CHAPTER 30

Industrial Engineering Applications in Transportation

CHRYSSI MALANDRAKI
DAVID ZARET
JUAN R. PEREZ
CHUCK HOLLAND
United Parcel Service

Handbook of Industrial Engineering: Technology and Operations Management, Third Edition.
Edited by Gavriel Salvendy Copyright © 2001 John Wiley & Sons, Inc.

1. OVERVIEW

Transportation and distribution play a critical role in the successful planning and implementation of today's supply chains. Although many view the transportation of goods as a non-value-added activity, effective transportation planning and execution will not only enhance a company's productivity but will also increase customer satisfaction and quality.

In this chapter, we will explore the factors that impact the transportation of goods and the different tools and techniques that the industrial engineer can apply in the development of effective transportation networks and systems to reduce or minimize costs, improve cycle time, and reduce service failures. A similar but inherently different aspect of transportation is that of transporting people. Although this chapter is concerned with the transportation of goods, the industrial engineer also plays an important role in designing these types of systems.

Today's logistics activities are concerned with the movement of goods, funds, and information. Information technology has now become an integral component of any transportation system. Technology is being used for scheduling and creating complex delivery and pickup routes and also for providing customers with up-to-the-minute information on the status of their shipments. The industrial engineer will not only aid in the development of efficient delivery routes and schedules, but will also help in the design of state-of-the-art transportation information systems.

2. INTRODUCTION

Transport is the process of transferring or conveying something from one place to another. Transportation is the process of transporting. The transportation of people, goods, funds, and information plays a key role in any economy, and the industrial engineer can play a key role in properly balancing the parameters and constraints that affect the effectiveness and efficiency of transportation systems.

This chapter will cover certain applications of industrial engineering in transportation. The emphasis is placed on the movement of goods. However, the methodologies described in the chapter can be applied to a variety of transportation problems.

Transportation plays a critical role in today's increasingly small world. Economies, businesses, and personal travel are, in one word, global. Information on the status of the items or persons being moved is as crucial as the movement itself. Confirmation of delivery in a timely, electronic form is often as important as on-time, damage-free, value-priced arrival.

The industrial engineer can apply a variety of mathematical and engineering tools and techniques in the planning and management of effective transportation networks and systems in order to reduce or minimize costs, improve cycle time, reduce service failures, and so on. The industrial engineer plays a critical role in the development of efficient delivery routes, schedules, and plans and also helps in the design and implementation of transportation information systems.

3. TRANSPORTATION AND INDUSTRIAL ENGINEERING

3.1 Transportation as a System

Designing a transportation system means, in most cases, the design of multiple integrated systems—systems to move goods or people, systems to move information about goods or people, and systems to move funds associated with goods or people. Within each of these three types of systems there may be multiple subsystems.

Today, in the package delivery business, transportation companies offer several types of services in which the primary distinction is the time it takes to move the package from its point of origin to its final destination. Services range from same-day or next-day delivery to multi-week delivery, utilizing different transportation modes such as airplanes, trucks, railcars, boats, and even bicycles. The packages may all originate at the same location (e.g., shipper) going to the same destination, or each package in a shipment may have a different destination. In either case, the transportation systems must be capable of supporting a variety of service offerings, depending on the customer's needs. The success of such systems is measured by the systems' effectiveness in meeting the promised service guarantees.

In air travel, the service offerings vary from first class to coach travel. Dividing the aircraft into separate travel compartments allows multiple systems to utilize the same asset, route, flight crew, and schedule to offer variations in service. Therefore, the air travel industry must also utilize multiple integrated systems when designing the processes to facilitate the movement of people.

Providing information about the goods or people being moved also involves the implementation and use of multiple integrated systems. Many customers today provide information about shipments to the carrier at the time of shipment or earlier. This information is introduced into multiple systems for a variety of uses. The shipper has information about the shipment: its contents, the carrier, the mode of transportation, the expected date of arrival, the value of the shipment, shipping charges, and so on. The carrier and shipper can, through integrated systems, track the status of the shipment as it moves to its final destination. Upon delivery, the carrier can provide proof of delivery proactively or upon request. The receiver can be prealerted of the upcoming shipment, its contents, and the expected date and time of arrival. For international shipments, information can be sent to customs before the item is moved to the customs site. The movement and timely availability of all this information increase the efficiency of the modern supply chain.

As with the movement of goods, people, or information, the transfer of funds makes use of integrated systems. The electronic billing of the carrier's charges to the shipper is integrated with the system providing information about the shipment itself. The payment and transfer of funds to the carrier can be initiated by the pickup or the delivery or through periodic billing cycles. Payments upon delivery for the contents of the shipment can be handled electronically and triggered by the electronic transmission of the signature at delivery.

In designing a transportation system, it is crucial to facilitate the integration of the multiple systems needed for the movement of the goods, people, information, and funds. Industrial engineers are often the catalysts in facilitating such integration.

4. THE PARAMETERS AND FUNCTIONS ASSOCIATED WITH TRANSPORTATION

There are two basic parameters that affect the design of freight transportation processes: the territory to be covered and the frequency with which the transportation events occur. These parameters are not static. Demand (e.g., number of shipments per week or day) fluctuations make these parameters dynamic in nature. Demand in transportation is often seasonal. The Thanksgiving holiday in the United States is one of the highest demand days for air travel. The Saturday prior to Mother's Day usually presents a substantial increase in package delivery volume compared to other Saturdays. Demand can vary by time of day or day of the week. Traffic volume in a major city varies greatly between Monday morning at 7:30 am and Thursday morning at 3:00 am. The transportation system must be designed to handle these variations in demand effectively.

It is important to note that what drives demand in the private freight transportation sector is cost and quality of service. The design (planning), execution, and measurement of the transportation system have an impact on the cost and quality of the transportation process. The better the costs and quality of service, the greater the demand and therefore, the greater the need to alter the service territory and the frequency of service.

Transportation planning, execution, and measurement are the fundamental functions associated with transportation and are integral to the success of the transportation system.

Planning is needed across all areas of the transportation system. Planning the overall distribution network, regional (e.g., certain geography) planning, and site (e.g., local) planning are crucial. Asset planning, including buildings and facilities, vehicles, aircraft, trailers, and materials-handling equipment, is also required. Demand variations drive decisions for asset quantity and capacity. The use of owned vehicles supplemented by leased vehicles during peak (high-demand) times is an example of the decisions that need to be made during the planning process. The impact of demand on scheduling, vehicle routing, and dispatching drives labor and asset needs. Labor can often be the largest component of cost in the transportation industry. The transportation planning activity is charged with developing plans that, while minimizing costs, meet all service requirements.

The execution of the plan is as important as the planning of the transportation process itself. An excellent plan properly executed results in lower costs and higher quality of service. This in turn drives demand and therefore the need for new plans.

Finally, the proper measurement of the effectiveness of the transportation system, in real time, offers a mechanism by which transportation managers, supply chain specialists, and industrial engineers can constantly reduce costs and improve service. With new technologies such as wireless communication, global positioning systems (GPS), and performance-monitoring technology, measurement systems allow the users to improve transportation processes as needed.

5. THE ROLE OF THE INDUSTRIAL ENGINEER IN TRANSPORTATION PLANNING AND TRANSPORTATION OPERATIONS

The role of the industrial engineer in the transportation industry is primarily to aid the organization in providing a high level of service at a competitive price. The industrial engineer has the skills necessary to assist in many areas that impact the effectiveness of the transportation system:

- *Work measurement and methods analysis:* Labor is a large component of the total cost of service in the transportation industry. The pilot, driver, captain, or engineer literally controls the ‘‘container’’ of goods or people in transit. The design of effective work methods and the development of the appropriate work measurement offer tools that aid management in the control and execution of transportation processes and provide a mechanism by which the performance of the system can be measured. Methods design and work measurement are often used to develop comprehensive work scheduling and vehicle routing systems aimed at reducing costs and meeting all service commitments.
- *Facility design and location:* The determination of the number of facilities required to move materials and finished goods from one point to another, their capacity, and their location is often considered one of the traditional roles of the industrial engineer. The use of single (e.g., decentralized) vs. regional distribution center locations and the determination of the territory served by a local terminal are part of the transportation system design process. In addition, industrial engineers will often aid in the decision process to determine whether a facility should be automated or be built using manual sorting systems.
- *System design:* The integration of the components of the transportation process into a highly efficient system also involves the industrial engineer.
- *Equipment:* Requirements, design and selection of trucks, trailers, containers, aircraft, scanners, communication devices, materials-handling systems, etc. are tasks undertaken by the industrial engineer.
- *Facility layout:* The industrial engineer is often responsible for the design of facility layouts that will offer the most effective arrangement of the physical components and materials-handling equipment inside distribution centers and delivery terminals.
- *Asset utilization and control:* The design of systems and procedures to balance and manage the number of trucks, trailers, containers, aircraft, scanners, communication devices, and materials-handling systems required to facilitate the transportation processes is one more area requiring the attention of the industrial engineer.
- *Measurement systems:* Industrial engineers also participate in the development of effective transportation performance measures, including customer satisfaction, cost, and plan accuracy.

As indicated by this list, the industrial engineer adds great value to the transportation industry. Industrial engineering principles are highly applicable in this complex industry. Industrial engineers continue to serve as key members of the teams responsible for the design and integration of systems to facilitate the movement of goods, people, funds, and information in this increasingly competitive industry.

6. TRANSPORTATION AND THE SUPPLY CHAIN

Supply chain management is a comprehensive concept capturing objectives of functional integration and strategic deployment as a single managerial process. Figure 1 depicts the supply chain structure.

This structure has been in place for decades. Even when the manufacturing activity takes place in minutes, the final delivery of a product may take days, weeks, or months, depending on the efficiency of the supply chain. The operating objectives of a supply chain are to maximize response, minimize variance, minimize inventory, maximize consolidation, maintain high levels of quality, and provide life-cycle support.

