} entry in the matrix, and so forth for each node respectively.

One advantage of this matrix setup is that it allows for unequal travel between points. In other words, it allows for the travel distance (or time) to be different traveling from point A to point B than traveling from point B to point A. If the travel times are symmetric, only half of the matrix need be considered.

The distance matrix created for the example is shown in Table 2. Notice that since the distances are symmetric, only one-half of the table is needed. The simplified distance matrix for the example is shown in Table 3.

	Depot	A	B	C	D	E	F
Depot	0	12	18	10	15	14	13
A	12	0	20	13	22	27	19
B	18	20	0	27	14	24	30
C	10	13	27	0	24	12	5
D	15	22	14	24	0	11	20
E	14	27	24	12	11	0	8
F	13	19	30	5	20	7	0

Table 2 - Distance Matrix for Single Supply Site Example (showing symmetry)

	Depot	A	B	C	D	E	F
Depot	0						
A	12	0					
B	18	20	0				
C	10	13	27	0			
D	15	22	14	24	0		
E	14	27	24	12	11	0	
F	13	19	30	5	20	7	0

Table 3 - Distance Matrix for Single Supply
Point Example (simplified matrix)

Given the distance matrix, the next step is to compute the distance saved matrix. The distance saved matrix is computed by calculating the distance for separate round trips to two demand points from the supply point, then comparing it to the distance from the supply point to the demand point then to the other demand point before returning to the supply point.

For example, to calculate the FEth entry in the distance saved matrix first calculate the distance to travel roundtrip to visit both customer E and F. Since the distance from the supplier (S) to customer E is 14 and to customer F is 13, a round-trip distance to visit each is 54. The distance between customer E and F is 7. Therefore, if the supplier (S) were to travel to customer E then to customer F and finally return to S, the total distance traveled would be 34. The distance saved by servicing the two nodes in one trip is 20. This number is what is stored in the FEth entry of the distance saved matrix. This example is illustrated in Figure 4.

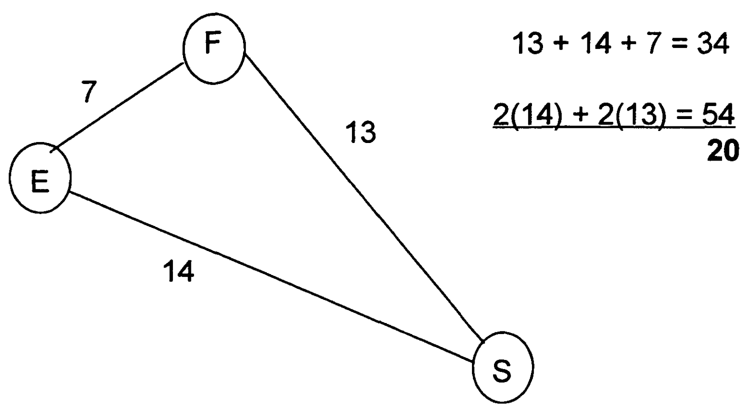

Figure 4 - Distance Saved Exhibit

The calculation for each entry in the distance saved matrix is equal to {2*[distance(S to E) + distance (S to F)] - [distance (S to E) + distance (S to F) + distance

(E to F)]} for each demand point respectively. Table 4 shows the distance saved matrix calculated for the example.

	A	B	C	D	E	F
A	0					
B	10	0				
C	9	1	0			
D	5	19	1	0		
E	-1	8	12	18	0	
F	6	1	18	8	20	0

Table 4 - Distance Saved Matrix
for Single Supply Site Example

From the distance saved matrix one can begin to select routes for customers, starting with the first pair of customers who has the largest distance saved. This value is 20 in the example representing the pairing of customers E and F. After the largest value is selected, the constraints must be checked.

The combined demand of units for customers E and F are within one vehicle's capacity. Customers E and F's demand totals 11 units and each vehicle has the capacity for 15 units. However, because the vehicle is not full, it must be determined whether other customers could be served on this route.

The next step is to look in the rows and columns of E and F for the next largest distance saved. This value is 18 and it occurs for both customer C and D. Either is acceptable.

If we were to choose C, we must evaluate it against the remaining capacity in the vehicle. The remaining capacity in the vehicle is 4 units (15 - 11). Since the order from customer C is for 10 units and there is not enough room in the vehicle for all of customer C's order, customer C must be rejected.

At this point we would evaluate customer D. Customer D's order is for 4 and the vehicle is now at capacity so it can be routed for delivery.

The order for the route is determined by the way the distance savings are chosen. Customer D was chosen because of the distance saved combined with customer E. Therefore, this vehicle will be routed S-F-E-D-S (where S stands for the supply point.)

This algorithm repeats itself until all remaining customers' deliveries have been satisfied. Table 5 shows the results of the example.

Vehicle No.	Route	Total Distance	Total Load
1	S-F-E-D-S	46	15 units
2	S-B-A-C-S	61	15 units

Table 5 - Results of Clarke & Wright Algorithm for Single Supply Site Network Example

This method does not usually yield optimal answers; however, it provides good solutions that are usually better than the random methods. It is also the quickest method to employ. It will provide a good solution to Coors vehicle routing problem given some alterations and tailoring of methodology to make it specific to their problem.

SITE APPLICATION

A vehicle routing algorithm has been developed for Coors Brewing Company's Columbus satellite distribution center as a pilot for the rest of their satellite distribution center network. The Columbus satellite services distributors in Ohio, Kentucky and West Virginia.

All of the distributors in the Columbus network receive less-than-truckload shipments at some point during the year. Some distributors never receive full loads during the year. Thus, there is a great need for a systematic approach to route these loads. A map of the Columbus Satellite Distribution Center network is contained in Appendix B as well as a distance matrix created by MileMaker© [21].

While there is quite a bit of research that has been accomplished in the area of vehicle routing problems, there are only a handful of software packages available in the market. Research of several software houses to investigate their available packages revealed that no canned software package existed would fit Coors unique situation without a significant number of customizations. For software company references see [15], [16], [17], [18], [19], [20]. Hence, there existed an opportunity and need to build a custom designed system for Coors to use.

The algorithm used to solved the problem has been modeled from Clarke and Wright's vehicle routing algorithm. It has been modified to fit specifications to serve the less-than-truckload delivery for Coors Brewing Company's Columbus satellite distribution center.

For application of the Clarke and Wright algorithm model criteria was defined. The start of the process will originate with a weekly list of the customers and their individual demand generated by CIL. The distances or mileage to be considered will be generated by a software package called MileMaker © [21]. The assumption will be made that the number of vehicles available is infinite.

In building the algorithm, there were some additional factors to consider. One consideration was that no combinations of LTLs could weigh more than 45,000 pounds. The second consideration concerned the maximum mileage that could be traveled in one day. The standard limit for travel is 500 miles per day. However, the round-trip distance to some distributors is greater than 500 miles. This meant that a variable mileage per day maximum had to be established such that it could accommodate the distributors at distances further than 250 miles from the satellite.

This became a tricky issue as the obvious solution to set a desirable maximum was not to simply double, triple or quadruple the mileage maximum in order to allow for drop shipment deliveries. For instance, in the case of the distributor in Henderson, Kentucky the distance to Henderson from Columbus is 333 miles, therefore, the round-trip distance is 666 miles. Since the round-trip distance is over 500 miles, the truck scheduled to deliver the order would plan on an overnight (or two day) trip. One would assume that the mileage maximum set would be 1000 miles.

