VEHICLE ROUTING AT QUALITY STORES

BY

Candace Arai Yano
with
Thomas J. Chan
Ted Cutler
Lori Kaplan
Katta G. Murty

University of Michigan

October 1985
Tech. Report #86-2

VEHICLE ROUTING AT QUALITY STORES

by

Candace Arai Yano*

with

Thomas J. Chan*

Theodore Cutler†

Lori Kaplan*

David McGettigan†

Katta G. Murty*

October 1985
Revised May 1986

* Dept. of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI.

†Quality Stores, North Muskegon, MI.

VEHICLE ROUTING AT QUALITY STORES

ABSTRACT

Vehicle routing optimization software has been implemented on a personal computer for a chain of retail stores. Customized route generation procedures are combined with an efficient branch-and-bound procedure to obtain optimal routings and assignment of routes to trucks. Both the database maintenance routines and optimization runs are completely menu-driven enabling personnel with no prior exposure to computers or optimization to operate the software unaided.

VEHICLE ROUTING AT QUALITY STORES

Corporations with private truck fleets often find that the fleets are not justified economically on the basis of outbound deliveries alone. Rather, it is the savings obtained by backhauls from suppliers (in lieu of using a common carrier) and/or transport of materials for a third party that make the private fleet worthwhile.

This is the situation at Quality Stores, a Michigan-based retail chain of approximately 40 stores with one centralized distribution center located near Toledo, Ohio. Most stores are located within a radius of 250 miles of the distribution center. Quality Stores owns and leases eleven trucks and uses the fleet to deliver goods to the stores and to pick up (or "backhaul") goods from nearby vendors. Deliveries are made to each store an average of twice a week and most deliveries represent a half-truckload up to several truckloads of goods. Quality Stores has found that savings from using its fleet (rather than common carriers) to backhaul goods from vendors in the vicinity of its stores significantly enhances the economic attractiveness of the fleet. However, because of seasonal and day-to-day fluctuations of the quantities to be transported, it is uneconomical to retain enough trucks to handle peak demand. Therefore, whatever cannot be transported by the fleet is contracted to common carriers, and it is preferred that backhauls rather than outbound deliveries be contracted out.

The Transportation Director faces the problem of jointly determining which requirements to transport using the private fleet, and how to route those trucks to meet transport requirements. With his many years of experience, he has no difficulty finding good, feasible solutions. However, it is not humanly possible to consider all possible routes and combinations of routes to guarantee minimum cost. The number of feasible route is quite large, even when the number of trucks, stores, and pickup points is small. For instance, even if there are only ten stores requiring delivery and ten supplier destinations on a given day, there may be as many as 3135 alternative ways to combine locations if it is possible to have up to two stores and up to two suppliers on a route. This leaves a huge combinatorial problem even if many of the assignments are not feasible because of truck capacity limitations and other route constraints. Moreover, for each possible combination of locations, the least-cost routing must be determined. The budget for the private fleet is large enough so that savings of a few percentage points can provide a significant improvement to the profitability of the company.

We describe an approach to this problem which combines a customized scheme for generating feasible routes with an efficient set covering procedure based on Lagrangian relaxation in a branch and bound framework. Our approach is similar in spirit to that of Bell, et al. (1983) for much larger problems solved on a mainframe computer. We also use a simple but effective heuristic to obtain an initial feasible solution, which reduces the computation time for the optimization procedure. In addition, we handle the constraint on the number of private fleet trucks by using the same set covering procedure to minimize the total number of trucks required. The algorithm is coded in TURBO PASCAL and implemented on a personal computer. Most personal computer implementations of vehicle routing software have used heuristics (e.g., Cullen, et al., 1981), or graphics (e.g., Belardo, et al., 1985). To our knowledge, the implementation reported here is the first personal-computer-based optimal solution procedure.

The reader is referred to Bodin, et al. (1983) for a recent survey on vehicle routing. In this paper, we intend to focus on the application while providing only brief explanations of the underlying optimization procedures.