Transportation is part of the physical distribution. The physical distribution components include transportation, warehousing, order processing, facility structure, and inventory. The major change in

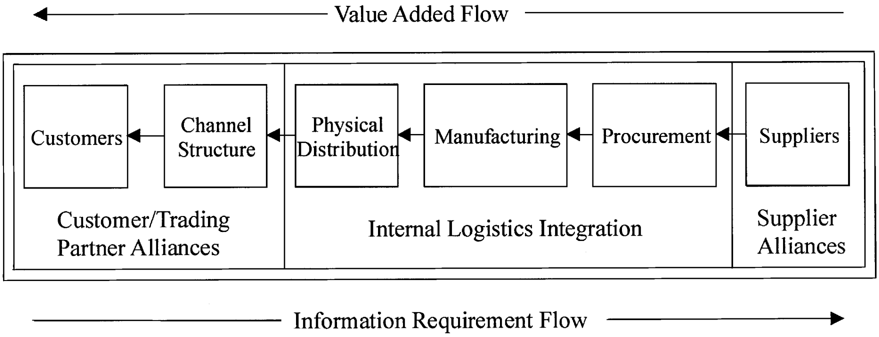

Figure 1 The Supply Chain Structure.

the supply chain in the last decade is information. It is information that allows transportation planners to reduce the costs in the supply chain in today's highly competitive environment. The integrated freight transportation systems that facilitate the movement of goods, information, and funds can target all areas of the supply chain. Every segment of the supply chain has a transportation need.

In an organization, transportation requirements may cover a wide range of territory and frequency characteristics. Decisions are usually made on the basis of cost as long as customer requirements are met. When making decisions that affect the organization's supply chain, it is important not to look at transportation alone or as an independent activity. Instead, transportation should be viewed in the context of the entire supply chain in order to make the most effective decisions (Figure 2).

The remainder of this chapter examines industrial engineering applications in transportation. We concentrate our attention in the transportation of goods. However, the techniques reviewed in the following sections can be applied to a variety of transportation problems. We do not provide a complete survey; instead, we present several representative applications in some detail.

7. TRANSPORTING GOODS

7.1. Cost, Time, and Quality Optimization in Transportation

In the transportation of goods, we often develop models that minimize total cost while maintaining acceptable levels of service. This cost may be a combination of the cost of operating a vehicle (fuel, maintenance, depreciation, etc.), the labor cost of the operator of the vehicle (driver, pilot, etc.), and possibly other fixed costs. In the transportation of small packages or less-than-truckload shipments, sorting cost is also considered, representing the cost of sorting and handling the packages to consolidate shipments. Cost may be considered indirectly, as in the case of minimizing empty vehicles or maximizing the load factor.

The time needed for transporting goods is another very important factor. Instead of minimizing general cost, transportation time may be minimized directly when time constraints are very tight or when the computation of cost is dominated by the labor cost. Whether the transportation time is

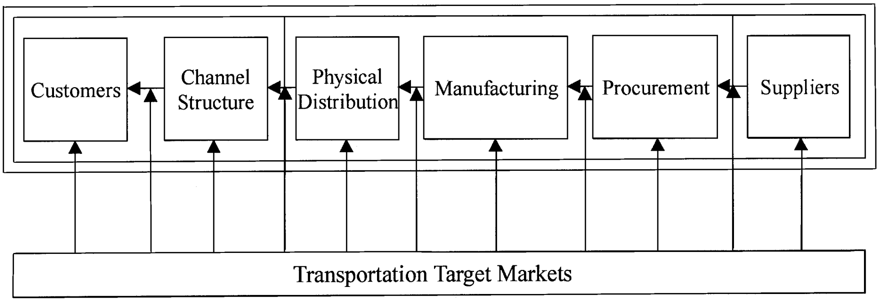

Figure 2 Transportation Target Markets in the Supply Chain.

minimized directly or not, the model may include time constraints (e.g., time windows, total time of a driver's route) or the time element may be incorporated in the input data of the model (e.g., included in the underlying network).

There is a tradeoff between transportation cost and transportation time. In the last decades, shorter product cycles in manufacturing and notions like just-in-time inventory have resulted in an increasing demand for smaller transportation times at higher prices and higher transportation cost. The small-package transportation industry has responded with many "premium" services that guarantee short transportation times. Even when the transportation time is not guaranteed, it represents a primary element of quality of service along with other considerations such as minimization of lost and damaged goods, which are often handled indirectly or are external to the cost optimization models.

7.2. Integrating Customer Needs in Transportation Planning

Although we often try to minimize cost in transportation planning, what we really want to achieve is maximization of profit (revenue minus cost). Cost minimization assumes that demand is external to the model and unaffected by the solution obtained. Demand remains at assumed levels only if the obtained solution satisfies customer requirements and customer needs are integrated into the transportation planning process.

Excluding price and transportation time, many customer needs are not easily incorporated into a mathematical model. They may be included in the input data (e.g., different types of service offered) or considered when alternative solutions obtained by optimization are evaluated. Flexibility, a good set of transportation options, and good communication are of primary importance to customers, permitting them to effectively plan their own operations, pickups, and deliveries.

7.3. Forecasting in Transportation Planning

The development of effective transportation plans is highly dependent on our ability to forecast demand. Demand, in the case of the transportation of goods, refers to the expected number of shipments, the number of packages associated with a shipment, and the frequency with which such shipments occur. When developing transportation routes and driver schedules, demand levels are used as input and therefore need to be forecasted. Changes in demand can occur randomly or can follow seasonal patterns. In either case, if an accurate forecast is not produced, the transportation planning effort will yield less accurate results. These results have implications in the design of facilities (e.g., capacity), the acquisition of assets (e.g., delivery vehicles), and in the development of staffing plans (e.g., labor requirements). It is important to note that several factors affect demand in the transportation industry. Business cycles, business models, economic growth, the performance of the shipper's business, competition, advertising, sales, quality, cost, and reputation all have a direct impact on the demand for transportation services.

When developing a forecast, the planner must address some basic questions:

1. Does a relationship exist between the past and the future?
2. What will the forecast be used for?
3. What system is the forecast going to be applied to?
4. What is the size of the problem being addressed?
5. What are the units of measure?
6. Is the forecast for short-range, long-range, or medium-range planning purposes?

Once these questions have been addressed, the steps shown in Figure 3 guide the planner towards the development of a forecasting model that can be used in generating the forecast to support the transportation planning process.

Depending on the type of the transportation problem being solved (e.g., local pickup and delivery operations, large-scale network planning), the user may select different forecasting techniques and planning horizons. For long-range network planning in which the planner is determining the location of future distribution centers, long-range forecasts are required. Sales studies, demographic changes, and economic forecasts aid in the development of such forecasts. When developing aggregate plans in order to determine future staffing needs and potential facility expansion requirements, the planner develops medium-range forecasts covering planning horizons that range from one or two quarters to a year. Known techniques such as time series analysis and regression are usually applied to historical demand information to develop the forecast. Finally, in order to develop weekly and daily schedules and routes to satisfy demand in local pickup and delivery operations, the planner may develop daily, weekly, or monthly forecasts (short-range forecasts). Because demand fluctuations can have a direct impact on the effectiveness of pickup and delivery routes, the use of up-to-date information from shippers is critical. Techniques such as exponential smoothing and trend extrapolation facilitate this type of forecasting.

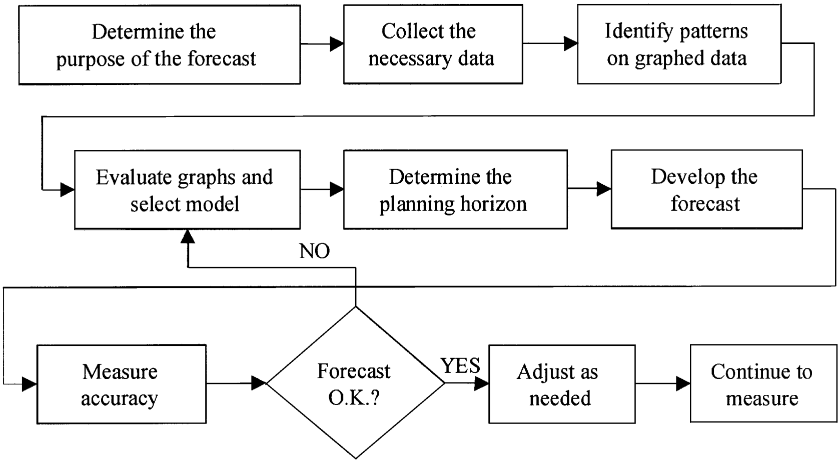
Determine the purpose of the forecast
Collect the necessary data
Identify patterns on graphed data
Evaluate graphs and select model
Determine the planning horizon
Develop the forecast
NO
Measure accuracy
Forecast O.K.?
YES
Adjust as needed
Continue to measure

Figure 3 Steps in Forecasting.

Forecasting is often considered an art. Because the inputs to any forecasting model are mostly historical and based on experience, the accuracy of such forecasts is based on the model selected and the frequency with which it is updated. Forecasting models are usually grouped into two categories: subjective and objective.

Subjective forecasting models are based on judgement and experience. Several techniques exist that attempt to make use of the "expert" 's knowledge to develop a forecast. These techniques include the Delphi method, jury of executives, and the use of sales force intelligence to develop the forecast.

Objective forecasting models are also known as quantitative models. The selection of a quantitative model is dependent on the pattern to be projected and the problem being addressed. There are two types of quantitative forecasting models: time series and explanatory models. In time series models, time is the independent variable and past relationships between time and demand are used to estimate what the demand will be in the future. Explanatory models, on the other hand, use other independent variables instead of or in addition to time. The variables used in the forecasting model are those that have shown a consistent relationship with demand.

When evaluating the impact of seasonal variability on the forecast, the planner has several indicators that can be used to refine the forecast. Leading and lagging economic indicators of the general business cycles can aid the forecaster in the refinement of plans. The Department of Commerce creates an index of well-known economic indicators. This index can be effectively applied to medium- and long-range forecasts to anticipate demand in the transportation of goods.

Several correlation coefficients can be calculated to determine how closely the forecasts correlate with actual demand. The sample mean forecast error (e.g., the square root of the sum of the squared forecast errors), which provides an approximation of the average forecast error of the forecasting model, can also be used. Choosing the best forecast technique requires an understanding of the particular forecasting problem and the characteristics of the available data. Ultimately, the planner's ability to use past and current information to forecast stops, delivery volume, and other key inputs to the transportation planning process will determine the quality and accuracy of the transportation plans developed. With changes in technology and with the increased availability and sharing of information between companies, we expect substantial improvements in the way forecasting is done in the transportation industry.

8. PICKUP AND DELIVERY

8.1. Pickup-and-Delivery Operations

Daily operations in parcel delivery include local pickup and delivery. Terminal facilities are assigned to service areas so that every package that has an origin or destination in the service area is handled through the assigned terminal. Each workday, a fleet of vehicles leaves a terminal (depot) carrying packages to be delivered to customers and returns carrying packages that have been picked up from customers. Most deliveries occur before the pickups, which take place late in the day. Each customer stop (either delivery or pickup) is characterized by a time window during which service must begin.