Unfortunately, this did not work because Coors can be charged for out-of-route mileage. While there are no set standards for out-of-route mileage charges, these charges occur when trucks deliver drop shipments that are not within the same general vicinity. An extreme example of a case for out-of-route charges would be a drop shipment that is routed from Columbus, Ohio to a distributor in Cleveland, Ohio and then to one in Louisville, Kentucky.

Hence, to allow for drop shipments to distributors at further distances, a formula was developed to accommodate them. The mileage maximum formula is determined with the initial combination chosen from the distance saved matrix and is used until the truck is routed. The mileage maximum is calculated as follows:

Let: (A,B) represent two distributors who are initial candidates for pairing drop shipments.

Let: (*) indicate the distributor whose distance is the furthest from the satellite (S).

Mileage Maximum♠ = Greater of [30% of [(S-A*) + (S-B)] + 2*(S-A*)]

or 500 miles

♠ - *rounded to the nearest whole number*

An example of the formula calculation is demonstrated below. Let A represent a distributor that is 333 miles from the SDC and B represent another distributor that is 315 miles from the SDC. Assume the distance between A and B is 30 miles. This example is exhibited in Figure 5.

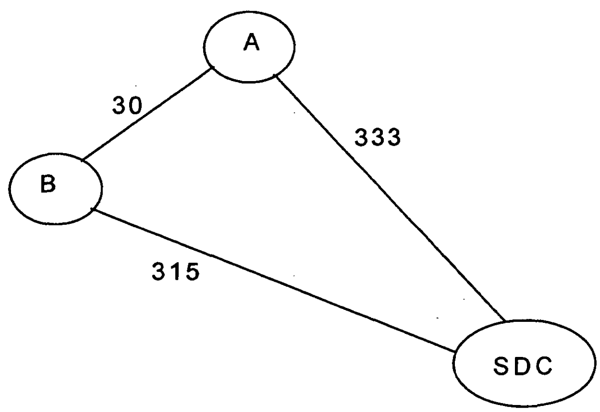

Figure 5 - Mileage Maximum Exhibit

Since distributor A is the furthest from the SDC, the formula is applied as follows: Mileage Maximum equals the greater of [.3*(333 + 315) + 2*333 = 860.4 (which rounds to 860)] or 500 miles. Therefore, the mileage maximum equals 860 miles.

This formula was designed to keep the drop shipment route within a certain radius of the distributor that is the furthest distance from the SDC. This formula has been tested against actual weeks with LTL orders and has proven sufficient.

A third consideration for the problem was to set a minimum weight from which to reject loads from routing. Since Coors pays for a full load whether or not a full load is being delivered, they want to be able to reject loads that are below a minimum weight. The minimum weight will be set at 75% full or 33,750 pounds. The only exception to the minimum weight rejection level is for states with FOB status. States with FOB laws require that goods ordered in a specific week must be delivered that week. In other words, orders cannot be carried over into another week. Ohio, Kentucky and West Virginia are all FOB states. Therefore, the weight minimum requirement will not be observed in the algorithm for the Columbus SDC.

The fourth consideration is for the recognition of less-than-truckloads orders by the same distributor. Since there can be multiple LTL orders for the same distributor in one week, a check will be added to route them together if possible. The orders to the same distributor will be considered before the distance matrix is built. If the combined orders to the same distributor add up to a total weight that is at least 90% full (or 40,500 pounds), then the orders will be combined and routed and not included when building the distance matrix. However, if the loads combine to greater than a full load or less than 90% full, then they will be put in consideration with the remaining sites and built into the distance matrix.

The final consideration is for pricing of the routes. The route price is determined by the rate to the distributor that is the furthest distance from the SDC. Each additional stop is charged a standard sum. For the analysis we will assume that sum will be assumed to be $50. Actual rates for the routes will be masked because they are considered confidential information.

Figure 6 exhibits a flowchart of the Less-Than-Truckload Vehicle Routing Algorithm .

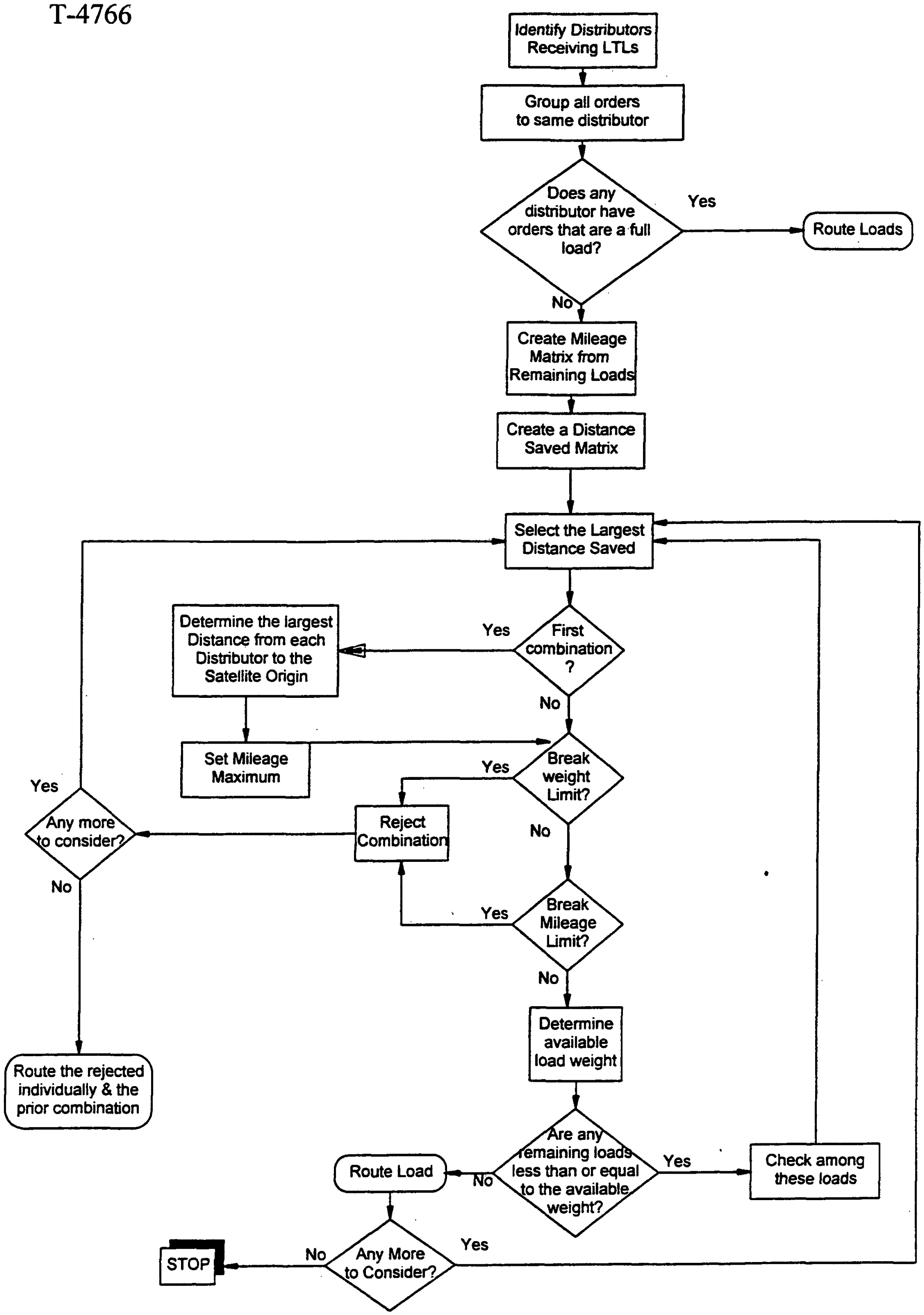

Figure 6 - Flowchart of the Less-than-Truckload Vehicle Routing Algorithm

TESTING OF THE ALGORITHM

To find out how effective this algorithm was, its performance was tested against 10 previous actual weeks' load marriages at the Columbus Satellite Distribution Center. The following is a demonstration of the algorithm with an actual week's drop shipments.

