Centralized Bakery Reduces Distribution Costs Using Simulation

Ephraim Martin IV

2936 Zaharias Drive
Orlando, Florida 32837

To improve the efficiency of product distribution for a centralized bakery, I first performed each person's tasks and discovered that constructing optimal minimum-distance routes would not significantly reduce costs but replacing the physical validation of new routes with a manual mathematical computation or simulation would. The trick was getting management to trust the simulation enough to use it.

Many organizations that deliver products for a profit must frequently select delivery routes that will ensure delivery by set times. The penalty for late delivery is a fine. There is profit in minimizing travel distance or maximizing load, but achieving timeliness at low cost does more to maximize profit. When this is the case, reducing the cost of determining feasible routes may be more profitable than minimizing travel or maximizing load.

The manager of a centralized bakery asked me to work on a problem of this sort. The bakery was a part of a larger food-service operation that provided meals to a county school system and other customers as could be sold the services offered. The bakery was responsible for delivering products to over 50 delivery points, also referred to as nodes. If the bakery does not deliver products on time at a profit, its manager, my client, can lose a bonus or be fired. The manager called me in as consultant to improve the profitability of the delivery system.

From the manager's description of the problem, I assumed that minimizing distance and maximizing load efficiency would reduce costs. The bakery trucks

0092-2102/98/2804/0038/$5.00
This paper was refereed.

TRANSPORTATION—VEHICLE ROUTING
SIMULATION—APPLICATION

traveled three or more simultaneous routes and the bakery frequently added and dropped delivery points, so dynamic formulation of this problem posed a mathematical challenge, but it was not the solution to the real problem. I solved this formulation for a point scenario, which yielded routes similar to those in the solution actually employed.

Minimizing mileage and maximizing load increased profit marginally. Delivery on time maximized profit and profit growth. By working in the system, I learned that my client really needed a cheaper way to quickly pick new routes that would ensure reliable on-time delivery. Delivery on time was so important that the bakery customarily physically tested new routes to make sure they would work. This expensive process could be replaced by simulation. Doing so was the true key to reducing costs and improving profitability.

The current algorithm was embedded in a seasoned human scheduler.

What the client needed was an intelligent algorithm that developed and tested routes more efficiently than the current method of guessing and physically testing and adjusting and testing and so forth. The client's current algorithm was embedded in a seasoned human scheduler who could do the job because of great experience and familiarity with the current service area. These are not qualities that could easily be replicated or exported to new service areas. Even with her experience, the scheduler would not initiate a new route without physically testing it first. This stifled growth and made the addition of new delivery points or routes expensive.

To develop the solution, I performed all aspects of the operation at least once. I assumed the drivers' duties, which include checking out the truck, driving from the motor pool to the bakery, loading the product, and finally delivery. I ran all routes several times. I collected data during delivery that included time between delivery points, time to deliver at each point, and volume delivered at each point. Participating in the process revealed several facts.

The most important fact was that the costs associated with route mileage represented an insignificant portion of total costs. This meant that attempts to reduce route mileage would yield low returns. Physically testing routes, on the other hand, represented a major expense with a greater secondary cost of discouraging the aggressive acquisition of new delivery points. Taken together, these suggested that replacing physical route testing with simulated route testing would yield high returns. The data I had collected participating in the system suggested a way to simulate route testing.

The travel time between delivery points had small variance and was generally a small portion of the total time required to complete a route. The time it took to unload at delivery points varied much more and made up the major portion of time required to complete a route. Straight-line distance between delivery points seemed to dictate the transit time between them, and volume of product delivered seemed to dictate time to unload at a delivery point. I quantified these intuitive observations by running a regression analysis, which showed a high correlation between the straight-line distance between delivery points and the time to travel between them, as well as a high correlation between load size and unload time. The experience I gained working within the system and the data I collected suggested that a deterministic simulation could estimate delivery times as accurately as physically testing the routes. I developed a model using the following elements.

The estimate I used for travel time, T, between any two nodes was the upper bound of a 95-percent confidence interval estimate [Walpole and Meyers 1989] for the regression estimate of T based on straight-line distance, S, between nodes. I used a similar estimate for unload time, U, based on load size, L. There is strong logical basis for these choices. I used a confidence-interval estimate because the clients had commented that they wanted to be at least 90-percent certain that deliveries would be on time. The 95-percent-confidence-interval estimates produced route-simulation estimates very close to the actual completion times for existing routes. A more important reason was that the client trusted these estimates and would therefore use them. In practice, these estimates routinely provided routes that could meet overall delivery time constraints in practice.