These windows are usually one-sided: a delivery must occur before a given time and a pickup after a given time. Some windows, though, may be wide open (e.g., a delivery must occur by the end of the workday) and other windows may be two-sided (a pickup must occur before an early closing time). Vehicle capacities are not a limitation in this problem; therefore, the fleet of vehicles may be considered homogenous if the difference in type of vehicle does not have a significant effect on cost. Maximum on-road time for the workday is an important limitation, where on-road time is defined as the total elapsed time from the moment a vehicle leaves the depot until it returns to the depot. The importance of on-road time derives from the fact that often drivers are paid overtime for hours of work beyond a certain length (e.g., 8 hours), and work rules prohibit work days beyond a certain length (e.g., 10 hours).

Efficient distribution implies good asset utilization. The primary assets in this problem are vehicles and drivers. Minimization of the number of drivers and of the total cost of the routes are frequently used objectives. Efficiency needs to be achieved while level of service is maintained; that is, service is provided to all customers in such a way as to satisfy all time-window constraints. This problem has a very short-term planning horizon. Since customer stops may differ from day to day and all the stops are known only a few hours before the actual pickup-and-delivery operation, the problem needs to be solved a few hours before implementation.

8.2. Modeling

The pickup-and-delivery problem can be modeled as a variation of the vehicle routing problem with time windows (VRPTW) and a single depot. Inputs for the VRPTW include matrices that specify the distance and travel time between every pair of customers (including the depot); service time and time window for each customer; maximum (or specified) number of drivers; starting time of the workday; and maximum on-road time of the workday. The maximum on-road time of the workday can be implemented as a time window on the depot, so that the pickup and delivery problem described above can be considered as a traveling salesman problem with time windows and multiple routes (m-TSPTW). However, the term *VRPTW* will be used in this section for this uncapacitated pickup-and-delivery problem.

Distances and travel times can be derived from the longitude and latitude of the customers and depot, assuming specific speed functions. Alternatively, distances and travel times can be obtained from an actual street network; this latter method provides greater accuracy. Longitude and latitude values are relatively easy to obtain; it is often considerably more difficult to acquire data for a street network.

The objective in the VRPTW is to minimize the number of drivers and/or minimize the total cost of the routes while satisfying all constraints. Cost is a function of total distance and on-road time. The output is a set of routes, each of which specifies a sequence of customers and starts and ends at the depot. Because of the short-term planning horizon, the drivers need to be notified for work before the problem is solved. In many cases, therefore, the number of drivers may be estimated and assumed given; the objective then becomes to minimize total cost.

The number of drivers is unlikely to be changed daily. However, the transportation of small packages is characterized by seasonality of demand. In the United States, demand increases by 40–50% during the months before Christmas. A pickup-and-delivery model can be used to obtain the minimum number of additional drivers that must be used to maintain level of service when demand increases.

The VRPTW is an extension of the traveling salesman problem with time windows (TSPTW), which in turn is an extension of the classical traveling salesman problem (TSP). The TSP and its variants have been studied extensively, and many algorithms have been developed based on different methodologies (Lawler et al. 1985). The TSP is NP-complete (Garey and Johnson 1979) and therefore is presumably intractable. For this reason, it is prohibitively expensive to solve large instances of the TSP optimally. Because the TSPTW (and the VRPTW with a maximum number of available drivers) are extensions of the TSP, these problems are NP-complete as well. In fact, for the TSPTW and VRPTW, not only is the problem of finding an optimal solution NP-complete; so is the problem of even finding a feasible solution (a solution that satisfies all time window constraints) (Savelsbergh 1985). For more on time constrained routing and scheduling problems, see Desrosiers et al. (1995).

Existing exact algorithms for the VRPTW have been reported to solve problems with up to 100 stops. However, the problems encountered in parcel delivery are often substantially larger than this. Moreover, these problems need to be solved quickly because of the short-term planning horizon. For these reasons, much of the work in this area has focused on the development of heuristic algorithms—algorithms that attempt to find good solutions instead of optimal solutions.

Heuristic algorithms for the TSPTW and VRPTW are divided into two general categories: route-construction heuristics and route-improvement heuristics. The first type of heuristic constructs a set of routes for a given set of customers. The second type of heuristic starts from an existing feasible solution (set of routes), and attempts to improve this solution. Composite procedures employ both types of heuristic, either by first constructing routes heuristically and then improving them or by applying route-improvement procedures to partially constructed routes at selected intervals during the route-construction process itself. Route-construction and route-improvement heuristics are discussed in more detail below.

8.3. Heuristic Construction Algorithms

There are two general strategies that a route-construction algorithm for the VRPTW can adopt. The first strategy is ''cluster first, route second,'' which first assigns stops to drivers, and then constructs a sequence for the stops assigned to each driver. The second strategy carries out the clustering and sequencing in parallel.

Because clustering is based primarily on purely spatial criteria, the cluster-first, route-second strategy is often more appropriate for the vehicle routing problem (VRP), where time windows are not present than for the VRPTW. However, cluster-first strategies can play a useful role in some instances of the VRPTW, as will be discussed later.

For now, we will confine our attention to algorithms that carry out clustering and sequencing in parallel. Within this general class of algorithms, there is a further distinction between sequential heuristics, which construct one route at a time until all customers are routed, and parallel heuristics, which construct multiple routes simultaneously. In each case, one proceeds by inserting one stop at a time into the emerging route or routes; the choice of which stop to insert and where to insert it are based on heuristic cost measures. Recent work suggests that parallel heuristics are more successful (Potvin and Rousseau 1993), in large part because they are less myopic than sequential approaches in deciding customer-routing assignments.

In the following discussion, we focus on heuristic insertion algorithms for the VRPTW. Solomon (1987) was the first to generalize a number of VRP route-construction heuristics to the VRPTW, and the heuristic insertion framework he developed has been adopted by a number of other researchers. Solomon himself used this framework as the basis for a sequential algorithm. In what follows, we briefly describe Solomon's sequential algorithm and then turn to extensions of the generic framework to parallel algorithms.

8.4. A Sequential Route-Construction Heuristic

This heuristic builds routes one at a time. As presented by Solomon (1987), sequential route construction proceeds as follows:

1. Select a ''seed'' for a new route.
2. If not all stops have been routed, select an unrouted stop and a position on the current route that have the best insertion cost. If a feasible insertion exists, make the insertion, else go to step 1.

In step 1, the heuristic selects the first stop of a new route (seed). There are various strategies for selecting a seed. One approach is to select the stop that is farthest from the depot; another is to select the stop that is most urgent in the sense that its time window has the earliest upper bound. In step 2, the heuristic determines the unrouted stop to be inserted next and the position at which it is to be inserted on the partially constructed route. In order to make this determination, it is necessary to compute, for each unrouted stop, the cost of every possible insertion point on the route.

Solomon introduced the following framework for defining insertion cost metrics. Let $(s_0, s_1, \ldots, s_m)$ be the current partial route, with s_0 and s_m representing the depot. For an unrouted stop u, $c_1(s_i, u, s_j)$ represents the cost of inserting u between consecutive stops s_i and s_j. If the insertion is not feasible, the cost is infinite. For a given unrouted stop u, the best insertion point $(i(u), j(u))$ is the one for which

$$c_1(i(u), u, j(u)) = \min_{p=1,\ldots,m} [c_1(s_{p-1}, u, s_p)]$$

The next stop to be inserted into the route is the one for which

$$c_1(i(u^*), u^*, j(u^*)) = \min_u [c_1(i(u), u, j(u))]$$

Stop u^* is then inserted in the route between $i(u^*)$ and $j(u^*)$. Possible definitions for the cost function c_1 are considered later.

8.5. A Parallel Route-Construction Heuristic

A parallel route-construction heuristic that extends Solomon's sequential algorithm is presented next; the presentation is based on ideas presented by Potvin and Rousseau (1993, 1995) and Russell (1995).

1. Run the sequential algorithm to obtain an estimate of the number of routes k. We can also retain the k seeds obtained by the sequential algorithm. Alternatively, once one has obtained an estimate of the number of routes, one can use some other heuristic to generate the seeds. A representative method is the seed-point-generation procedure of Fisher and Jaikumar (1981). The result of this process is a set of k routes that start from the depot, visit their respective seeds, and return to the depot.
2. If there are no unrouted stops, the procedure terminates. Otherwise, for each unrouted stop u and each partially constructed route $r = (s_0, \ldots, s_m)$, find the optimal insertion point $(p_r(u), q_r(u))$ for which

$$c_1(p_r(u), u, q_r(u)) = \min_{j=1,\ldots,m} [c_1(s_{j-1}, u, s_j)]$$

for some cost function c_1. For each unrouted u, its optimal insertion point $(p(u), q(u))$ is taken to be the best of its route-specific insertion points:

$$c_1(p(u), u, q(u)) = \min_r [c_1(p_r(u), u, q_r(u))]$$

Select for actual insertion the node u^* for which

$$c_2(p(u^*), u^*, q(u^*)) = \text{optimal}_u [c_2(p(u), u, q(u)]$$

for some cost function c_2 that may (but need not) be identical to c_1. If a feasible insertion exists, insert the stop u^* at its optimal insertion point and repeat step 2. Otherwise, go to step 3.
3. Increase the number of routes by one and go to step 2.

Possible definitions for c_1 and c_2 are presented below; again, the presentation here follows that of Potvin and Rousseau (1993).

We can measure the desirability of inserting u between i and j as a weighted combination of the increase in travel time and cost that would result from this insertion. Thus, let

d_{ij} = cost from stop i to stop j
$d(i,u,j) = d_{iu} + d_{uj} - d_{ij}$
e_j = current service start time at j
$e_{u,j}$ = new service start time at j, given that u is now on the route
$t(i,u,j) = e_{u,j} - e_j$

Then a possible measure of the cost associated with the proposed insertion is given by

$$c_1(i,u,j) = \alpha_1 d(i,u,j) + \alpha_2 t(i,u,j) \tag{1}$$

where α_1 and α_2 are constants satisfying

$$\alpha_1 + \alpha_2 = 1$$
$$\alpha_1, \alpha_2 \geq 0$$

One way of defining the measure c_2 is simply by setting $c_2 = c_1$. In this case, the optimal c_2-value is a minimum. An alternative approach is to define c_2 as a "maximum regret" measure.