Following the flowchart, the first step is to identify the distributors receiving LTLs during the week. These distributors are listed in Table 6.

Distributor	Weight
Parkersburg, WV	22,732
Bellaire, OH	16,112
New Martinsville, WV	1,224
Morgantown, WV	11,032
Kingwood, WV	23,083
Owensboro, KY	33,732
Henderson, KY	9,022
Perrysburg, OH	18,083
Cincinnati, OH	16,414
Clarksburg, WV	21,990

Table 6 - Customer Demand - Example 1

Since each distributor is receiving only one order for the week, the next step is to calculate the distance matrix. Figure 7 shows a map of the distribution network for the example.

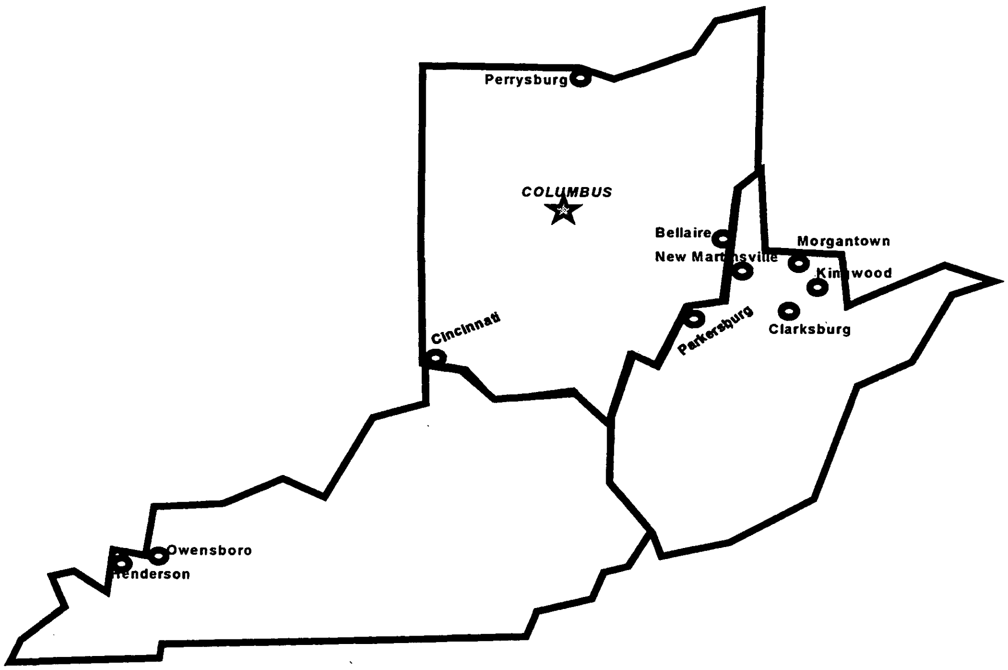

Figure 7 - Map of Distribution Network for Example 1

Table 7 shows the distance matrix calculated using MileMaker©.

	Owen	Hend	Cinc	Bell	Perr	Clar	King	Morg	NMar	Park
Columbus	315	333	110	129	122	181	224	201	134	111
Owensboro	0	30	205	441	388	463	514	502	443	393
Henderson	30	0	223	459	396	481	532	520	461	411
Cincinnati	205	223	0	236	194	261	312	300	241	191
Bellaire	441	459	236	0	204	118	103	80	33	91
Perrysburg	388	396	194	204	0	295	299	276	234	231
Clarksburg	463	481	261	118	295	0	51	39	54	70
Kingwood	514	532	312	103	299	51	0	23	86	121
Morgantown	502	520	300	80	276	39	23	0	63	109
New Martinsville	443	461	241	33	234	54	86	63	0	50
Parkersburg	393	411	191	91	231	70	121	109	50	0

Table 7 - Distance Matrix for Example 1

Next, the distance saved matrix is calculated from the distance matrix in Table 7 using the formula from the Clarke and Wright algorithm:

The ABth distance saved = {2*[distance(S-A) + distance(S-B)] - [distance(S-A) + distance(S-B) + distance(A-B)]}

Repeating this formula for each customer the distance saved matrix is completed. The completed distance saved matrix is shown in Table 8. Since the distances are symmetric, only one-half of the matrix is needed.

	Owen	Hend	Cinc	Bell	Perr	Clar	King	Morg	NMar	Park
Owensboro	0									
Henderson	618	0								
Cincinnati	220	220	0							
Bellaire	3	3	3	0						
Perrysburg	49	59	38	47	0					
Clarksburg	33	33	30	192	8	0				
Kingwood	25	25	22	250	47	354	0			
Morgantown	14	14	11	250	47	343	402	0		
New Martinsville	6	6	3	230	22	261	272	272	0	
Parkersburg	33	33	30	149	2	222	214	203	195	0

Table 8 - Distance Saved Matrix
for Example 1

The next step is to select the largest distance saved from Table 8. The largest value found in Table 8 is 618 which combines Owensboro, KY with Henderson, KY. Since this is the first combination, the mileage maximum must be determined. Henderson, KY is the furthest distance from Columbus at 333 miles. Owensboro is 315 miles from Columbus. Using the mileage maximum formula, we calculate the mileage maximum is calculated to be 860 miles.

Continuing with the steps of the flowchart, the combined weight of both loads is 42,754 pounds. This does not break the weight limit of 45,000 pounds. The total mileage is 678 miles. This does not break the mileage maximum. The remaining load weight available is 2,246 pounds. There is one remaining load that is less than this weight. It is an order for New Martinsville. However, after checking the total mileage for the three stops, we find that the additional stop will break the mileage maximum. Therefore, we must reject the additional load and save it for further consideration. Since there are no other loads to consider, the original combination should be routed.

Repeating this process until all possible combinations have been paired yields the results listed in Table 9.

Route (including SDC)	Total Load	% Full
Owensboro - Henderson	42,754	95%
Kingwood - Morgantown - New Martinsville	35,339	79%
Parkersburg - Clarksburg	44,722	99%
Perrysburg - Bellaire	34,195	76%
Cincinnati	16,414	36%

Table 9 - Scheduled Routes for Example 1

The resulting total cost of the drop shipment routes scheduled for this particular week is $2,353 for five routes. The total cost from the actual schedule of drop shipments was $2,589 for six routes. The savings realized by using the new method is $236.

However, the actual savings that would have been realized by Coors is much greater due to discrepancies in billing for drop shipments. Presently, Coors relies on a manual billing process for drop shipments because there is no system in place to monitor the actual occurrences of drop shipments. Due to this situation, the actual savings realized are very difficult to estimate. Thus, the estimated savings resulting from using the algorithm and exact billing falls within a range of $236 to $1,853. This range is estimated from calculating the cost for the low range of savings by the charge for the furthest distance and additional drop shipment charges and for the high range by charging for each individual stop.

The preceding example is just one showing the savings for a single week, for a further basis of comparison nine other actual examples have been worked and are presented in Appendix A. The examples presented in Appendix A represent nine separate weeks of the year at the Columbus satellite. They vary in the number of less-than-truckload shipments from four to eighteen destinations. In all but one case the algorithm provided a less expensive solution.