DETAILS, DETAILS

Quality Stores has a fixed weekly delivery schedule with an average of 15 stores receiving deliveries each day. During peak seasons, a total of over 40 full or partial truckloads are delivered each day, while at other times of the year, there are commonly 20 to 30 deliveries. Deliveries are made Monday through Saturday and, when necessary, on Sunday. Every morning, the Transportation Director receives a report on the weight and volume of the shipment to each store for the following day. Thus, this data is already available.

The eleven trucks are nearly identical with respect to weight and volume capacities, and combinations of partial truckload deliveries must satisfy both constraints if they are to be consolidated onto the same truck. Additional trucks can be rented on short notice, or common carriers can be contracted to satisfy excess demand.

Quality Stores must also abide by federal laws which specify upper limits on driving time (ten hours) and total time on the road (fifteen hours including load and unload times) for each driver. However, each truck may be used by more than one driver on a given day. All drivers are based at the distribution center (and thus must return there at the end of the route) and, with a few minor exceptions, the policy is to use one driver per truck. In addition, deliveries to stores and pickups from suppliers must be made between opening and closing times of each destination.

Because the trucks are loaded from the rear, it is necessary to make all deliveries before picking up goods from suppliers. On occasion, it would be possible to pick up goods at supplier A, deliver the goods to store B, and then proceed to store C to deliver goods which originated from the distribution center. However, such situations are rare and are ignored in current practice.

There also are several other constraints which are too complex to discuss in detail. It suffices to say that generation of all feasible routes is not an easy task if done manually.

Thus, while the problem may appear on the surface to be a standard vehicle routing problem, the special constraints, the presence of both delivery and pickup requirements, and the necessity of considering common carrier alternatives, make the problem a complex but interesting one which many companies with small private fleets face daily.

GENERATION OF FEASIBLE ROUTES

We initially generated two types of routes for the private fleet. We refer to the first type as "cycles." These are single- and multi-destination routes beginning and ending at the warehouse, with no intermediate warehouse stops. These routes must satisfy driving time and "on the road" constraints, truck capacity limits with regard to both weight and volume, and the closing time constraints.

Cycles with both deliveries and pickups have the characteristic that all outbound deliveries must be made before a load can be picked up at a supplier. This is common practice in vehicle routing. Further, because the trucks are of the rear-loading type, and because most backhauls are destined for the distribution center, it is generally not feasible to pick up goods before the last delivery is complete. Otherwise the goods being transported to the distribution center would be at the back of the truck trailer, making it impossible to unload the other goods (at the front of the trailer) which must be delivered to stores.

We also limit the number of deliveries and pickups in one cycle to a maximum of four of each. We use the term "transport requirement" to refer to either a delivery, or a supplier pickup along with its associated deliveries. Note that the deliveries corresponding to a pickup must be done by the same truck, so it is necessary to treat them jointly as one requirement. For this application, these limits are more than adequate because even cycles with four transport requirements are rare because of truck volume and route distance constraints. Because the number of transport requirements on a route is limited, it is possible to solve the constrained travelling salesman problem (with deliveries preceding pickups) optimally (by total enumeration) to determine the shortest route. Examples of several types of cycles are illustrated in Figure 1.

One database feature permits many special routing constraints to be handled. The user specifies which store-to-store and store-to-supplier links are permitted. Thus, temporary road closings can be incorporated and unreasonable routings can be excluded. This feature reduces route generation time by eliminating in advance some routings which are likely to violate the distance or time constraints.

The second type of route encompasses combinations of cycles from the first category which also satisfy closing time constraints for the latter portions of the route. For convenience, we call a route a "figure eight" when two cycles are combined or a "cloverleaf" when three or more cycles are combined. The combined route need not satisfy the driving time and "on the road" constraints, since it is possible to change drivers between cycles. However, if the route does satisfy these constraints, the entire route may be assigned to one driver. We generated the combined routes because doing so permits a one-to-one correspondence between trucks and routes in instances where cycles can be combined. We discovered that many combinations of cycles were feasible because of early morning starts by the drivers, long store hours, and possible change of drivers between cycles. This resulted in too many routes (typically about 200 cycles and several hundred figure eights or cloverleafs) for the optimization code to handle efficiently. For this reason, we decided to limit the routes to cycles only. Unfortunately, this made incorporation of the constraint on number of private fleet trucks more difficult. We discuss resolution of this difficulty later in the paper.