A regret measure is a kind of "look-ahead" heuristic: it helps select the next move in a search procedure by heuristically quantifying the negative consequences that would ensue, if a given move were not selected. In the context of vehicle routing, a simple regret measure for selecting the next customer to be routed might proceed as follows. For each unrouted customer, the "regret" is the difference between the cost of the best feasible insertion point and the cost of the second-best insertion point; the next customer to be added to a route is one whose regret measure is maximal. But this is still a relatively shortsighted approach. One obtains better results by summing the differences between the best alternative and *all* other alternatives (Potvin and Rousseau 1993; Kontoravdis and Bard 1995). The regret measure that results from this idea has the form

$$c_2(u) = \Sigma_{r \neq r^*} [c_1(p_r(u), u, q_r(u)) - c_1(p_{r^*}(u), u, q_{r^*}(u))] \tag{2}$$

Underlying the use of this regret measure is a notion of urgency: the regret measure for a stop u is likely to be high if u has relatively few feasible or low-cost insertion points available. We would like

TABLE 1 Travel Times between Each Pair of Nodes Including the Depot (node 0)

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
0	0	48	29	36	44	54	44	49	51	45	43	42	39	21	43	16	29	46	65
1	56	0	38	43	49	89	59	16	18	54	29	61	64	53	25	56	62	40	72
2	39	40	0	37	45	76	55	41	43	48	31	54	64	38	32	39	46	42	63
3	44	43	35	0	62	86	33	44	49	27	46	34	54	53	33	51	60	24	77
4	52	49	43	62	0	67	63	53	45	74	37	62	67	42	52	47	42	64	58
5	62	89	74	86	67	0	93	92	89	92	79	89	73	73	87	73	64	92	51
6	52	59	53	33	63	93	0	55	64	21	62	20	52	65	51	61	74	38	89
7	57	16	39	44	53	92	55	0	24	50	30	58	63	55	23	57	64	36	75
8	59	18	41	49	45	89	64	24	0	56	28	66	67	53	29	57	62	44	70
9	53	54	46	27	74	92	21	50	56	0	55	28	53	61	43	58	70	28	85
10	51	29	29	46	37	79	62	30	28	55	0	63	61	43	29	48	53	44	62
11	50	61	52	34	62	89	20	58	66	28	63	0	44	61	53	57	70	42	85
12	47	64	62	54	67	73	52	63	67	53	61	44	0	61	72	54	64	64	79
13	29	53	36	53	42	73	65	55	53	61	43	61	61	0	52	19	24	56	59
14	51	25	30	33	52	87	51	23	29	43	29	53	72	52	0	53	58	29	72
15	24	56	37	51	47	73	61	57	57	58	48	57	54	19	53	0	28	55	63
16	37	62	44	60	42	64	74	64	62	70	53	70	64	24	58	28	0	66	54
17	54	40	40	24	64	92	38	36	44	28	44	42	64	56	29	55	66	0	81
18	73	72	61	77	58	51	89	75	70	85	62	85	79	59	72	63	54	81	0

to route such a *u* relatively early to prevent its few good insertion points from being taken by other customers.

8.6. A Numerical Example

We illustrate the route-construction process by considering an instance of the VRPTW with 18 stops, together with the depot (node 0). Table 1 displays the time matrix for this problem, and Table 2 the time windows. All times are listed in hundredths of an hour; for example, 9:45 is represented as 975. The starting time from the depot is 875 for all drivers. Three routes are to be generated, using as seeds nodes 18, 6, and 4.

The parallel route-construction procedure initializes each route by inserting its seed node. Hence, this process creates the three one-stop routes, (0–18–0), (0–6–0), and (0–4–0). Next, for each un-

TABLE 2 Time Windows

Node	Earliest	Latest
0	875	1300
1	875	1200
2	875	1050
3	875	1200
4	875	1300
5	875	1300
6	875	1200
7	875	1050
8	875	1200
9	875	1200
10	875	1200
11	875	1050
12	875	1200
13	875	1200
14	875	1200
15	875	1200
16	875	1050
17	875	1200
18	875	1300

routed node, the best feasible insertion cost for each of the three current routes is computed from formula (1), where $\alpha_1 = 0$, $\alpha_2 = 1$, and time is used as the cost measure. This computation obtains for node 1 the best insertion costs (55 63 53) for the three routes. Hence, the best feasible insertion of node 1 is into route 3 ($r^* = 3$ in formula (2)). The regret measure for node 1 using formula (2) is calculated as follows:

$$c_2\,(1) = 55 - 53 + 63 - 53 + 53 - 53 = 12$$

The regret measures for the remaining unrouted nodes are computed similarly. The regret values for each node are shown in the following array, where *x* indicates a node that has already been included in a route.

$$\text{Regrets} = (12\ 16\ 52\ x\ 100\ x\ 3\ 23\ 96\ 29\ 86\ 21\ 31\ 1\ 24\ 50\ 48\ x)$$

At each iteration of the procedure, the next stop to be inserted into a route is a stop with the maximal regret measure. In the present case, node 5 has the maximal regret. Hence, the procedure inserts node 5 into its best feasible insertion point (first stop after the depot on route 1). After node 5 is inserted, the three routes under construction become (0–5–18–0), (0–6–0), and (0–4–0). The algorithm repeats this process until all nodes have been inserted.

The three routes returned by the procedure are shown in Table 3 and Figure 4.

8.7. Infeasible Problems

One of the inputs to a typical instance of the VRPTW is the maximum number of drivers available. In the case where the time windows are relatively tight, an algorithm may not be able to find a feasible solution (i.e., a solution in which all time windows are satisfied). The way one proceeds in this case depends on the nature of the application. In some applications, where there is a service guarantee and the same penalty applies no matter how close to meeting the service commitment a late delivery happens to be, we wish to minimize the number of missed time windows. In other applications, where the penalty is proportional to the lateness of a delivery, we wish to minimize,

TABLE 3 Solution Routes

Node	Time Window		Arrival Time
Route 1			
0	875	1300	875
5	875	1300	929
18	875	1300	980
16	875	1050	1034
13	875	1200	1058
15	875	1200	1077
0	875	1300	1101
Route 2			
0	875	1300	875
12	875	1200	914
11	875	1050	958
6	875	1200	978
9	875	1200	999
17	875	1200	1027
3	875	1200	1051
0	875	1300	1095
Route 3			
0	875	1300	875
2	875	1050	904
14	875	1200	936
7	875	1050	959
1	875	1200	975
8	875	1200	993
10	875	1200	1021
4	875	1300	1058
0	875	1300	1110

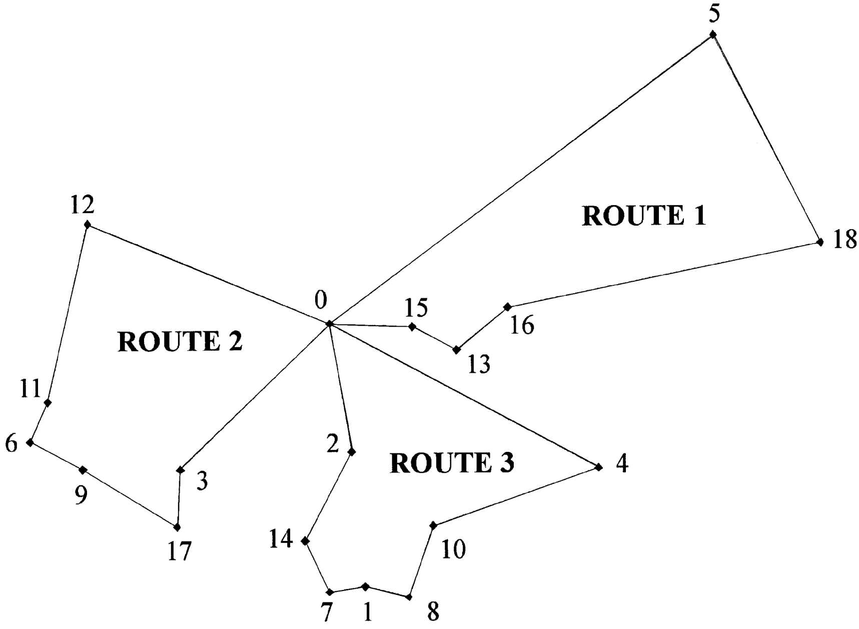

Figure 4 Example Solution.

over all stops, the total time by which the upper bounds of the time windows of stops are missed. In either case, we can still apply the heuristic insertion framework described above by adding an appropriate penalty function to the standard heuristic cost functions.

8.8. Work Breaks

The algorithms presented thus far provide a framework for solving a variety of routing and scheduling problems. However, the specific algorithms we have considered are applicable directly only to a small range of problems; they generally have to be extended in some way in order to deal with real-world applications. One important extension arises in routing applications in which the driver spends an entire day on the road. For such applications, we must take into account the necessity of scheduling lunch and other breaks in order to satisfy work rules and other constraints. Taking these factors into account complicates an already difficult problem.

We can characterize a break by specifying the following values: earliest start time for the break, latest start time for the break, and duration of the break. For example, we might specify that we are to include a 50-minute lunch break that begins any time between 11 am and 1 pm.

To accommodate breaks within the heuristic insertion framework, the most natural way to proceed is to represent the breaks themselves as nodes. Note that a break does come with a natural time window (earliest and latest start times) and an analogue of service time (duration of break). The one crucial missing item is location. In some cases, the location of breaks may be specified as part of the input to the algorithm. In general, however, the algorithm will have to select a location for each break, based on the locations of the nodes that represent actual stops.

For example, suppose we wish to schedule a break after stop a but before stop c. According to the heuristic insertion framework, this is just the problem of inserting the break node b between nodes a and c. One possible approach is simply to assign a's location to b. In this case, we calculate the arrival time at c as follows. Let $[e_x, l_x]$ be the time window for a given stop x, svcTime_x the service time at x, and t_{xy} the travel time from x to y. If x has been assigned to a route, we let arrival_x be the arrival time of the driver at location x. Then the time at which service begins at stop x is given by

$$\text{startTime}_x = \max(\text{arrival}_x, e_x)$$

In the case being considered here, where break node b is inserted between stops a and c, we have

$$\text{startTime}_c = \text{startTime}_a + \text{svcTime}_a + \text{svcTime}_b + t_{ac}$$

Recall that svcTime_b is just the duration of the break.

This is the basic picture, but there is at least one complication that we have to take into account. Suppose, as before, that we wish to insert break node *b* between *a* and *c*. Suppose also that the travel time from *a* to *c* is 30 minutes, that service is completed at *a* at 10:45, and that the time window for *b* is [11:00, 1:00]. Then, according to the simple scheme just described, we would have to wait for 15 minutes at location *a* before starting lunch at *a*!

To avoid this sort of awkward behavior, we would actually insert *b* at the first point on segment (*a*, *c*) such that there is no waiting time at *b*. If there is no such point, then we conclude that no lunch break can be inserted between stops *a* and *c*. For example, in the scenario just described, we would insert *b* halfway between *a* and *c*.