This instance that created a higher charge occurred in example 10. The reason the example's actual cost was less than the cost computed by the algorithm was because constraints were broken. All loads routed in example 10 were below the minimum truckload weight and opportunities to maximize the truckload were not taken.

The average savings for the ten examples ranged from \$134 to \$1,320 per week which translates to approximately \$70,000 to \$675,000 per year. Table 10 represents the cumulative results from the examples in Appendix A.

	Low	High
Example	Savings	Savings
1	$236	$1,853
2	$304	$2,018
3	$0	$484
4	$328	$1,645
5	$283	$1,351
6	$0	$1,276
7	$50	$1,394
8	$15	$653
9	$143	$1,242
10	($24)	$1,285
Total	$1,335	$13,201

Table 10 - Cumulative Results from Actual Examples

CONCLUSIONS

The preceding algorithm provides an effective and efficient process for routing less-than-truckload shipments at Coors Brewing Company's Columbus satellite distribution center. This process could also be applied to the other satellite distribution centers.

The method chosen is straightforward and understandable which would make it easy to implement and use for management as well as for the distribution center personnel. It is also capable of providing a good solution in a reasonable amount of time. Additionally, the algorithm was developed to be a generalized method for the entire satellite distribution center network. The algorithm was designed with flexibility. Adjustable parameters have been built in to accommodate: maximum mileage traveled per day, minimum weight requirements at each individual satellite, and the opportunity to manipulate input and output data.

Furthermore, since the minimum weight constraint would restrict the algorithm from routing a load that it didn't consider to be a "full" load, the financial impact may even be greater in states without special 'FOB' laws. These states would have the potential to possibly double savings. Some additional benefits derived from this method aside from the financial aspects are: (1) uniform methodology used throughout the distribution network for less-than-truckload shipments; (2) automation of a time consuming manual process; and (3) a means to track and monitor the shipments which in turn will provide a means for correct billing. In view of these comments this approach has the potential to provide increased tangible and intangible savings for Coors transportation and distribution system.

REFERENCES CITED

1. Gendreau, Michel, Alain Hertz, and Gilbert Laporte. 1994. A Tabu Search Heuristic for the Vehicle Routing Problem. Management Science (October): 1276 - 1290.

2. Clarke, G. and J. W. Wright. 1964. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points.. Operations Research (Vol.12): 568 - 581.

3. Noon, Charles E., John Mittenthal and Rekha Pillai. 1994. A TSSP + 1 Decomposition Strategy for the Vehicle Routing Problem. European Journal of Operational Research (Vol. 79): 524 - 536.

4. Gillett, Billy E. and Leland R. Miller. 1974. A Heuristic Algorithm for the Vehicle-Dispatch Problem. Operations Research (Vol. 22): 340 - 349.

5. Balakrishnan, Nagraj. 1993. Simple Heuristic for the Vehicle Routing Problem with Soft Time Windows. Journal of Operations Research Society (Vol. 44): 279 - 287.

6. Fisher, Marshall L.. 1994. Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees. Operations Research (Vol. 42): 626 - 642.

7. Wilger, Thomas. 1992. A Heuristic-Based Algorithm for Solving Multiple Supply Vehicle Routing Problems. T-4215. Golden, Colorado: The Colorado School of Mines.

8. Christofides, N., A. Mingozzi and P. Toth. 1981. Exact Algorithms for the Vehicle Routing Problem, Based on Spanning Tree and Shortest Path Relaxations. Mathematical Programming (Vol. 20): 255 - 282.

9. Reeves, Colin R. 1993. Modern Heuristic Techniques for Combinatorial Problems. New York: Halsted Press.

10. Glover, Fred. 1990. Tabu Search: A Tutorial. Interfaces (Vol. 20): 74-94.

11. Glover, Fred. 1977. Heuristic for Integer Programming Using Surrogate Constraints. Decision Sciences (Vol. 8): 156 -166.

12. Ingber, Lester. 1993. Simulated Annealing: Practice versus Theory. Mathematical and Computer Modelling (Vol. 18): 29 -57.

13. Pate, Bill, Darrin Vohs, and Tom Wilger. 1994. Computer Integrated Logistics - Load Configurator Analysis Report. Coors Brewing Company, Golden, Colorado, November 8, 1994.

14. Coyle, John J., Edward J. Bardi, and C. John Langely, Jr. 1988. The Management of Business Logistics. St. Paul, MN: West Publishing Company

15. Vendor: Pitney Bowes Transportation Software, Bloomington, MN. System Name: PB PathwayData Services.

16. Vendor: TanData Corporation, Tulsa, OK. System Name: Progistics - LTL Processing System.

17. Vendor: Logistics Information Systems, Wellesley, MA. System Name: maxPAYLOAD.

18. Vendor: Uniteq Application Systems, Inc., Santa Clara, CA. System Name: Distribution Logistics Management System.

19. Vendor: Distribution Sciences, Inc., Des Plaines, IL. System Name: Preshipment Planning/400.

20. Vendor: Data Group, Gloucester City, NJ. System Name: Logistics Management Software.

21. Vendor: Rand-McNally - TDM, Inc. System Name: MileMaker

SELECTED BIBLIOGRAPHY

Bertanzetti, Arthur. 1994. Solution of the Axle Weight Distribution Problem for Piggyback Trailers Using a Mixed Integer, Two Algorithm Method. T-4566. Golden, Colorado: The Colorado School of Mines.

Dror, Moshe, Gilbert Laporte and Pierre Trudeau. 1994. Vehicle Routing with Split Deliveries. Discrete Applied Mathematics (Vol. 50): 239 - 254.

Gaudioso, M. and G. Paletta. 1992. A Heuristic for Periodic Vehicle Routing Problem. Transportation Science (Vol. 26): 86 - 92.

Lee, Chi-Kang. 1994. A Multiple Path Routing Strategy for Vehicle Route Guidance Systems. Transportation Research (Vol. 2): 185 - 195.

Powell, Warren B. and Ioannis A. Koskosidis. 1992. Shipment Routing Algorithms with Tree Constraints. Transportation Science (Vol. 26): 230 - 245.

Woolsey, Robert E. D. and Ruth Maurer. 1991. Useful Management Series. Golden, Colorado: The Colorado School of Mines.

APPENDIX A

Table A-1

Example 2
Customer Demand during Week 35

Distributor	Weight
Tiffin	27,843
Lima	15,294
New Martinsville	4,872
Parkersburg	16,969
Logan	14,353
Welch	17,184
So. Charleston	22,014
Rainelle	21,891
Weirton	14,658
Bellaire	21,969
Cincinnati	19,488
Brooklyn	20,412

Table A-2
Example 2
Distance Matrix for Week 35

	Broo	Cinc	Lima	Bell	Tiff	Loga	NMar	Park	Rain	SCha	Weir	Welc
Columbus	134	110	91	129	86	210	134	111	262	159	146	275
Brooklyn	0	243	145	144	82	311	174	182	351	259	121	363
Cincinnati	243	0	130	236	174	217	241	191	293	190	253	284
Lima	145	130	0	214	63	297	220	201	350	247	217	363
Bellaire	144	236	214	0	171	220	33	91	236	168	33	272
Tiffin	82	174	63	171	0	296	198	189	348	245	170	361
Logan	311	217	297	220	296	0	179	135	114	64	252	67
New Martinsville	174	241	220	33	198	179	0	50	182	127	66	231
Parkersburg	182	191	201	91	189	135	50	0	175	83	123	187
Rainelle	351	293	350	236	348	114	182	175	0	104	243	95
S. Charleston	259	190	247	168	245	64	127	83	104	0	200	112
Weirton	121	253	217	33	170	252	66	123	243	200	0	304
Welch	363	284	363	272	361	67	231	187	95	112	304	0