We also could have generated common carrier routes, but did not do so for two reasons. First, hundreds of routes would have been generated because the route characteristics are less constrained than for the private fleet. The primary factor underlying this is that many common carrier routes can be one-way rather than round-trip, so the route distance and driving time constraints are much less binding. The second reason was that even the least expensive common carrier was frequently (but not always) more expensive than the marginal cost of using the private fleet. Thus a common carrier route would be selected only infrequently in an optimization procedure without a constraint on number of private fleet trucks.

Our experience indicates that generation of all feasible private fleet cycles (typically 150 to 250) for this application takes a few seconds at most on a personal computer, so that more sophisticated column generation procedures (see, for example Cullen, Jarvis, and Ratliff (1981)) are unnecessary. (In fact, generation of all feasible routes, including figure eights and cloverleafs, requires little time. The primary limiting factors are computer memory on a personal computer and processing time for the optimization code).

AN EFFICIENT HEURISTIC

Optimization is done using a branch and bound based procedure. As with other branch and bound approaches, having a good initial feasible solution reduces computation time. We investigated two efficient list-processing heuristics for the purpose of finding a good initial feasible solution. Descriptions of the heuristics follow.

We generate a list of all feasible cycles, sequenced (approximately) in increasing order of number of transport requirements handled. Starting at the bottom of the list, a route is added if every requirement on the route is not yet satisfied by a route already selected. These initial solutions were found to be relatively good, occasionally providing an optimal solution, and if not, one within about 30% of optimum.

In the course of using this heuristic, we found that choices regarding requirements which had few alternative routes were responsible for much of the difference between the cost of the initial feasible solution and optimality. The procedure discussed above tended to pick "good" routes (satisfying many requirements) for requirements which were easy to satisfy (in the sense of having many alternatives), and tended to pick "bad" routes (i.e., many single destination routes) for requirements which were difficult to satisfy. If the "hard to satisfy" requirements were satisfied using "good" routes, we might expect that decisions regarding the "easy to satisfy" requirements would not matter as much relative to optimality. On the basis of this logic, we developed a heuristic in which transport requirements are satisfied in increasing order of alternative routes fulfilling that requirement. Starting with the requirement with fewest alternative routes, we select a route such that: (i) the route contains the greatest number of unsatisfied requirements, and (ii) every requirement on the route has not been satisfied already. The second condition eliminates any redundancy in the initial feasible solution. We found that the second heuristic reduced the difference in cost between the initial feasible solution from the simpler heuristic and the optimal solution by approximately one third to one half and requires little additional computation time. Unfortunately, however, these better initial solutions did not lead to a singificant reduction in run-time for the optimization routine because better solutions are found quite quickly. Nevertheless, we chose to use the improved heuristic to find an initial feasible solution.

OPTIMIZATION

The objective is to minimize the total cost of the routes while ensuring that each transport requirement is satisfied. If only feasible routes are considered, the problem can be formulated as:

$$\text{minimize} \quad \sum_j c_j x_j$$

$$\text{subject to} \quad \sum_j a_{ij} x_j \geq 1, \quad \forall\, i$$

$$x_j = 0 \text{ or } 1, \quad \forall\, j$$

where c_j = cost or distance of route j

$$x_j = \begin{cases} 1 \text{ if route j is selected} \\ 0 \text{ otherwise} \end{cases}$$

$$a_{ij} = \begin{cases} 1 \text{ if transport requirement i is on route j} \\ 0 \text{ otherwise} \end{cases}$$

This is a standard set covering problem. We have used a set covering approach rather than a set partitioning approach in which the first set of constraints must be satisfied at equality. There were two reasons for this choice. First, the set covering problem is considerably easier to solve than the set partitioning problem. Second, we assume that the "triangle inequality" is always satisfied, and that therefore, it will always be more costly to satisfy a requirement more than once than to do it once only. (If the triangle inequality is not satisfied because distance data are inaccurate one can simply include stop-off charges which are so common in the current rate structures into the route costs).