When a break has been fully represented this way as a node with time window, location, and service time, it is treated just like any other node within the heuristic insertion framework.

8.9. Route-Improvement Heuristics

Heuristic route-improvement procedures play a fundamental role in vehicle routing algorithms. Such procedures take as input a feasible solution consisting of a route or set of routes and seek to transform this initial solution into a lower-cost feasible solution.

One of the most successful route-improvement strategies has involved the use of edge-exchange heuristics. The edge-exchange technique was introduced by Lin and Kernighan (1973) as a local search strategy in the context of the conventional traveling salesman problem, but it has since been applied to a variety of related problems, including the vehicle routing problem.

For an individual route, a *k*-exchange involves replacing *k* edges currently on the route by *k* other edges. For example, a 2-exchange involves replacing two edges (say $(i, i + 1)$ and $(j, j + 1)$) by two other edges ((i, j) and $(i + 1, j + 1)$). Usually, all the available *k*-exchanges are examined and the best one is implemented. This is repeated as long as an improved solution is obtained. Since there are $\binom{n}{k}$ subsets of *k* edges in a cycle of *n* edges, the computational complexity of the edge-exchange method increases rapidly with *k*. Even one *k*-exchange requires $O(n^k)$ time, so attention is usually confined to the cases $k = 2$ and $k = 3$.

The idea of an edge exchange can be extended in a straightforward way to the case of pairs of routes. In this case, one exchanges entire paths between routes, where a path consists of a sequence of one or more nodes. For example, let $\text{routeA} = (a_1, a_2, \ldots, a_k)$, $\text{routeB} = (b_1, b_2, \ldots, b_n)$. Then, after exchanging paths between the two routes, the routes that result would be of the form, $\text{routeA} = (a_1, \ldots, a_i, b_{j+1}, \ldots, b_{j+r}, a_m, \ldots, a_k)$, and similarly for routeB.

As mentioned, the goal of a route-improvement heuristic is to reduce routing costs; typically this means that we wish to minimize route duration. However, when we are dealing with time windows, we must also verify that any proposed exchange retains time window feasibility. Techniques for efficient incorporation of time window constraints into edge-exchange improvement methods were developed by Savelsbergh (1985, 1990, 1992); see also Solomon et al. (1988).

The most recent work on route improvement has focused on the development of metaheuristics, which are heuristic strategies for guiding the behavior of more conventional search techniques, with a view towards avoiding local optima and thereby achieving higher-quality solutions. Metaheuristic techniques include tabu search, genetic algorithms, simulated annealing, and greedy randomized adaptive search (Glover 1989, 1990; Kontoravdis and Bard 1995; Potvin et al. 1996; Rochat and Taillard 1995; Taillard et al. 1997; Thangiah et al. 1995).

To illustrate the pertinent concepts, we focus here on tabu search. Tabu search helps overcome the problem of local optimality, which often arises when traditional deterministic optimization heuristics are applied. The problem of local optimality arises in the following way. A wide range of optimization techniques consists of a sequence of moves that lead from one trial solution to another. For example, in a vehicle-routing algorithm, a trial solution consists of a route for each vehicle; and a "move," in the route-improvement phase, might consist of some sort of interroute exchange. A deterministic algorithm of this general type selects a move that will most improve the current solution. Thus, such a procedure "climbs a hill" through the space of solutions until it cannot find a move that will improve the current solution any further. Unfortunately, while a solution discovered with this sort of hill-climbing approach cannot be improved through any local move, it may not represent a global optimum.

Tabu search provides a technique for exploring the solution space beyond points where traditional approaches become trapped at a local optimum. Tabu search does not supplant these traditional approaches. Instead, it is designed as a higher-level strategy that guides their application.

In its most basic form, the tabu method involves classifying certain moves as forbidden (or "tabu"); in particular, a move to any of the most recently generated solutions is classified as tabu.

(When we speak of "moves" here, we mean moves generated by some traditional search method such as edge exchange.) The tabu algorithm then selects the best move from the set of moves not classified as tabu, in order to drive the search into new regions. No concern is given to the fact that the best available moves may not improve the current solution. In particular, because a recently achieved local optimum will be classified as tabu, the method can drive the search down the hill away from this local optimum. The hope is that the expanded search will lead to an improved final solution. There is no guarantee that an improved solution will be found, but variations on the method just described have achieved considerable success and have become increasingly popular.

What has been described here is a very simple tabu search framework, which performs "bookkeeping" operations to ensure that the algorithm does not return to recently explored solutions. More sophisticated metaheuristic strategies more explicitly guide the direction of the search itself.

8.10. Preassigned Routes and Preassigned Territories

The heuristic route-construction methods described thus far are designed to minimize route cost while satisfying time window and other constraints. Route cost is typically a function of total on-road time and distance. In an approach to routing that seeks only to minimize travel time and distance, one typically follows a reoptimization strategy. That is, one is faced each day by a new problem, defined by the set of customers who actually require service that day; and one solves that problem by applying a heuristic route-construction algorithm. Because there is no concept, in such algorithms, of geographical area or regularity of service, the solution for one day will be independent of the solution for another. While these solutions may be very efficient from the standpoint of total cost and travel time, they may not be satisfactory for some applications. The reason for this shortcoming is that in some applications, there may be additional, hidden costs that the reoptimization strategy fails to take into account.

One such cost involves driver knowledge of the area in which he or she is providing service. A driver who is familiar with the road network and traffic conditions in his service area is likely to be more efficient than a driver for whom the delivery area is relatively new. A second, related cost involves the development of business relationships. For many service providers, an important goal is to achieve regularity and personalization of service by having the same driver visit a particular customer every time that customer requires service.

These considerations suggest that for some applications, it may be desirable to maintain some consistency from one day to the next in assigning customers to drivers. In this case, we need to devise some alternative to the reoptimization strategy.

One way of devising such an alternative is to interpret our problem as a probabilistic vehicle routing problem (PVRP) (Jaillet and Odoni 1988). According to this interpretation, we are given a service region and the set of all potential customers in that region. On a given day, only a subset of the customers will actually need service and the exact customer set for a given day cannot be predicted. However, we are also given a probability distribution, based on historical data, over the set of potential customers. The probability assigned to a potential customer represents the probability that that customer will require service on any given day.

One way of defining an objective function for this problem is provided by the a priori optimization strategy investigated by Jaillet (1988) and Bertsimas et al. (1990). In order to develop the ideas underlying this strategy, it is helpful to focus on the TSP. In the conventional TSP, we are given a complete graph $G = (V,E)$ and wish to find a lowest-cost tour that visits every node in V. We obtain the probabilistic TSP (PTSP) by supposing that, on any given day, the traveling salesman may have to visit only some subset S of the nodes in V. The probability that any v in V is present (must be visited) in a given problem instance is given by a probability function p over V. We identify a problem instance with the subset S of nodes that are present in that instance. Thus there are 2^n possible instances of the PTSP on G, where $n = |V|$.

According to the a priori optimization strategy, one finds a priori a tour through all n nodes of G. For any given instance of the problem, the $k \leq n$ nodes that are actually present are visited in the same order as they appear in the a priori tour; the $(n - k)$ missing nodes are simply skipped.

A natural measure of effectiveness for the a priori strategy is given by the notion of minimum length in the expected value sense. Thus, let τ be the a priori tour, and let L_τ be the length of τ. If the problem instance S occurs with probability $p(S)$ and requires covering a total length $L_\tau(S)$ according to the a priori tour, then it will receive a weight of $p(S)L_\tau(S)$ in the computation of expected length. Therefore, according to this construal of the PTSP, the objective is to find an a priori tour τ_0 through the n nodes of G, which minimizes the quantity

$$E[L_\tau] = \Sigma_{S \subseteq V}\, p(S)L_\tau(S)$$

We extend this model to the VRP by requiring that an a priori tour be constructed for each driver in such a way that all potential customers are assigned to some driver. Note that a set of a priori tours of this sort does provide the kind of regularity of service described above.

At first glance, the task of calculating $E[L_\tau]$ for a given a priori tour τ may appear problematic because the summation is over all 2^n subsets of V. However, Jaillet derived an efficient closed-form expression for $E[L_\tau]$, which requires only $O(n^2)$ time to compute (see discussion in Jaillet 1988).

Of course, being able to calculate $E[L_\tau]$ efficiently for a given a priori tour is very different from actually finding a tour that minimizes $E[L_\tau]$. Because the PTSP is at least as hard as the TSP, there is little hope of developing exact optimization methods that could solve more than modestly sized instances of the problem. Consequently, one must employ heuristics in order to develop practically viable PTSP algorithms. However, it has proven difficult to develop effective heuristic approaches to the PTSP, at least in part because the class of optimal solutions to the PTSP has properties very different from those associated with the conventional (Euclidean) TSP. For example, one of the fundamental properties of the Euclidean TSP is that an optimal solution cannot intersect itself; this property follows directly from the triangle inequality. In contrast, an optimal solution for the PTSP *can* intersect itself, even when the triangle inequality is satisfied (systematic treatments of the differences between the TSP and PTSP are presented by Jaillet [1988] and Jaillet and Odoni [1988]). One consequence of the fact that the PTSP has features very different from the TSP is that, in general, we cannot expect heuristic approaches designed specifically for the TSP to be extended successfully to the PTSP. In general, therefore, entirely new solution approaches are needed. One of the few promising approaches to the PTSP that has been discussed in the literature is based on the idea of "spacefilling curves" (Bartholdi and Platzman 1988). But the probabilistic versions of the TSP and VRP remain very much under the heading of research topics.

The problem becomes even more difficult when we turn to probabilistic versions of the VRPTW. For this problem, presumably, the a priori strategy would involve multiple objectives; in addition to minimizing the expected length of the a priori tour, we would also want to minimize the expected number of missed time windows. But it is difficult even to formulate this problem properly, much less solve it.

For this reason, when time windows are involved, we try in actual applications to capture some subset of the key features of the problem. In this case, instead of trying to construct a priori tours of the kind just described, we simply try to construct a solution in which a large percentage of repeat customers is handled by the same driver and each driver is relatively familiar with most of the territory to which he or she is assigned.

One way of achieving such a solution is to partition the service area into geographical regions and then always assign the same driver to the same region. In operational terms, a solution of this sort is one in which drivers have well-defined territories: a driver travels from the depot to his or her service area, carries out his or her work there, and then returns to the depot. This picture is to be contrasted with the one that emerges from the heuristic insertion framework, which usually results in routes that form a petal structure: in general, it creates routes in the form of intersecting loops, in which stops appear in a roughly uniform way.