A more deep-pocketed client could make gains in accuracy by using Monte Carlo simulation or a more sophisticated deterministic simulation. Among other benefits, added sophistication and complexity would better address accumulation of variance in load and transit times. I accounted for these by safe-siding the estimates in the method actually used. This would come with higher costs to collect the data required to drive the simulation, costs to code the simulation, and higher training and salary costs for the users of the simulation. My experience building

> This algorithm is easy to use and easy to understand, and it reduces costs.

and running simulations had taught me that for this example these costs would be much greater than the gains in efficiency and would not be beneficial to the client. The bakery had no need for this increased complexity, so I dismissed these technically superior methods.

The client's rejection of unfamiliar or overly complicated solutions dictated a simple and familiar solution. For this reason, I used a nearest-neighbor heuristic to pick routes and simple, if overly conservative, estimates from the regressions to estimate completion times.

Using the confidence-interval estimates of the regression of $S \times T$, I constructed a time ruler (Figure 1). The time ruler measured the travel time between any two points based on straight-line distance. I used the confidence-interval estimates of the regression of $L \times U$ to construct a loading table (Figure 2). The loading table simulated the time to unload at a node based on load size. Employed together, these two tools can be used to quickly estimate or simulate the time required to complete a route.

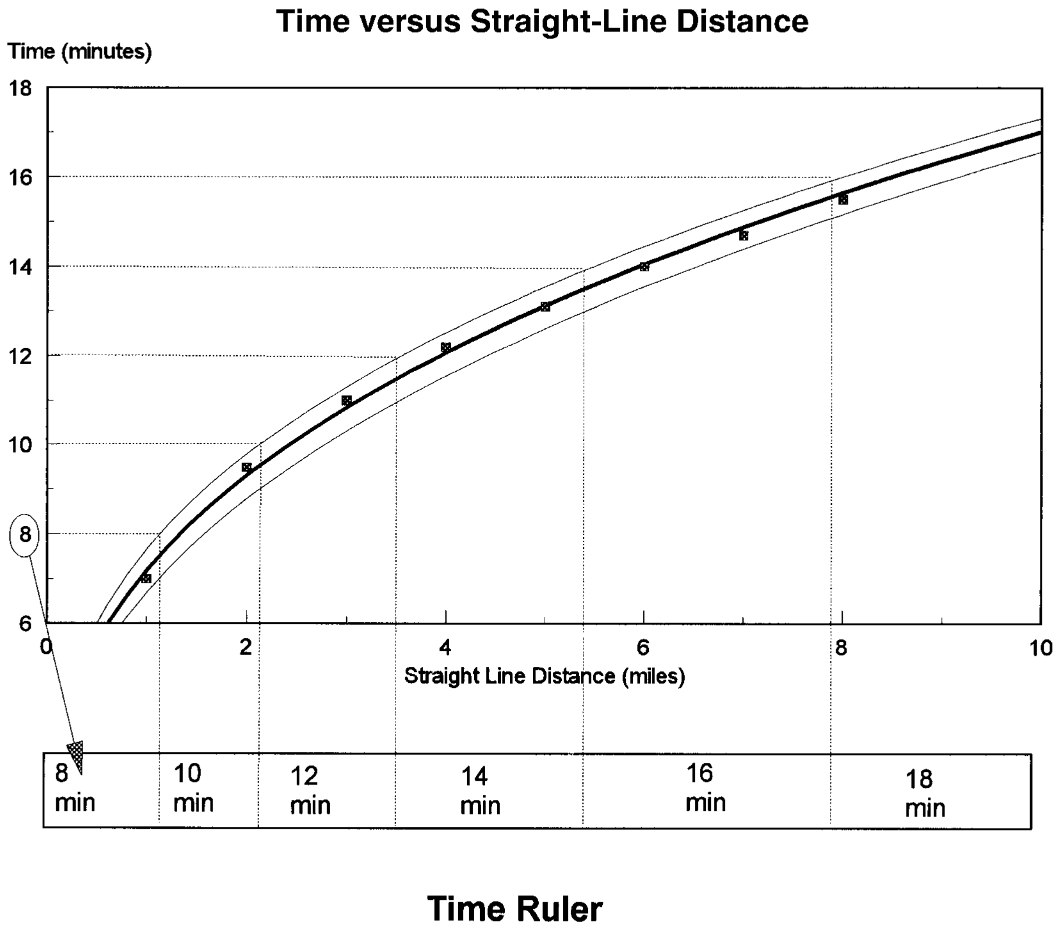
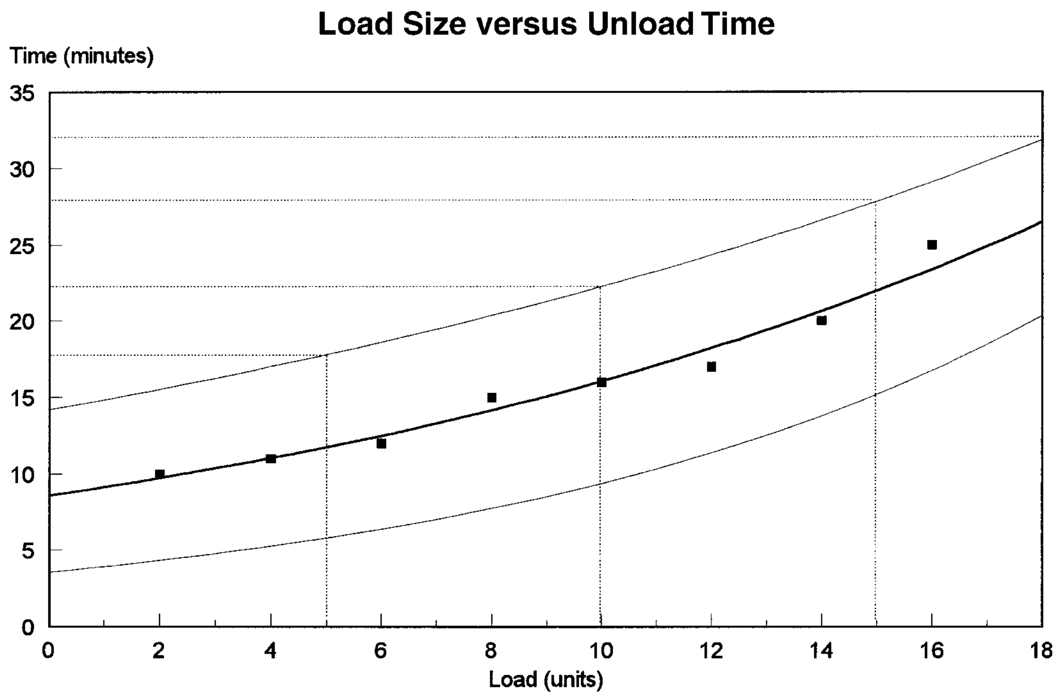