Table A-3

Example 2
Distance Saved Matrix - Week 35

	Broo	Cinc	Lima	Bell	Tiff	Loga	NMar	Park	Rain	SCha	Weir	Welc
Brooklyn	0	1	80	119	138	33	94	63	45	34	159	46
Cincinnati	1	0	71	3	22	103	3	30	79	79	3	101
Lima	80	71	0	6	114	4	5	1	3	3	20	3
Bellaire	119	3	6	0	44	119	230	149	155	120	242	132
Tiffin	138	22	114	44	0	0	22	8	0	0	62	0
Logan	33	103	4	119	0	0	165	186	358	305	104	418
New Martinsville	94	3	5	230	22	165	0	195	214	166	214	178
Parkersburg	63	30	1	149	8	186	195	0	198	187	134	199
Rainelle	45	79	3	155	0	358	214	198	0	317	165	442
S. Charleston	34	79	3	120	0	305	166	187	317	0	105	322
Weirton	159	3	20	242	62	104	214	134	165	105	0	117
Welch	46	101	3	132	0	418	178	199	442	322	117	0

Table A-4

Example 2
Routes Scheduled - Week 35

Route (including SDC)	Total Load	% Full
Welch - Rainelle - New Martinsville	43,947	98%
So. Charleston - Logan	36,367	81%
Wierton - Bellaire	36,627	81%
Tiffin - Lima	43,137	96%
Brooklyn - Parkersburg	37,381	83%
Cincinnati	19,488	43%

Table A - 5

Example 2
Cost Analysis - Week 35

New Cost	Old Cost	Savings
$2,748	$3,052	$304

Table A - 6

Example 3
Customer Demand - Week 3

Distributor	Weight
Rainelle	15,443
Huntington	25,373
Parkersburg	18,071
Bellaire	9,694

Table A - 7

Example 3
Distance Matrix - Week 3

	Bell	Hunt	Park	Rain
Columbus	129	134	111	262
Bellaire	0	192	91	236
Huntington	192	0	102	150
Parkersburg	91	102	0	175
Rainelle	236	150	175	0

Table A - 8

Example 3
Distance Saved Matrix - Week 3

	Bell	Hunt	Park	Rain
Bell	0	71	149	155
Hunt	71	0	143	246
Park	149	143	222	198
Rain	155	246	198	0

Table A-9

Example 3
Routes Scheduled - Week 3

Routes (including SDC)	Total Load	% Full
Huntington - Rainelle	40,816	91%
Bellaire - Parkersburg	27,765	62%

Table A - 10

Example 3
Cost Analysis - Week 3

New Cost	Old Cost	Savings
$1,067	$1,067	$0

Table A - 11

Example 4
Customer Demand - Week 8

Distributor	Weight
Youngstown	16,599
Lorain	28,246
Clarksburg	14,381
Kingwood	9,722
Bellaire	16,864
Parkersburg	15,864
Welch	16,957
Logan	19,370
Bluefield	17,871
Mabscott	21,685 24,977
Chillicothe	19,507

Table A - 12

Example 4
Distance Matrix - Week 8

	Lora	Chil	Bell	Youn	Blue	Clar	King	Loga	Mabs	Park	Welc
Columbus	124	45	129	168	272	181	224	210	225	111	275
Lorain	0	169	147	94	364	249	239	314	317	185	366
Chillicothe	169	0	151	208	229	168	219	165	182	98	232
Bellaire	147	151	0	89	270	118	103	220	223	91	272
Youngstown	94	208	89	0	345	167	152	301	296	172	349
Bluefield	364	229	270	345	0	188	229	102	49	185	51
Clarksburg	249	168	118	167	188	0	51	183	139	70	192
Kingwood	239	219	103	152	229	51	0	224	180	121	233
Logan	314	165	220	301	102	183	224	0	72	135	67
Mabscott	317	182	223	296	49	139	180	72	0	138	53
Parkersburg	185	98	91	172	185	70	121	135	138	0	187
Welch	366	232	272	349	51	192	233	67	53	187	0

Table A - 13

Example 4
Distance Saved Matrix - Week 8

	Lora	Chil	Bell	Youn	Blue	Clar	King	Loga	Mabs	Park	Welc
Lora	0	0	106	198	32	56	109	20	32	50	33
Chil	0	0	23	5	88	58	50	90	88	58	88
Bell	106	23	0	208	131	192	250	119	131	149	132
Youn	198	5	208	0	95	182	240	77	97	107	94
Blue	32	109	312	440	0	286	344	181	201	211	198
Clar	56	58	192	182	265	0	354	208	267	222	264
King	109	50	250	240	267	354	0	210	269	214	266
Loga	20	90	119	77	380	208	210	0	363	186	418
Mabs	32	88	131	97	448	267	269	363	0	198	447
Park	50	58	149	107	198	222	214	186	198	0	199
Welc	33	88	132	94	496	264	266	418	447	199	0

Table A - 14

Example 4
Routes Scheduled - Week 8

Routes (including SDC)	Total Load	% Full
Bluefield - Welch	34,828	77%
Logan - Mabscott	44,347	99%
Kingwood - Clarksburg - Bellaire	40,967	91%
Parkersburg - Mabscott	37,549	83%
Youngstown - Lorain	44,845	100%
Chillicothe	19,507	43%

Table A - 15

Example 4
Cost Analysis - Week 8

New Cost	Old Cost	Savings
$3,309	$3,637	$328

Table A - 16

Example 5
Customer Demand - Week 9 & 10

Distributor	Weight
Chillicothe	39,107 2,436
Parkersburg	23,239 23,847
Clarksburg	14,620 16,929
Bluefield	4,911
South Charleston	14,174
Wierton	17,299
Bellaire	9,046
Henderson	12,227 5,924
Huntington	21,474
Athens	22,483
Tiffin	22,021 23,613
Kingwood	16,138
New Martinsville	4,812

Table A - 17

Example 5
Distance Matrix - Week 9 & 10

	Hend	Athe	Chil	Bell	Tiff	Blue	Clar	King	NMar	Park	SCha	Weir	Hunt
Columbus	333	74	45	129	86	272	181	224	134	111	159	146	134
Henderson	0	375	313	459	386	455	481	532	461	411	369	476	322
Athens	375	0	62	122	160	211	107	158	87	37	98	144	86
Chillicothe	313	62	0	151	131	229	168	219	148	98	116	168	89
Bellaire	459	122	151	0	171	270	118	103	33	91	168	33	192
Tiffin	386	160	131	171	0	358	253	266	198	189	245	170	220
Bluefield	455	211	229	270	358	0	188	229	229	185	114	302	160
Clarksburg	481	107	168	118	253	188	0	51	54	70	131	125	172
Kingwood	532	158	219	103	266	229	51	0	86	121	172	110	219
New Martinsville	461	87	148	33	198	229	54	86	0	50	127	66	152
Parkersburg	411	37	98	91	189	185	70	121	50	0	83	123	102
S. Charleston	369	98	116	168	245	114	131	172	127	83	0	200	47
Weirton	476	144	168	33	170	302	125	110	66	123	200	0	222
Huntington	322	86	89	192	220	160	172	219	152	102	47	222	0