To construct a solution in which drivers are assigned to territories in this way, it is appropriate to adopt the cluster-first, route-second strategy mentioned earlier. Thus, we proceed by first assigning stops to drivers, thereby partitioning the service area. We then construct a sequence for the stops assigned to each driver. This strategy represents a departure from the heuristic insertion framework, according to which clustering and routing proceed in parallel.

One way of implementing a cluster-first strategy is as follows. Suppose that, on a given day, we wish to construct k routes. We can proceed by (heuristically) solving a k-median problem (Daskin 1995); we then carry out the initial clustering by assigning each stop to its closest median. Having created the clusters, we then construct a route for each cluster individually by solving the resulting TSPTW. Unfortunately, it is often impossible to construct a feasible route for each cluster. When feasible routes cannot be constructed, we have to shift stops between clusters in such a way as to achieve feasibility while maintaining well-defined territories as much as possible.

If we begin by solving a new k-median problem each day, then we may succeed in producing well-defined territories for drivers; but we are unlikely to achieve consistency from one day to the next in assigning customers to drivers. A natural way of extending this approach to achieve consistency is to take historical information into account. Thus, we can solve a weighted k-median problem over the set of all locations that have required service over a specified number of past days, where the weight of a location is proportional to the number of times it has actually required service. We would then consistently use these medians as the basis for clustering. There may well be days in which the initial clusters thus constructed are significantly unbalanced with respect to numbers of customers per driver; in this case, we have to shift stops between clusters, as described above.

Whether the procedure just outlined can be applied successfully in the presence of time windows depends in part on the nature of those time windows. For some applications, time windows are one-sided, and there are only a few different service levels: for example, delivery by 9:00 am for express packages, delivery by 11:00 am for priority packages, and so on. For such applications, it is plausible to suppose that the sort of strategy just described, which emphasizes the spatial aspects of the problem, can be implemented successfully. If, on the other hand, there is a variety of different, overlapping, two-sided time windows, then we are faced with what is really a three-dimensional problem, with time as the third dimension. In such a case, where the temporal aspect of the problem predominates, the prospects for a successful cluster-first strategy of the type described here are considerably less promising.

An alternative heuristic strategy is to employ a sweep method for clustering, introduced by Gillet and Miller (1974); see also Fisher and Jaikumar (1981) and Hall et al. (1994). According to this strategy, one begins by dividing the service area into annuli centered at the depot, using some heuristic method for determining the radius of each annulus. For each annulus, one then sorts the set of customers by increasing polar angle and then builds routes sequentially, inserting stops in order into the current route.

As in the *k*-median method, the prospects for successful application, in the presence of time windows, of what is essentially a spatial method depend on the nature of those time windows. Again as in the *k*-median method, one would have to take historical data into account in order to achieve assignments of customers to drivers that are reasonably consistent over time.

8.11. Implementation Issues

It is evident, on the basis of the discussion in this section, that for vehicle-routing applications one may require a suite of algorithms, each of which best serves a somewhat different objective. Furthermore, to make the best use of these algorithms, it is important that the actual user be specially trained for the application.

Accuracy of the input data is very important in this problem (and in optimization problems in general). Because the objective is to minimize cost, the model is very sensitive to data that affect cost calculations. Because traffic conditions change with the time of day, weather conditions, and so on, travel times are generally estimated even when the best data are available. When longitude and latitude values are used as the basis for travel time estimates, a further loss of accuracy results. This indicates that the extra cost of getting provably optimal solutions may not be justified, and also that effort needs to be made to develop methodologies that are robust with respect to data inaccuracies. One must certainly take into account the possibility of such inaccuracies when one reports the output routes. It is especially important to do so if the intention is to provide precise directions to drivers who are inexperienced in their assigned areas. Obvious inaccuracies may generate distrust for the model by the users.

The pickup-and-delivery problem discussed in this section is static in the sense that all demand is known before any service begins. But one can also consider real time dispatching problems, in which demand is generated in real time and routing decisions are made after the driver has left the depot. Advances in communications technology make it feasible to implement real-time systems of this sort. They also make feasible the implementation of hybrid systems, in which drivers are given predetermined routes but these routes may be modified dynamically as new demand arises during the workday. Dynamic problems of this sort will require new algorithmic techniques to supplement those presented in this section.

9. LARGE-SCALE TRANSPORTATION NETWORK PLANNING

9.1. Line-Haul Movement of Packages

During the pickup and delivery operation of a parcel delivery company, packages are picked up and brought to a local terminal, as discussed in Section 8. These packages are sometimes transported to their destination terminal directly and delivered to the customers. In most cases, however, packages go though a series of transfer terminals where they get sorted and consolidated. Transportation between terminals is achieved using several modes of transportation. The most prevalent modes on the ground are highway and rail, using a fleet of tractors and trailers of different types. A fleet of aircraft is used for transporting by air.

A transfer terminal often shares the same building with one or more local terminals. Several sorting operations may be run daily in a transfer terminal and are repeated every day of the work week at the same times. A large transfer terminal may have four sorting operations at different times of the day or night. Packages get unloaded, sorted, and reloaded on trailers. Several types of trailers may be used with different capacities and other characteristics. One or more trailers are attached to a tractor and depart for the next terminal on their route. This may be another transfer terminal or their destination terminal, or a rail yard for the trailers to be loaded and transported by rail, or an air terminal, if the packages have to be transported by air. In this section, we will examine a model that determines the routes of packages from their origin to their destination terminals and the equipment used to transport them, only for packages that travel on the ground using two modes, highway or rail.

Package handling and transporting between terminals represents a large part of the operating costs for a package transportation company. A good operating plan that minimizes cost and/or improves service is crucial for efficient operations. In recent years, many new products and services have been introduced in response to customer demand. These changes as well as the changes in package volume require additions and modifications to the underlying transportation network to maintain its efficiency. A planning system that can evaluate alternatives in terms of their handling and transporting costs and their impact on current operations is very important. There is a tradeoff between the number of sorting operations a package goes through and the time needed to reach its destination. More sorting operations increase the handling cost, decrease the transportation cost because they consolidate loads better, but also increase the total time needed until a package is delivered to the customer. All these factors need to be taken into account in the design of a transportation network for routing packages.

9.2. A Network-Optimization Problem

The network-optimization problem for transporting packages on the ground can be defined as follows. Determine the routes of packages, mode of transportation (highway or rail), and types of equipment to minimize handling and transportation costs while the following constraints are met: service requirements are respected, the capacities of sorting operations are not surpassed, no sorting operation is underutilized, the number of doors of each facility available for loading trailers is considered, trailers balance by building, and all the packages to the same destination are routed along the same path.

To maintain level of service, packages must be transported between particular origin/destination (OD) pairs in a given time. The capacity of a sorting operation cannot be surpassed but also, if the number of packages through a sorting operation falls below a given value, the sorting operation needs to be closed down. The number of loading doors of each facility represents the maximum number of trailers that can be loaded simultaneously during a sorting operation. Note that two or more of the trailers loaded simultaneously may be going to the same terminal next on their route.

Trailers need to balance by building daily. This means that the trailers of each type that go into a building must also leave the building during the daily cycle. Balancing is achieved by introducing empty trailers into the system. There are several types of trailers that are used, with different capacities. Some types can be used both on highway and rail. Trailers can also be rented from the railroads, and these trailers may not need to balance. Each tractor can pull one, two, and, on rare occasions, three trailers, depending on the trailer types and highway restrictions. At a terminal, tractor–trailer combinations are broken up, the trailers unloaded, the packages sorted, the trailers loaded again, and the tractor and trailers reassembled to form new combinations. There is no constraint on the number of trailers of any type that are used.

An important operational constraint is that all the packages to the same destination must be routed along the same path. This implies that all packages with the same destination terminal follow the same path from the terminal where they first meet to their destination. This constraint results from the fact that sorting is done by destination. If sorting became totally automated, this constraint might be modified.

A network-optimization model can be used as a long-term planning tool. It can evaluate alternatives for locating new building facilities, opening or closing sorting operations, and changing an operating plan when new products are introduced or the number of packages in the system changes. Because any such changes require retraining of the people who manually sort the packages, the routing network in parcel delivery is not changed often or extensively. For a network-optimization model to be used as an intermediate-term planning tool to modify the actual routing network, the model needs to obtain incremental changes, that is, to obtain solutions as close as possible to the current operations.

9.3. Modeling

The network-optimization problem is formulated on a network consisting of nodes that represent sorting operations and directed links that represent movement of packages from one sorting operation to another. Both the highway and rail modes are represented by one link if the same time is needed to traverse the link by either mode. If a different time is needed by each mode, two different links are used, each representing one mode. In the latter case, all the packages need to travel by only one mode even if both modes are feasible so that all the packages that start together arrive at the same time to their destination. It is assumed that every day is repeated unchanged without any distortion of the pattern because of weekends. Since a sorting operation is repeated every day at the same time, one node represents a sorting operation for as many days as necessary for a complete cycle of operations to occur in the system. To avoid any confusion, we will use in the description of the network design problem the terms *origin* and *destination* or *OD pair* to refer to the origin and destination nodes of a package; we will use the terms *initial* and *final* nodes to refer to the origin and destination nodes of a directed link.

Each node is characterized by a daily starting and ending time of the sorting operation, a sorting capacity, a sorting cost, and the number of loading doors available. Each link is characterized by a travel time and distance between its initial and final nodes and the types of combinations that are permitted on the link with their costs and capacities. The combinations representing both highway and rail are associated with the same link when the travel time is the same for highway and rail. If the travel time is different and the link represents a particular mode, only the combinations of this mode are associated with the link. In addition, for each OD pair, the number of packages that need to be transported is given as well as the maximum time permitted from the moment a package starts at its origin until it reaches its destination.

A network-optimization system that solves the problem presented above utilizes large amounts of data and must have a reliable data interface in addition to the solver. The data interface extracts, verifies, and validates the data needed for the problem corresponding to the particular geographical area applicable to the problem. It then generates the network and, after the solver is applied, produces reports of the results. The solver consists of optimization algorithms that obtain solutions of the problem.

The network described above can be very large. To facilitate the application of the optimization algorithms, the network size is reduced using various aggregation rules. OD pairs may be consolidated by building or destination group, and local terminals in the same building may be combined into one node. If all the possible connections between each pair of sorting operations are added, the network becomes very dense. For this reason, only links that are likely to be used in the solution are included (e.g., direct links between sorting operations that are farther apart than a given distance are omitted).