We used an optimization procedure, due to Murty (1974), and Etcheberry (1977) which is a set covering routine. It uses a branch and bound framework along with an efficient lower bounding technique. The branching, unlike the typical procedure which sets a variable to zero or one, creates candidate problems (at nodes in the branch and bound tree) which are characterized by their unordered Cartesian products, or UCPs for short, and a set of unincluded constraints. Each factor in a UCP represents a set of routes. If at least one route in a UCP factor is selected, at least one constraint is satisfied. The branching creates UCP factors and reduces the number of unincluded constraints in a systematic fashion, so that eventually all constraints are satisfied.

At each node in the branch and bound tree, subgradient optimization is used to determine "good" Lagrange multipliers for those constraints which remain unsatisfied, by maximizing the dual problem. (Note that the Lagrange multipliers of any constraints already satisfied are zero). Given these Lagrange multipliers, it is straightfoward to compute the relative (adjusted) cost of each of the routes. Thus, the procedure involves selecting all routes with negative adjusted costs, and ensuring that each UCP factor has at least one route selected. The routes thus selected become the new relaxed solution with a lower bound equal to the sum of the relative costs of the included routes. If the solution is feasible to the original problem, the actual solution cost is computed. If the cost is less than that of the current incumbent solution, the solution becomes the new incumbent, providing a new upper bound, and portions of the branch and bound tree may be pruned. The process continues until no other candidate problems remain. The reader is referred to Murty (1976) for a detailed description of the algorithm.

We have found the procedure to be efficient, finding solutions for problems with twenty to forty transport requirements in ten to thirty minutes in most instances on a personal computer. We are, however, implementing improvements on the Murty-Etcheberry algorithm which take advantage of duality theory and the special structure of the VRP in order to improve running times for larger problems.

As an aside, we comment on one modification that we found to be helpful in reducing computation times during our initial investigation when figure eights and cloverleafs were included in the optimization routine. We found that the set covering algorithm would sometimes be unstable, alternating between two cycles and an equivalent figure eight, thus requiring more computation time than necessary. When there are no binding truck limitations, one may be indifferent between the two alternatives. However, there is usually a truck constraint of some sort, making the figure eight preferable in most instances. By setting λ to a small value (e.g., 1) prior to processing, and setting $c_j' = c_j + \lambda$, the tie between the alternatives is broken and the procedure is much more stable. In only extremely rare instances would the solution be altered by using a small value of λ in this way.

PRIVATE FLEET CONSTRAINT

We first describe what we intended to do, had it been possible to include figure eight and cloverleaf routes, and then describe what we did because of the limitations imposed by personal computer implementation.

Delivery and backhaul requirements sometimes exceed the capacity of the eleven trucks. To resolve this difficulty, we initially planned to relax the truck availability constraint using a Lagrange multiplier. This results in a relaxed problem in which private fleet truck routes are penalized by the amount of the Lagrange multiplier, while common carrier routes are not. Numerous common carrier alternatives are available, and it would have been desirable to include all undominated routes in the set covering algorithm. (For any set of transport requirements, only the best common carrier alternative need be included). With an appropriate choice of Lagrange multiplier, the private fleet size constraint can be satisfied at equality.

The procedure above is what we had hoped to do. However, because only private fleet cycles could be included in the optimization routine, we needed to combine these cycles into figure eights and cloverleafs (where possible) so as to maximize the utilization of the private fleet trucks. We accomplished this by using set covering again.

Using the optimal cycles from the first set covering solution, we determined which figure eights and cloverleafs were feasible. Examples of figure eights and cloverleafs are shown in Figure 2. These cloverleafs, figure eights, and the optimal cycles then comprise the "routes" which can be selected to cover the set of optimal cycles. The optimal cycles are akin to the transport requirements in the first set covering problem. The objective function minimizes the number of trucks required (i.e., $c_j = 1$ for all j). In this implementation of the set covering routine, it was necessary to check for redundancy of route coverage before updating the incumbent. It was often possible to include some of the shorter cycles several times without affecting the total number of trucks required. If the number of trucks exceeds eleven, the Transportation Director simply chooses the "best" routes and subcontracts the remainder, or rents additional trucks as necessary.