Figure 1: In this figure showing the regression of travel time versus straight-line distance between delivery points, the dark curve is the regression. The lighter curves show the upper and lower limits for a 95-percent confidence interval of the regression estimate. The time ruler is constructed using the upper limit. Using the time ruler shown, I estimated that travel between two delivery points separated by a straight-line distance of less than 1.2 miles would require eight minutes; 1.2 to 2.2 miles, 10 minutes; 1.2 to 3.5 miles, 14 minutes, and so forth.

At the time, the bakery was selecting routes using a loosely disciplined nearest-neighbor heuristic. I define the method as loosely disciplined because it usually picked the nearest neighbor as the next stop but had multiple exceptions for deviating from this procedure. From reading Bartholdi et al. [1983] and Hesse and Woolsey [1980] and from personal experience with the system, I concluded that a nearest-neighbor heuristic was probably nearly optimal. To verify this theory, I constructed minimum-distance routes using mixed-integer programming. These optimal routes yielded minimal mileage savings over the routes in use. Because the costs associated with mileage were a small portion of total cost, reducing the mileage



Loading Table

Loading Table

Load (Not to exceed)	Unload Time (minutes)
5	17
10	22
15	28
18	32

Figure 2: In this figure showing regression of time to unload versus the number of units of product being delivered, the dark curve is the regression. The lighter curves are the upper and lower limits for a 95-percent confidence interval of the regression estimate. The loading table is constructed using the upper limit. Using the loading table shown, I estimated that delivery of one to five units would require 17 minutes; six to 10 units, 22 minutes; 11 to 15 units, 28 minutes, and so forth.

required to travel scheduled routes resulted in minimal savings. The added complexity and cost of a more optimal algorithm was not justified. We used the nearest-neighbor heuristic.

To demonstrate the practicality of this system, I constructed a worksheet with instructions that allowed an untrained scheduler (my spouse) to construct a near-optimal route and compute whether or not it could be completed in the required amount of time. The system works conceptually as follows. Mark the delivery points with push pins on a map. Start at the bakery and go to the closest delivery point. Measure and estimate transit time with the time ruler. Estimate loading time with the loading table. Annotate the worksheet. Continue this process until you cannot add a new delivery point without exceeding the delivery time constraint. For the bakery situation, all deliveries had to be completed four hours after the start time.