The network design problem is formulated below as a mixed-integer programming problem (MIP) (Nemhauser and Wolsey 1988). A graph $G(N,E)$ is given with N a set of nodes and E a set of directed links. A node represents a sorting operation for each consecutive day of a whole cycle of operations. A directed link (i,j) represents the movement of packages from sorting operation i (initial node of the link) to sorting operation j (final node of the link). Because of the presence of two different links that are used in parallel between the same sorting operations representing two different modes with different travel times, link (i,j) may not be unique. To simplify the presentation, however, we disregard the presence of parallel links. The formulation can be easily extended to include this case because at most one of such parallel links can be used in the solution. Additional notation is introduced next.

9.3.1. Notation

Parameters

A = set of OD pairs = $\{(k,l) \mid k \in N$ is the origin and $l \in N$ is the destination, $k \neq l\}$

M = set of all trailer-combination types (a trailer combination consists of a tractor and one or more trailers)

M_{ij} = set of trailer-combination types of link $(i,j) \in E$; $M_{ij} \subseteq M$

Q = set of trailer types

F = set of trailer types that must balance (rented rail trailers may not need to balance); $F \subseteq Q$

B = $\{B_k \mid B_k$ is a building; $B_k \subseteq N\}$; $j \in B_k$ means that sorting operation $j \in N$ resides in building B_k

c_{ijm} = cost of moving trailer-combination type $m \in M_{ij}$ on link $(i,j) \in E$

d_{ikl} = cost of handling all the packages of OD pair $(k,l) \in A$ through node $i \in N$

k_q = capacity of trailer type $q \in Q$ in number of average-size packages

δ_{qm} = number of single trailers of type $q \in Q$ in trailer-combination type $m \in M$

h_{ij} = starting time of sorting operation j minus ending time of sorting operation i so that the packages sorted at $i \in N$ are transported on $(i,j) \in E$ and sorted next at $j \in N$

r_j = duration of sorting operation j

s_{kl} = starting time at origin $k \in N$ of the packages to be transported to destination $l \in N$

v_{kl} = number of average-size packages of OD pair $(k,l) \in A$

τ_{kl} = total time permitted for the packages of OD pair (k,l) to travel from k to l

u_i = capacity of sorting operation $i \in N$ (maximum number of packages that can go through node i)

g_i = minimum number of packages going through $i \in N$, if sorting operation i is open

f_i = maximum number of outgoing-trailer positions (loading doors) available at node $i \in N$

INF = a very large number

Decision variables

x_{ijm} = number of trailer combinations of type $m \in M_{ij}$ used on link $(i,j) \in E$

w_{ijq} = number of nonempty trailers of type $q \in Q$ on link $(i,j) \in E$

y_{ijkl} = 1 if link $(i,j) \in E$ is used to transport the packages of OD pair $(k,l) \in A$; 0 otherwise

z_i = 1 if sorting operation $i \in N$ is open; 0 otherwise

t_{ikl} = departure time of packages of OD pair $(k,l) \in A$ from node $i \in N$ (end time of sorting operation i).

The cost d_{ikl} is estimated as the average cost of a package through sorting operation $i \in N$ multiplied by the number of packages of OD pair $(k,l) \in A$. The cost c_{ijm} is estimated as a function of the distance of link $(i,j) \in E$, fuel prices, driver wages as well as depreciation and cost of the trailer-combination type $m \in M_{ij}$.

It is assumed that all the packages processed through a sorting operation are available at the beginning and leave at the end of the sorting operation. The time h_{ij} of link $(i,j) \in E$ is the difference between the starting time of the sorting operation j and the ending time of the sorting operation i with the appropriate time difference included for the packages to be transported on link (i,j) and be available at j on time. The time h_{ij} as defined above makes it unnecessary to consider the time windows of the sorting operations explicitly in the following formulation. The time h_{ij} is also equal to the travel time on link (i,j) plus the difference in time between the beginning of the sorting operation at j and the arrival of the packages at j, which corresponds to the wait time until the sorting operation j starts. The capacity of a sorting operation is represented by an upper bound that cannot be exceeded. A lower bound may also be used to force sorting operations to be closed if they are underutilized.

9.4. Network Design Formulation

$$\text{Min } Z(\mathbf{x},\mathbf{w},\mathbf{y},\mathbf{z},\mathbf{t}) = \Sigma_{j\in N} \Sigma_{(k,l)\in \mathrm{A}} d_{jkl} \Sigma_{i\in N} y_{ijkl} + \Sigma_{(i,j)\in E} \Sigma_{m\in M_{ij}} c_{ijm} x_{ijm} \tag{3}$$

subject to

$$\Sigma_{j\in N} y_{ijkl} - \Sigma_{p\in N} y_{pikl} = b \qquad \forall\ (k,l) \in A,\ i \in N \tag{4}$$

where $b = 1$ for $i = k$; $b = -1$ for $i = l$; $b = 0$ otherwise

$$t_{jkl} - t_{ikl} - \mathrm{INF}(y_{ijkl} - 1) \ge h_{ij} + r_j \qquad \forall\ i \in N, j \in N, (k,l) \in A \tag{5}$$

$$t_{kkl} = s_{kl} \qquad \forall\ (k,l) \in A \tag{6}$$

$$t_{lkl} \le \tau_{kl} + s_{kl} + r_l \qquad \forall\ (k,l) \in A \tag{7}$$

$$\Sigma_{i\in N} \Sigma_{(k,l)\in A} v_{kl}\, y_{ijkl} \le u_j \qquad \forall\ j \in N \tag{8}$$

$$\Sigma_{i\in N} \Sigma_{(k,l)\in A} v_{kl}\, y_{ijkl} - g_j z_j \ge 0 \quad \forall\ j \in N \tag{9}$$

$$z_j - y_{ijkl} \ge 0 \qquad \forall\ i \in N, j \in N, (k,l) \in A \tag{10}$$

$$y_{ijkl} + y_{ij'k'l} \le 1 \quad \forall\ (i,j) \in E, (i,j') \in E, (k,l) \in A, (k',l) \in A, k < k', j \ne j' \tag{11}$$

$$\Sigma_{j\in B_k} \Sigma_{i\in N} \Sigma_{m\in M_{ij}} \delta_{qm} x_{ijm} - \Sigma_{j\in B_k} \Sigma_{p\in N} \Sigma_{m\in M_{ij}} \delta_{qm} x_{jpm} = 0 \qquad \forall\ B_k \in B, q \in F \tag{12}$$

$$\Sigma_{q\in Q} k_q w_{ijq} - \Sigma_{(k,l)\in A} v_{kl}\, y_{ijkl} \ge 0 \qquad \forall\ (i,j) \in E \tag{13}$$

$$\Sigma_{j\in N} \Sigma_{q\in Q} w_{ijq} \le f_i \qquad \forall\ i \in N \tag{14}$$

$$\Sigma_{m\in M_{ij}} \delta_{qm} x_{ijm} - w_{ijq} \ge 0 \qquad \forall\ (i,j) \in E,\ q \in Q \tag{15}$$

$$x_{ijm} \ge 0 \text{ and integer} \qquad \forall\ i,j,m \tag{16}$$

$$w_{ijq} \ge 0 \text{ and integer} \qquad \forall\ i,j,q \tag{17}$$

$$y_{ijkl} = 0 \text{ or } 1 \qquad \forall i,j,k,l \tag{18}$$

$$z_i = 0 \text{ or } 1 \qquad \forall\ i \tag{19}$$

$$t_{jkl} \ge 0 \qquad \forall\ j,k,l \tag{20}$$

The objective function (3) minimizes the total cost of handling and transporting the packages of all the OD pairs. Constraints (4) are balancing constraints ensuring that the packages of an OD pair start from their origin, end at their destination, and, if they enter an intermediate node, also exit the node. Each constraint (5) computes the departure time of the packages of OD pair (k,l) from node j, using the departure time of the preceding node i. If link $(i,j) \in E$ is used to transport the packages of OD pair $(k,l) \in A$ ($y_{ijkl} = 1$), the constraint becomes $t_{jkl} - t_{ikl} \ge h_{ij} + r_j$. If link $(i,j) \in E$ is not used ($y_{ijkl} = 0$), the constraint is not binding. The starting time at each origin is set with constraints (6).

Constraints (7) ensure that the service requirements are met, that is, the movement of the packages of OD pair (k,l) from their origin k to their destination l takes no more than τ_{kl} time. These constraints also force constraints (5) to apply with equality if necessary.

Constraints (8) ensure that the capacities of the sorting operations are not violated, and constraints (9) prevent sorting operations that are open from being underutilized. Constraints (10) ensure that packages enter node j only if the sorting operation j is open. That is, if $y_{ijkl} = 1$ for any i, j, k, l then $z_j = 1$ from constraints (10). If $y_{ijkl} = 0$ for all i, j, k, l then $z_j = 0$ from constraints (9) and (10).

Constraints (11) ensure that all the packages going though sorting operation i and having the same destination l use the same network link to the next sorting operation on their path, that is, there are no split paths to the same destination. The only case that is permitted is $k \neq k', j = j'$. The case $k = k', j \neq j'$ is excluded by constraints (4) which ensure a unique path for each OD pair (k,l). The case $k \neq k', j \neq j'$ is excluded by constraints (11). Using $k < k'$ avoids repeating the constraints (11) twice.

Constraints (12) are balancing constraints ensuring that for each building B_k and for each trailer type $q \in F$ that must balance, the same number of trailers that enter building B_k also exit it. To achieve balancing of trailers, empty trailers may be introduced into the system by constraints (12). These constraints are more complicated than they would be if each trailer-combination type rather than each trailer type needed to balance. Constraints (13) are the volume constraints, ensuring that for each link (i,j) trailers with enough capacity are used to transport the number of packages on the link.

Constraints (14) ensure that the number of trailers that are filled during a sorting operation is no larger than the number of available loading doors. It is assumed that at most one trailer is filled from each door during a sorting operation. This is a good approximation although it may be conservative at times because occasionally it is possible to fill a trailer at a door and replace it with a new trailer during the same sorting operation. Constraints (15) ensure that the number of nonempty trailers of type $q \in Q$, on link $(i,j) \in E$ represented by w_{ijq} is no larger than the total number of trailers of type $q \in Q$ (empty and nonempty) forming the trailer combinations of type $m \in M_{ij}$ on link $(i,j) \in E$ represented by x_{ijm}. Note that w_{ijq} can have alternative solution values in the previous formulation apart from being exactly equal to the number of nonempty trailers. The value of w_{ijq} is always at least as large as the number of nonempty trailers from constraint (13). If empty trailers are introduced for balancing so that constraint (15) is not binding, w_{ijq} can have alternative solution values larger than the number of nonempty trailers, depending on how tight the corresponding constraint (14) is. Constraints (16) to (20) are the integrality and nonnegativity constraints.