Unfortunately, this two stage procedure does not guarantee optimality for the original truck-constrained problem. It is possible that different cycles may have been selected if the private fleet constraint had been incorporated earlier. However, omission of this constraint creates a relaxed problem in the selection of cycles, resulting in a solution with a smaller total <u>distance</u>. We anticipate that these savings will compensate in part for the occasional use of more common carrier or rental trucks than would have been required by the integrated approach. In most cases, the Transportation Director knows how much can be done with the private fleet and uses his judgment to determine which backhaul requirements to include in the optimization procedure.

We are working toward inclusion of the fleet size constraint. While the potential number of common carrier routes does pose a difficult problem, the most difficult problem lies in uncertain availability of trucks from the various common carriers. Availability of trucks often depends upon the actual route the truck must travel (or at a minimum, the route distance), and the actual route "assigned" to a common carrier cannot be known until the solution is obtained. At this point, it may be discovered that such an assignment is infeasible and that assignment to another carrier will incur a significant cost penalty. Because of these difficulties, we are working on a heuristic which will first provide a "good" allocation of the work between the private fleet and common carriers and then solve the common carrier selection/routing problem in the second phase.

IMPLEMENTATION

Completely menu-driven interactive data entry and maintenance routines are used to maintain files with store-to-store, supplier-to-store, and supplier-to-supplier distances, permitted store-to-store and store-to-supplier links, as well as closing times.

The database currently allows for approximately 50 stores and 200 supplier locations. The users have opted to use supplier cities rather than supplier names to permit more flexibility in the future. The store database can be expanded to permit 100 or more stores with little difficulty. Only relevant data need to be input to the database because default distances make the store-to-store and store-to-supplier links infeasible.

Other system parameters such as average driving speed, unloading and loading time per truckload, and "on the road" limits also are maintained through interactive routines. Daily data entry consists of store delivery requirements and supplier pickup requirements, which also are input interactively.

Data files are maintained and the program is being run quite effectively (and with little assistance) by a clerk with limited experience with personal computers, and with no background on the underlying optimization procedures. The time required for daily data entry is less than the time required by the Transportation Director to construct a set of feasible routes.

The program generates all feasible cycles, in a few seconds at most, then goes on to find optimal set of cycles. The user has the option of assigning these routes to trucks, or to have the set covering routine optimize this assignment. The time required to solve one problem varies from a few minutes to about thirty minutes, depending upon the size of the problem, but requires no monitoring during execution so that little staff time is required. We are currently working to replace the subgradient optimization routine with a more efficient multiplier adjustment procedure for the set covering problem. Preliminary results indicate that the computation times will be reduced considerably. This modification should permit solution of problems with many more destinations. Nevertheless, the initial implementation represents a significant advance in solving vehicle routing problems in particular, and set covering problems in general, with a personal computer.

We have installed a feature which permits the user to stop the procedure at any time and to use the best solution obtained so far. Our experience indicates that (as with other branch and bound routines) optimal solutions are obtained relatively quickly for most problems, and that several times as long is spent "validating" that the solution is optimal. Thus, in most instances, the use can safely stop the procedure after ten or fifteen minutes.

BENEFITS

The company estimates that the vehicle routing software will help it to save nearly a half-million dollars in 1986. Most of the savings result from increasing utilization of the trucks for relatively short-distance backhauls. Since most common carriers have a minimum charge per route, short routes can be very expensive to subcontract, but relatively inexpensive to handle with private fleet trucks.

There were also more subtle benefits. The availability of the software led the company to consider acquiring additional trailers so that load and unload times could be reduced or eliminated. This would be accomplished by switching trailers instead of waiting for loading or unloading. This, in turn, has led to consideration of 20 hour per day operation instead of the usual 10 to 14 hour per day operations, since deliveries could now be made even when the stores are closed. Over the long term, the savings resulting from understanding the impact of relaxing constraints may be many times the cost of the additional trailers.

One other problem became apparent as the software was implemented. We became aware that an increasing portion of the backhaul capacity was being used to return overstocked items in stores to the distribution center to permit redistribution of the goods. It soon became clear that the opportunity cost of this policy in terms of handling costs and lost backhaul opportunities far outweighed the profit that would be lost from reducing prices enough to clear excess stock. The amount of redistribution has been drastically reduced and the number of backhauls from suppliers has been increased in response to these cost tradeoffs.