To construct a delivery route, one starts by picking any start point. The example shown in Figure 3 uses coordinate (8,8). Go to the closest node. Using the time ruler, estimate the transit time based on straight-line distance. In this example, the transit time from the start node to Node 1 is 10 minutes. Estimate the loading time using the loading table and the units to be delivered at the node. At Node 1 this is four units, which takes 17 minutes according to the loading table in Figure 2. Add travel time and load time (to get 27 minutes as a delivery-completion time at Node 1). Check to see that delivery-completion time is less than the deadline time. As long as you meet this condition, continue the process. Go to the closest node, estimate travel time using the time ruler and load time using the loading table. Add travel time and unload time to the last delivery-completion time. When delivery-completion time exceeds deadline time, stop the route and do not deliver to that node. In this example, that occurs at Node 9, and we therefore stop at Node 8. The requirement to deliver to further nodes must be met by another route.

On the example worksheet (Figure 4), the deadline is expressed as 240 minutes, since we assumed we would start the route at 6:00 AM and all the nodes had a deadline of 10:00 AM. The difference of four hours (240 minutes) provided the deadline.

The travel times in the "travel time" column are those between the node indicated and the node above, for example, START to Node 1 is 10 minutes. The time ruler is used to estimate the transit time between nodes.

The load times shown in the "unload time" column are computed using the loading table given the units to be delivered at the node.

The real test of the method was with the client. I asked the experienced scheduler to pick the existing route that was most difficult to meet. She did and then methodically simulated the route on a form similar to that shown in Figure 4 and computed the estimated delivery times. With the algorithm, she estimated that the route would take three hours and 47 minutes. She confessed that it actually took three hours and 30 minutes on a good day and never more than three hours and 45 minutes. She repeated this exercise on a second difficult route and produced an estimated performance of two hours and 13 minutes with actual performance varying between one hour and 45 minutes to two hours and 10 minutes. I had, of course, conducted these and many other tests prior to working through them with the scheduler. She was absolutely sold on the