Table A - 18

Example 5
Distance Saved Matrix - Week 9&10

	Hend	Athe	Chil	Bell	Tiff	Blue	Clar	King	NMar	Park	SCha	Weir	Hunt
Hend	0	32	65	3	33	150	33	25	6	33	123	3	145
Athe	32	0	57	81	0	135	148	140	121	148	135	76	122
Chil	65	57	0	23	0	88	58	50	31	58	88	23	90
Bell	3	81	23	0	44	131	192	250	230	149	120	242	71
Tiff	33	0	0	44	0	0	14	44	22	8	0	62	0
Blue	150	135	88	131	0	0	265	267	177	198	317	116	246
Clar	33	148	58	192	14	265	0	354	261	222	209	202	143
King	25	140	50	250	44	267	354	0	272	214	211	260	139
NMar	6	121	31	230	22	177	261	272	0	195	166	214	116
Park	33	148	58	149	8	198	222	214	195	0	187	134	143
SCha	123	135	88	120	0	317	209	211	166	187	0	105	246
Weir	3	76	23	242	62	116	202	260	214	134	105	0	58
Hunt	145	122	90	71	0	246	143	139	116	143	246	58	0

Table A - 19

Example 5
Routes Scheduled - Week 9&10

Route(including SDC)	Total Load	% Full
Chillicothe	41,543	92%
Kingwood - Clarksburg - New Martinsville	37,879	84%
Bluefield - So. Charleston - Huntington	40,559	90%
Weirton - Bellaire - Clarksburg	40,965	91%
Athens	22,483	50%
Parkersburg	23,847	53%
Parkersburg	23,239	52%
Henderson	18,151	40%
Tiffin	45,000	100%

Table A - 20

Example 5
Cost Analysis - Week 9 & 10

New Cost	Old Cost	Savings
$3,775	$4,058	$283

Table A - 21

Example 6
Customer Demand - Week 6

Distributor	Weight
Bluefield	14,872
Huntington	12,300
Bellaire	14.189
Parkersburg	15,481
Morgantown	14,040
Clarksburg	10,717
Kingwood	13,444

Table A - 22

Example 6
Distance Matrix - Week 6

	Bell	Blue	Clar	Hunt	King	Morg	Park
Columbus	129	272	181	134	224	201	111
Bellaire	0	270	118	192	103	80	91
Bluefield	270	0	188	160	229	217	185
Clarksburg	118	188	0	172	51	39	70
Huntington	192	160	172	0	219	207	102
Kingwood	103	229	51	219	0	23	121
Morgantown	80	217	39	207	23	0	109
Parkersburg	91	185	70	102	121	109	0

Table A - 23

Example 6
Distance Saved Matrix - Week 6

	Bell	Blue	Clar	Hunt	King	Morg	Park
Bellaire	0	131	192	71	250	250	149
Bluefield	131	0	265	246	267	256	198
Clarksburg	192	265	0	143	354	343	222
Huntington	71	246	143	0	139	128	143
Kingwood	250	267	354	139	0	402	214
Morgantown	250	256	343	128	402	0	203
Parkersburg	149	198	222	143	214	203	0

Table A - 24

Example 6
Routes Scheduled - Week 6

Route(including SDC)	Total Load	% Full
Kingwood - Morgantown - Clarksburg	38,201	85%
Bluefield - Huntington - Parkersburg	42,653	95%
Bellaire	14,189	32%

Table A - 25

Example 6
Cost Analysis - Week 6

New Cost	Old Cost	Savings
$1,800	$1,800	$0

Table A - 26

Example 7
Customer Demand - Week 7

Distributor	Weight
Morgantown	16,643
Clarksburg	21,965
Bellaire	17,650
New Martinsville	3,869
Rainelle	17,153
Henderson	16,972

Table A - 27

Example 7
Distance Matrix - Week 7

	Bell	Clar	Morg	NMar	Rain
Columbus	129	181	201	134	262
Bellaire	0	118	80	33	236
Clarksburg	118	0	39	54	128
Morgantown	80	39	0	63	157
New Martinsville	33	54	63	0	182
Rainelle	236	128	157	182	0

Table A - 28
Example 7
Distance Saved Matrix - Week 7

	Bell	Clar	Morg	NMar	Rain
Bellaire	0	192	250	230	155
Clarksburg	192	0	343	261	315
Morgantown	270	363	0	292	326
New Martinsville	230	261	272	0	214
Rainelle	155	315	306	214	0

Table A - 29

Example 7
Routes Scheduled - Week 7

Route(including SDC)	Total Load	% Full
Morgantown - Clarksburg - New Martinsville	42,477	94%
Bellaire - Rainelle	34,803	77%
Henderson	16,972	38%

Table A - 30

Example 7
Cost Analysis - Week 7

New Cost	Old Cost	Savings
$1,317	$1,367	$50

Table A - 31

Example 8
Customer Demand - Week 5

Distributor	Weight
New Martinsville	6,074
Bellaire	19,339
Welch	21,508
Weirton	14,616
Henderson	10,246

Table A - 32

Example 8
Distance Matrix - Week 5

	Hend	Bell	NMar	Weir	Welc
Columbus	333	129	134	146	275
Henderson	0	459	461	476	433
Bellaire	459	0	33	33	272
New Martinsville	461	33	0	66	231
Weirton	476	33	66	0	304
Welch	433	272	231	304	0

Table A - 33

Example 8
Distance Saved Matrix - Week 5

	Hend	Bell	NMar	Weir	Welc
Henderson	0	3	6	3	175
Bellaire	3	0	230	242	132
New Martinsville	6	230	0	214	178
Weirton	3	242	214	0	117
Welch	175	132	178	117	0

Table A - 34

Example 8
Routes Scheduled - Week 5

Route(including SDC)	Total Load	% Full
Weirton - Bellaire - New Martinsville	40,029	89%
Welch	10,246	23%
Henderson	21,508	48%

Table A - 35

Example 8
Cost Analysis - Week 5

New Cost	Old Cost	Savings
$1,463	$1,478	$15

Table A - 36

Example 9
Customer Demand - Week 50

Distributor	Weight
Belaire	19,135
New Martinsville	5,945
Clarksburg	7,027
Kingwood	10,672
Parkersburg	20,399
Bluefield	6,712

Table A - 37

Example 9
Distance Matrix - Week 50

	Bell	Blue	Clar	King	NMar	Park
Columbus	129	272	181	224	134	111
Bellaire	0	270	118	103	33	91
Bluefield	270	0	188	229	229	185
Clarksburg	118	188	0	51	54	70
Kingwood	103	229	51	0	86	121
New Martinsville	33	229	54	86	0	50
Parkersburg	91	185	70	121	50	0

Table A - 38

Example 9
Distance Saved Matrix - Week 50

	Bell	Blue	Clar	King	NMar	Park
Bell	0	131	192	250	230	149
Blue	131	0	265	267	177	198
Clar	192	265	0	354	261	222
King	250	267	354	0	272	214
NMar	230	177	261	272	0	195
Park	149	198	222	214	195	0

Table A - 39

Example 9
Routes Scheduled - Week 50

Route(including SDC)	Total Load	% Full
Kingwood - Clarksburg - New Martinsville - Bellaire	42,779	95%
Parkersburg - Bluefield	27,111	60%

Table A - 40

Example 9
Cost Analysis - Week 50

New Cost	Old Cost	Savings
$1,419	$1,562	$143

Table A - 41

Example 10
Customer Demand - Week 4

Distributor	Weight
Weirton	6,810
Parkersburg	8,486
Bellaire	16,288
Morgantown	16,241
Kingswood	14,242
Huntington	27,583
Rainelle	5,002