Quality Stores is now beginning to use the software to aid in planning of delivery schedules to stores and backhauls from high volume suppliers. The current weekly delivery schedule had been established to balance the workload and for convenience in routing. The software makes routing much easier, permitting management to consider different weekly schedules to accommodate seasonal workload fluctuations and to incorporate backhauls into the weekly schedule.

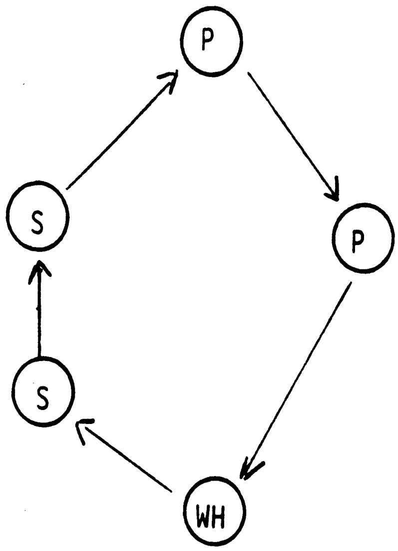
DISCUSSION

The implementation process was not without difficulties. During the development process, shipment volumes increased dramatically and the software had to be modified to accommodate these increased requirements. The transportation department staff also needed to learn about optimization and how imposition of constraints increases transportation costs. Finally, the data collection and input process was a time-consuming task even though much of the data was already available.

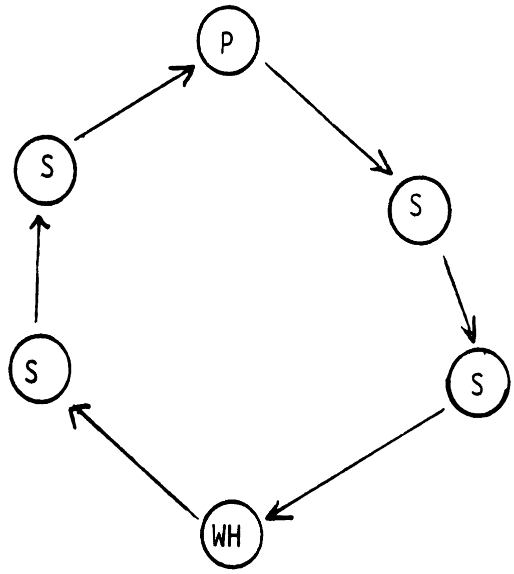
We believe that personal computer implementation was a key factor in the acceptance of this new approach. The transportation department staff did not have much experience with computers, and the personal computer implementation facilitated the development of very user-friendly software which was fully under the control of the user. As mentioned earlier, a clerk with no prior computer experience could use the software with very little training.

For this implementation, it was also important that the procedure provide optimal solutions. The transportation director had so much experience in selecting routes that a good heuristic may not have provided much, if any, improvement. Furthermore, a heuristic procedure would not have permitted an investigation of the effects of various constraints on total system cost.

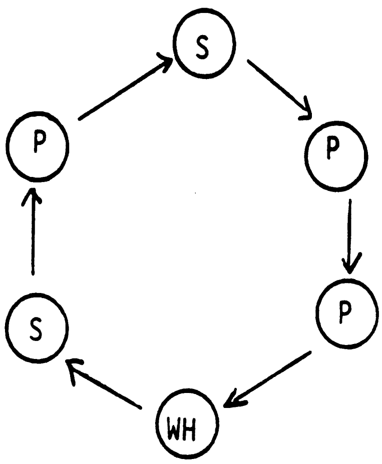
The vehicle routing problem at Quality Stores is not unique. Many companies make deliveries and backhauls using small, private fleets, with routes which are generated manually. Examples include traditional retail stores as well as fast-food chains and dry-cleaning establishments. The general approach presented here can be used to help many of these companies to reduce transportation costs, and thereby increase profits commensurately. Many of these businesses have relatively slim profit margins (after accounting for all variable costs), so even a few percentage point reduction in transportation costs can have a big impact on the bottom line.