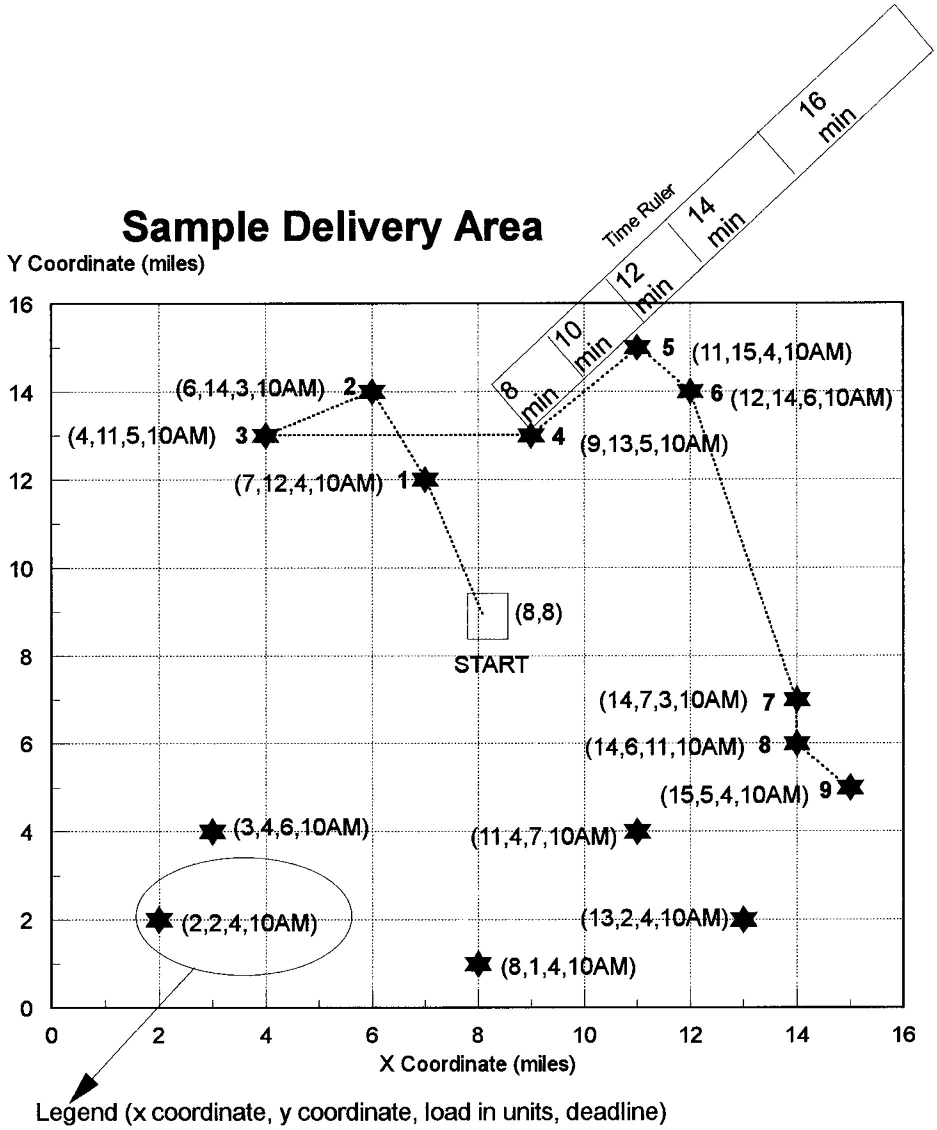


Figure 3: In this sample delivery area, the stars represent delivery points or nodes. Points 1 through 9 are used to construct a sample route on the worksheet shown in Figure 4. Each delivery point has a coordinate vector. The coordinate vector for delivery point 1 is (7, 12, 4, 10AM). The first element, 7, is the x distance in miles from the origin. The second element, 12, is the the y distance in miles from the origin. The origin can be any point on the map. The third element, 4, is the number of units of product that must be delivered at this point. The fourth element, 10AM, is the deadline for delivery.

system. This algorithm is easy to use and easy to understand, and it reduces costs. Routes do not have to be physically tested; they can be simulated on the form. The availability of this algorithm encouraged the business to expand, and it was later

Route Construction Worksheet

	Travel Time	Unload Time	Delivery Complete	Deadline	Deliver
START					
Node 1	10	17	27	240	1
Node 2	10	22	59	240	1
Node 3	10	17	86	240	1
Node 4	14	17	117	240	1
Node 5	10	17	144	240	1
Node 6	10	22	176	240	1
Node 7	14	17	207	240	1
Node 8	8	22	237	240	1
Node 9	10	22	269	240	0

Figure 4: This is a completed route-construction worksheet for the delivery area shown in Figure 3. Travel times are estimated using the time ruler shown in Figure 1. Unload times are estimated using the loading table shown in Figure 2. The deadline assumes that the route starts at 6:00 AM and must be completed at 10:00 AM, a total of 240 minutes. Using this worksheet, I estimate that delivery to points 1 through 8 can be completed in 237 minutes.

used to bid contracts and establish contractual guarantees before setting up similar operations in new territories.

The algorithm can easily be coded in C++ or even in Basic and transported on a personal computer to a variety of users. Depending on the features desired of the software, one could handle more than 10,000 delivery points on a 486 PC. As a coded algorithm, this system for selecting and testing routes could address more complex situations and could be self-improving or intelligent with some collection of operational data. If one collects and feeds route performance, for example, to the algorithm, one can constantly update the estimates for travel time and load time and give them tighter confidence intervals with little additional effort or risk. This algorithm has possible application in any industry where products with a time value must be distributed or picked up. Application would be heavily dependent upon the parameters of the particular problem.

References

Bartholdi, John J.; Platzman, Loren K.; Collins, Lee R.; and Warden, William H. 1983, "A minimal technology routing system for meals on wheels," *Interfaces*, Vol. 13, No. 3 (June), pp. 1–8.

Hesse, Rick and Woolsey, Gene 1980, *Applied Management Science*, Science Research Associates, Chicago, Illinois, pp. 21–35.

Walpole, Ronald E. and Myers, Raymond H. 1989, *Probability and Statistics for Engineers and Scientists*, Collier Macmillan, New York, pp. 236–240.

Shirley Brooke, Food Service Coordinator, Jefferson County Public Schools, 1829 Denver West Drive, Building 27, Golden, Colorado 80401, writes: "This system is easy to use, understandable, and, most important, I feel it is reliable. On more than one occasion, it has allowed me to develop routes without the tedious task of juggling and continual physical testing. When I have used this method, it has performed within the projection of the system. This has proved valuable to us.

"I recently used this system to determine delivery routes for a bid, a new area which could provide additional income to Food Services. The system met our needs and allowed me to feel comfortable without physically driving the routes I had established. This saved me a significant amount of time and labor.

"Since we are in the business of delivering perishable products and our program is continually changing, I foresee the potential for using this system time and again in the future."