Table A - 42

Example 10
Distance Matrix - Week 4

	Bell	Hunt	King	Morg	Park	Rain	Weir
Columbus	129	134	224	201	111	262	146
Bellaire	0	192	103	80	91	236	33
Huntington	192	0	219	207	102	150	222
Kingwood	103	219	0	23	121	171	110
Morgantown	80	207	23	0	109	157	87
Parkersburg	91	102	121	109	0	175	123
Rainelle	236	150	171	157	175	0	243
Weirton	33	222	110	87	123	243	0

Table A - 43

Example 10
Distance Saved Matrix - Week 4

	Bell	Hunt	King	Morg	Park	Rain	Weir
Bell	0	71	250	250	149	155	242
Hunt	71	0	139	128	143	246	58
King	250	139	0	402	214	315	260
Morg	250	128	402	0	203	306	260
Park	149	143	214	203	0	198	134
Rain	155	246	315	306	198	0	165
Weir	242	58	260	260	134	165	0

Table A - 44

Example 10
Routes Scheduled - Week 4

Route(including SDC)	Total Load	% Full
Kingwood - Morgantown - Weirton	37,293	83%
Rainelle - Huntington - Parkersburg	41,071	91%
Bellaire	16,288	36%

Table A - 45

Example 10
Cost Analysis - Week 4

New Cost	Old Cost	Savings
$1,637	$1,613	($24)