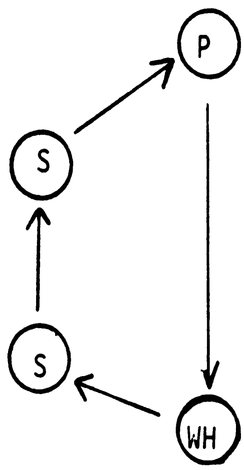
(a) Two deliveries followed by two pickups destined for the warehouse.



(b) Two deliveries followed by a pickup which must be distributed to two stores and the warehouse.



(c) One delivery followed by a pickup which must be delivered to a store, followed by two pickups to be delivered to the warehouse.



(d) Two store deliveries followed by a single pickup.

LEGEND

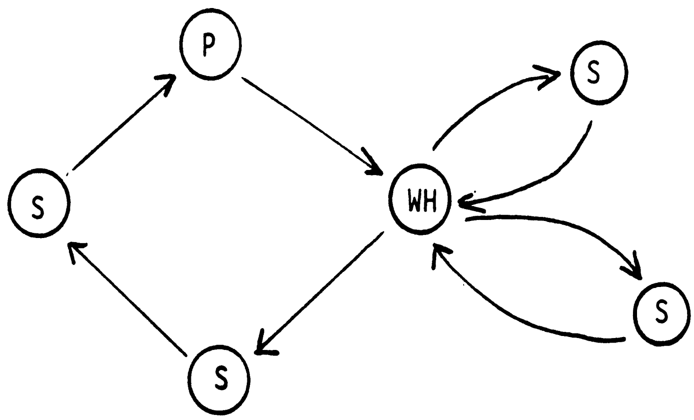
WH: warehouse

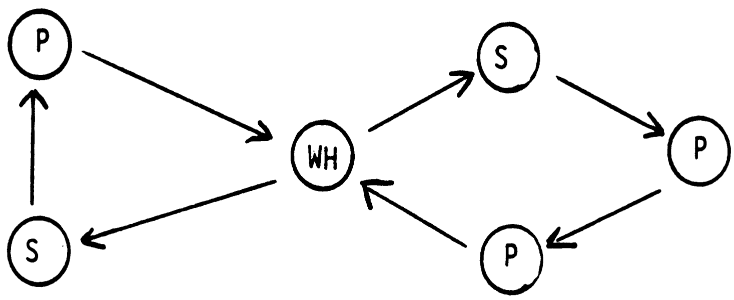
S: store

P: pickup location

FIGURE 1

EXAMPLES OF CYCLES





LEGEND

WH: warehouse

S: store

P: pickup location

FIGURE 2

EXAMPLES OF A CLOVERLEAF AND

A FIGURE EIGHT ROUTE

REFERENCES

Belardo, S., P. Duchessi, and J.P. Seagle (1985), "Microcomputer Graphics in Support of Vehicle Routing," Interfaces 15(6), 84-92.

Bell, W.J., L.M. Dalberto, M.L. Fisher, A.J. Greenfield, R. Jaikumar, P. Kedia, R.G. Mack, P.J. Prutzman (1983), "Improving the Distribution of Industrial Gases with an On-line Computerized Routing and Scheduling Optimizer," Interfaces 13(6), 4-23.

Bodin, L., B. Golden, A. Assad, and M. Ball, (1983), "Routing and Scheduling of Vehicles and Crews - The State of the Art," Computers and OR 10(2), 69-211.

Cullen, F., J. Jarvis, and H.D. Ratliff (1981), "Set Partitioning Based Heuristics for Interactive Routing," Networks 11(1), 125-143.

Etcheberry, J. (1977), "The Set-Covering Problem: A New Implicit Enumeration Algorithm," Operations Research 25(5), 760-772.

Murty, K.G. (1973), "On the Set Representation and Set Covering Problems", in S.F. Elmaghraby (editor), Proceedings of the Symposium on the Theory of Scheduling and its Applications, Heidelberg: Springer-Verlag, pp. 143-163.

Murty, K.G., (1976), Linear and Combinatorial Programming, New York: John Wiley and Sons (now available from Krieger Publishing Co., Melbourne, FL), pp. 458-466.