APPENDIX B

Table B-1

Map of Columbus Satellite
Distribution Center Network

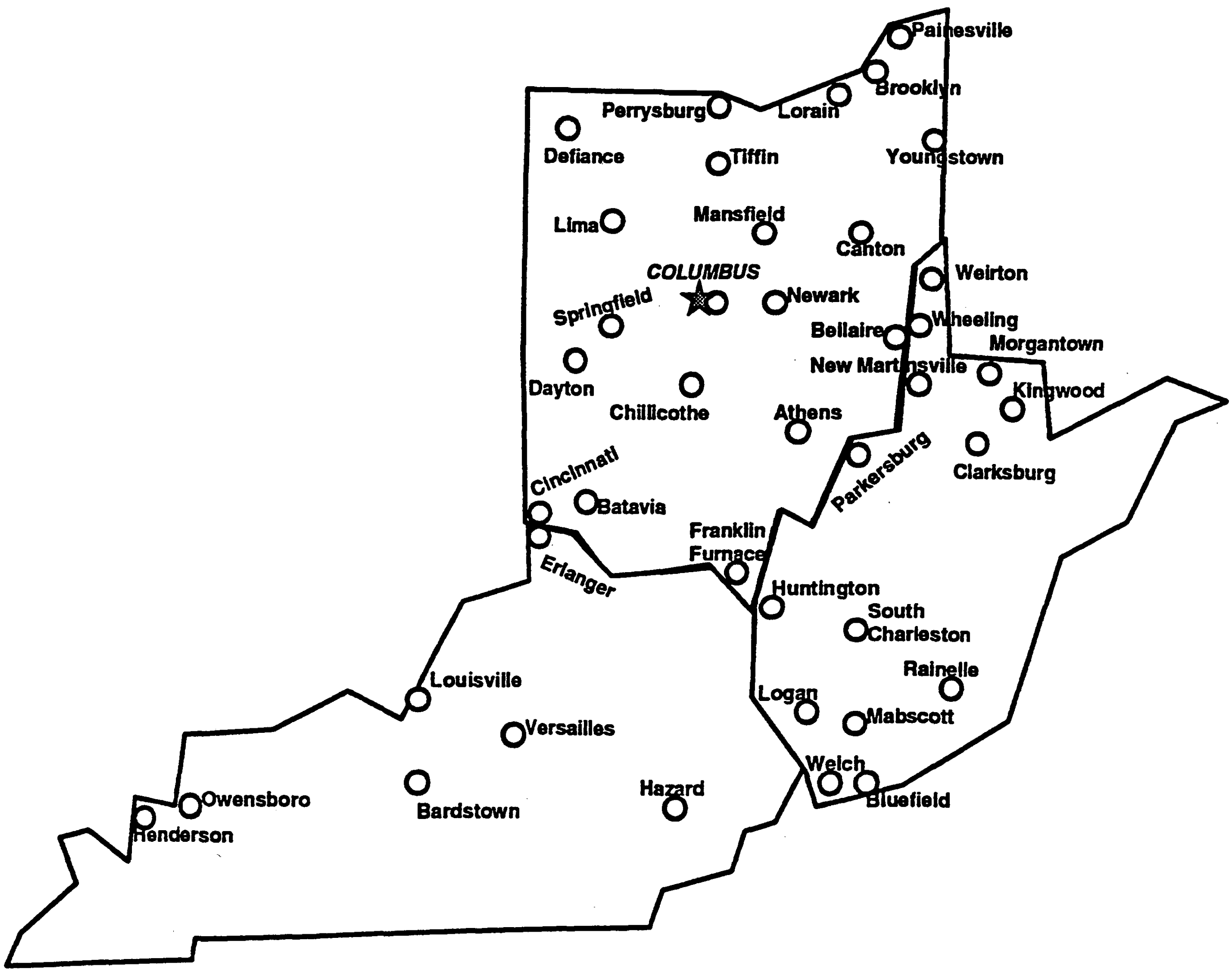

Table B-2

Distance Matrix for Columbus Satellite Distribution Center Network

	Owen	Vers	Bard	Erla	Haza	Hend	Loui	Lora	Athe	Bata	Broo	Cant	Chil	Cinc	Dayt	Defi	FrFu	Lima	Mans	Newa
Columbus	315	195	239	118	244	333	211	124	74	99	134	117	45	110	74	133	103	91	64	34
Owensboro	0	169	121	197	271	30	104	432	357	224	448	431	295	205	259	339	290	318	376	348
Versailles	169	0	48	77	131	188	66	312	201	104	328	311	167	85	139	256	129	215	256	228
Bardstown	121	48	0	121	177	151	40	356	247	148	372	355	213	129	183	300	175	259	300	272
Erlanger	197	77	121	0	187	215	93	235	160	27	251	234	98	8	62	179	112	138	179	151
Hazard	271	131	177	187	0	301	191	368	228	200	378	356	199	188	238	356	144	314	308	265
Henderson	30	188	151	215	301	0	122	449	375	242	466	449	313	223	277	347	308	326	394	366
Louisville	104	66	40	93	191	122	0	328	253	120	344	327	191	101	155	260	186	186	272	244
Lorain	432	312	356	235	368	449	328	0	177	221	28	76	169	227	181	125	227	126	60	110
Athens	357	201	247	160	228	375	253	177	0	133	186	136	62	155	140	206	84	164	117	67
Batavia	224	104	148	27	200	242	120	221	133	0	233	216	71	25	59	177	100	135	163	132
Brooklyn	448	328	372	251	378	466	344	28	186	233	0	66	179	243	200	144	237	145	72	119
Canton	431	311	355	234	356	449	327	76	136	216	66	0	157	226	190	177	215	152	62	91
Chillicothe	295	167	213	98	199	313	191	169	62	71	179	157	0	93	81	174	58	132	109	66
Cincinnati	205	85	129	8	188	223	101	227	155	25	243	226	93	0	54	171	111	130	171	143
Dayton	259	139	183	62	238	277	155	181	140	59	200	190	81	54	0	118	139	76	129	107
Defiance	339	256	300	179	356	347	260	125	206	177	144	177	174	171	118	0	232	42	120	155
Franklin Furnace	290	129	175	112	144	308	186	227	84	100	237	215	58	111	139	232	0	190	167	124
Lima	318	215	259	138	314	326	186	126	164	135	145	152	132	130	76	42	190	0	92	113
Mansfield	376	256	300	179	308	394	272	60	117	163	72	62	109	171	129	120	167	92	0	50
Newark	348	228	272	151	265	366	244	110	67	132	119	91	66	143	107	155	124	113	50	0
Painesville	480	360	404	283	410	498	376	56	206	265	37	70	211	275	233	177	269	178	104	151
Springfield	281	161	205	84	260	299	177	154	118	76	172	161	70	76	29	111	128	69	100	78
Bellaire	441	318	364	244	335	459	337	147	122	222	144	82	151	236	202	243	205	214	128	103
Tiffin	378	259	303	182	330	386	275	63	160	174	82	105	131	174	126	73	189	63	51	97
Perrysburg	388	279	323	202	366	396	295	85	196	194	103	138	167	194	146	49	225	72	93	140
Youngstown	482	362	406	285	407	500	378	94	187	167	76	51	208	277	241	210	266	202	112	142
Bluefield	425	285	331	304	154	455	345	364	211	292	361	300	229	303	309	402	196	360	321	271
Clarksburg	463	306	352	266	298	481	359	249	107	239	239	185	168	261	247	313	185	271	206	162
Huntington	304	140	186	144	123	322	200	253	86	132	262	220	89	143	170	263	32	221	193	143
Kingwood	514	356	402	317	339	532	410	239	158	290	224	174	219	312	397	338	236	309	223	198
Logan	375	211	257	218	125	393	271	314	158	206	311	250	165	217	246	339	110	297	268	218
Mabscott	414	250	296	257	197	432	310	317	164	245	314	253	182	256	262	355	149	313	274	224
Morgantown	502	244	390	305	327	520	398	216	146	278	201	151	207	300	274	315	224	286	200	175
New Martinsville	443	286	332	246	294	461	339	177	87	219	174	113	148	241	207	262	165	220	151	108
Parkersburg	393	236	282	196	250	411	289	185	37	169	182	121	98	191	177	243	115	201	142	92
Rainelle	451	287	333	294	239	469	347	354	201	282	351	290	219	293	299	392	186	350	311	261
S. Charleston	351	187	233	191	170	369	247	262	98	181	259	198	116	190	196	289	85	247	208	158
Weston	458	335	381	261	367	476	354	136	144	239	121	65	168	253	219	242	226	217	127	120
Welch	413	251	297	285	142	433	311	366	214	273	363	302	232	284	312	405	177	363	323	274
Wheeling	439	316	362	242	340	457	335	144	122	220	141	79	149	234	200	240	207	212	125	101

	Pain	Spri	Bell	Tiff	Perr	Youn	Blue	Clar	Hunt	King	Loga	Mabs	Morg	NMar	Park	Rain	SCha	West	Welc	Whee
Columbus	166	45	129	86	122	168	272	181	134	224	210	225	201	134	111	262	159	146	275	127
Owensboro	480	281	441	378	388	482	425	463	304	514	375	414	502	443	393	451	351	458	413	439
Versailles	360	161	318	259	279	362	285	306	140	356	211	250	244	286	236	287	187	335	251	316
Bardstown	404	205	364	303	323	406	331	352	186	402	257	296	390	332	282	333	233	381	297	362
Erlanger	283	84	244	182	202	285	304	266	144	317	218	257	305	246	196	294	191	261	285	242
Hazard	410	260	335	330	366	407	154	298	123	339	125	197	327	294	250	239	170	367	142	340
Henderson	498	299	459	386	396	500	455	481	322	532	393	432	520	461	411	469	369	476	433	457
Louisville	376	177	337	275	295	378	345	359	200	410	271	310	398	339	289	347	247	354	311	335
Lorain	56	154	147	63	85	94	364	249	253	239	314	317	216	177	185	354	262	136	366	144
Athens	206	118	122	160	196	187	211	107	86	158	158	164	146	87	37	201	98	144	214	122
Batavia	265	76	222	174	194	167	292	239	132	290	206	245	278	219	169	282	181	239	273	220
Brooklyn	37	172	144	82	103	76	361	239	262	224	311	314	201	174	182	351	259	121	363	141
Canton	70	161	82	105	138	51	300	185	220	174	250	253	151	113	121	290	198	65	302	79
Chillicothe	211	70	151	131	167	208	229	168	89	219	165	182	207	148	98	219	116	168	232	149
Cincinnati	275	76	236	174	194	277	303	261	143	312	217	256	300	241	191	293	190	253	284	234
Dayton	233	29	202	126	146	241	309	247	170	397	246	262	274	207	177	299	196	219	312	200
Defiance	177	111	243	73	49	210	402	313	263	338	339	355	315	262	243	392	289	242	405	240
Franklin Furnace	269	128	205	189	225	266	196	185	32	236	110	149	224	165	115	186	85	226	177	207
Lima	178	69	214	63	72	202	360	271	221	309	297	313	286	220	201	350	247	217	363	212
Mansfield	104	100	128	51	93	112	321	206	193	223	268	274	200	151	142	311	208	127	323	125
Newark	151	78	103	97	140	142	271	162	143	198	218	224	175	108	92	261	158	120	274	101
Painesville	0	204	140	115	136	57	370	223	290	208	320	323	185	173	191	341	268	110	372	137
Springfield	204	0	173	99	119	212	298	225	159	268	235	251	245	178	155	288	185	190	301	171
Bellaire	140	173	0	171	204	89	270	118	192	103	220	223	80	33	91	236	168	33	272	5
Tiffin	115	99	171	0	45	143	358	253	220	266	296	311	243	198	189	348	245	170	361	168
Perrysburg	136	119	204	45	0	169	394	295	256	299	332	347	276	234	231	384	281	203	397	201
Youngstown	57	212	89	143	169	0	345	167	271	152	301	296	129	122	172	285	249	56	349	86
Bluefield	370	298	270	358	394	345	0	188	160	229	102	49	217	229	185	85	114	302	51	275
Clarksburg	223	225	118	253	295	167	188	0	172	51	183	139	39	54	70	128	131	125	192	113
Huntington	290	159	192	220	256	271	160	172	0	219	74	113	207	152	102	150	47	222	141	197
Kingwood	208	268	103	266	299	152	229	51	219	0	224	180	23	86	121	171	172	110	233	98
Logan	320	235	220	296	332	301	102	183	74	224	0	72	212	179	135	114	64	252	67	225
Mabscott	323	251	223	311	347	296	49	139	113	180	72	0	168	182	138	39	67	254	53	228
Morgantown	185	245	80	243	276	129	217	39	207	23	212	168	0	63	109	157	160	87	221	75
New Martinsville	173	178	33	198	234	122	229	54	152	86	179	182	63	0	50	182	127	66	231	38
Parkersburg	191	155	91	189	231	172	185	70	102	121	135	138	109	50	0	175	83	123	187	96
Rainelle	341	288	236	348	384	285	85	128	150	171	114	39	157	182	175	0	104	243	95	231
S. Charleston	268	185	168	245	281	249	114	131	47	172	64	67	160	127	83	104	0	200	112	173
Weston	110	190	33	170	203	56	302	125	222	110	252	254	87	66	123	243	200	0	304	30
Welch	372	301	272	361	397	349	51	192	141	233	67	53	221	231	187	95	112	304	0	277
Wheeling	137	171	5	168	201	86	275	113	197	98	225	228	75	38	96	231	173	30	277	0