For a large transportation company, the $G(N,E)$ network may have 40,000 links even when care is taken to keep it as sparse as possible. The number of binary variables y_{ijkl} of the above MIP formulation may be 14,000,000 or more. That is also the number of constraints (5), while the number of constraints (11) is much larger.

Formulation (3)–(20) is too large to be supplied to an MIP solver and solved directly. Instead, a heuristic solution for the network design problem can be obtained by solving sequentially two smaller problems that are embedded in formulation (3)–(20). These are the package-routing problem and the trailer-assignment problem. In the first step, the routes of the packages are obtained assuming a representative trailer type for highway and rail. This step produces the number of packages that are transported from the initial node to the final node of each link of the network. In the second step, given the number of packages and the types of permitted equipment on each link, the actual modes and trailer combinations that balance are obtained. These two problems are presented next.

9.5. Package-Routing Problem

The package-routing problem determines the routes of packages on the $G(N,E)$ network for each OD pair so that the service requirements are met, the capacities of the sorting operations are not exceeded, the sorting operations are not underutilized, the number of loading doors of the buildings is not exceeded, packages to a common destination are routed along the same path, trailers are not underutilized, and total cost is minimized. The package-routing problem does not determine the trailer combinations that are used, nor does it balance trailer types.

In the following, we still present a simplified formulation where we do not indicate mode, although we continue to use one link for both modes if the travel times are the same for both modes on the link and two different links if the travel times are different. We extract from formulation (3)–(20) constraints (4)–(8), (18), and (20), which involve only the y_{ijkl} and t_{jkl} variables, putting aside for the moment the constraints on the number of loading doors, underutilized sorting operations, split paths to common destination, and maximization of trailer utilization. Because of their very large number, constraints (11) are not included.

Because the objective function (3) involves variables other than y_{ijkl} and t_{jkl}, we replace it with a new objective function that uses an estimated cost parameter. First, we select a representative trailer-combination type for each mode and estimate the cost per package, per mode for each link by dividing the combination cost on the link by the combination capacity. For links that share both modes, the smaller cost of the two modes is used. Using the cost per package, the following cost parameter is computed:

d_{ijkl} = cost of transporting all the packages of OD pair $(k,l) \in A$ on link $(i,j) \in E$

Using this new cost, the objective function (3) is replaced by the following approximation:

$$\text{Min } Z(\mathbf{y},\mathbf{t}) = \Sigma_{j \in N} \Sigma_{(k,l) \in A} d_{jkl} \Sigma_{i \in N} y_{ijkl} + \Sigma_{(i,j) \in E} \Sigma_{(k,l) \in A} d_{ijkl}\, y_{ijkl} \tag{21}$$

Objective function (21) can be simplified if we combine the transporting cost of all the packages of OD pair (k,l) along link (i,j) and their handling cost through sorting operation j in the following cost parameter:

c_{ijkl} = cost of transporting all the packages of OD pair $(k,l) \in A$ on link $(i,j) \in E$ and handling them through node $j \in N$

The objective function becomes

$$\text{Min } Z(\mathbf{y},\mathbf{t}) = \Sigma_{(i,j) \in E} \Sigma_{(k,l) \in A} c_{ijkl}\, y_{ijkl} \tag{22}$$

The MIP formulation (22), (4)–(8), (18), and (20) of the package-routing problem is still a very large problem and difficult to solve directly. If the complicating constraints (8) are removed, the problem is solved relatively easily. We take advantage of this characteristic by using a Lagrangian relaxation approach (Ahuja et al. 1993; Fisher 1985; Geoffrion 1974). This is combined with search heuristics that also implement the constraints that are completely omitted in the formulation, which are the constraints on the number of loading doors, the lower bound constraints on the capacity of each sorting operation that is open, the split paths constraints, and an additional lower bound constraint on the number of packages on any link $(i,j) \in E$ that is actually used. The last constraint forces more efficient trailer utilization by ensuring that no link carries too few packages, which would result in the use of trailers with very low load factors in the trailer-assignment step.

The following Lagrangian dual problem is obtained by dualizing constraints (8) using their nonnegative Lagrange multipliers λ_j, $j \in N$.

9.6. Lagrangian Relaxation of the Package-Routing Problem

$$\begin{aligned}\text{Min } Z(\mathbf{y},\mathbf{t},\lambda) &= \Sigma_{(i,j) \in E} \Sigma_{(k,l) \in A} c_{ijkl}\, y_{ijkl} + \Sigma_{j \in N} \lambda_j \left(\Sigma_{i \in N} \Sigma_{(k,l) \in A} v_{kl}\, y_{ijkl} - u_j\right) \\ &= \Sigma_{(i,j) \in E} \Sigma_{(k,l) \in A} (c_{ijkl} + \lambda_j v_{kl})\, y_{ijkl} + \text{CONSTANT} \\ &= \Sigma_{(i,j) \in E} \Sigma_{(k,l) \in A} \bar{c}_{ijkl}\, y_{ijkl} + \text{CONSTANT}\end{aligned} \tag{23}$$

subject to

$$\Sigma_{j \in N} y_{ijkl} - \Sigma_{p \in N} y_{pikl} = b \qquad \forall\ (k,l) \in A,\ i \in N \tag{24}$$

where $b = 1$ for $i = k$; $b = -1$ for $i = l$; $b = 0$ otherwise

$$t_{jkl} - t_{ikl} - \text{INF}(y_{ijkl} - 1) \geq h_{ij} + r_j \qquad \forall\ i \in N, j \in N, (k,l) \in A \tag{25}$$

$$t_{kkl} = s_{kl} \quad \forall\ (k,l) \in A \tag{26}$$

$$t_{lkl} \leq \tau_{kl} + s_{kl} + r_l \quad \forall\ (k,l) \in A \tag{27}$$

$$y_{ijkl} = 0 \text{ or } 1 \qquad \forall\ i,j,k,l \tag{28}$$

$$t_{jkl} \geq 0 \quad \forall\ j,k,l \tag{29}$$

$$\lambda_j \geq 0 \qquad \forall\ j \tag{30}$$

The cost $\bar{c}_{ijkl} = c_{ijkl} + \lambda_j v_{kl}$ is the modified cost of transporting all the packages of OD pair $(k,l) \in$ A on link $(i,j) \in E$ and handling them through node $j \in N$, given the Lagrange multipliers λ_j, $j \in N$.

The Lagrangian dual problem (23)–(30) can be solved relatively easily because it decomposes into $|A|$ constrained shortest-path problems, one for each OD pair. A constrained shortest-path problem finds a path from an origin $k \in N$ to a destination $l \in N$ with the smallest cost that has total time no greater than τ_{kl}. Although the constrained shortest-path problem is NP-complete, it can be used as part of an algorithm for the package-routing problem because there are several algorithms that solve large enough problems in good computer running times (Ahuja et al. 1993; Desrosiers et al. 1995).

A heuristic algorithm for the package-routing problem is presented next, based on the previous analysis. It is applied to the network described before that includes only one link between a pair of sorting operations representing the cheaper mode if the times h_{ij} for both modes are the same, and two links, one for each mode, if the times are different. A more detailed (but still high-level) description of each step follows the algorithm.

9.7. Package-Routing Heuristic Algorithm

1. Find the OD pairs that have a unique feasible path from their origin to their destination using a k-shortest path algorithm. Route all the packages with unique shortest paths along the obtained paths, remove them from the package-routing problem, and update all the parameters.
2. Solve a constrained shortest path for each OD pair using the costs $\bar{c}_{ijkl} = c_{ijkl} + \lambda_j v_{kl}$, where the Lagrange multipliers are set to zero for the first iteration.
3. Reroute packages to (i) eliminate split paths from each sorting operation to a common destination, (ii) eliminate split paths between different modes that have different times on links so that only one mode is used, (iii) enforce the capacity constraint, and (iv) enforce the constraint on the number of loading doors for each sorting operation.
4. If the solution can be improved and the limit on the number of Lagrangian iterations is not reached, compute new costs $\bar{c}_{ijkl}$ using subgradient optimization or some other methodology and go to step 2.
5. Reroute packages from underutilized links to consolidate loads while preserving solution feasibility.
6. Reroute packages to close the underutilized sorting operation that is permitted to be closed and realizes the most savings. If a sorting operation is closed, disable the node representing it and start over from step 2.

In step 1, a k-shortest path algorithm (Minieka 1978) for $k = 2$ is used to obtain all the OD pairs that have unique feasible paths. The packages of these OD pairs are routed along their single paths and removed from the problem, updating all the parameters.

Steps 2–4 implement the Lagrangian relaxation algorithm. In step 2, a constrained shortest-path problem is solved for each OD pair using the costs $\bar{c} = c_{ijkl} + \lambda_j v_{kl}$. For the first iteration, the original link costs are used that result from setting the Lagrange multipliers to zero.

Step 3 takes care of the split-paths, capacity, and loading-door constraints. Each node of the network is examined if it satisfies the split-paths constraints (11). If multiple paths to the same destination exist starting from the node, the best path is selected based on several criteria and all the packages from the node to the common destination are routed along this path. Also, if packages are split between two parallel links of different modes, they are rerouted along only one parallel link. Each node is also examined to find whether it satisfies the capacity constraints (8). For each node that surpasses the permitted capacity, OD pairs that use the node are selected according to several criteria and their packages are removed until the capacities are satisfied. The removed packages are routed again sequentially using the constrained shortest-path algorithm on a network where the nodes that have reached their capacities are disabled. The split-paths constraints and the capacity constraints are examined repeatedly until they are satisfied or the algorithm indicates that it cannot find a feasible solution. Finally, each node is examined if it satisfies constraint (14) on the number of loading doors. If a node is found that violates the door constraint, packages are rerouted to sequentially reduce the number of doors but keep the capacity and split constraints valid.

In step 4, if the solution can be improved and more Lagrangian iterations are permitted, new Lagrange multipliers are computed using subgradient optimization (Ahuja et al. 1993; Crowder 1976) or some other methodology. These are used to obtain new costs $\bar{c}_{ijkl}$ and a new Lagrangian iteration starts at step 2. No more details are given here about the Lagrangian relaxation or the subgradient optimization algorithm. A Lagrangian relaxation approach is used to solve the trailer-assignment problem and a more detailed description of this optimization procedure is given in that section. When the Lagrangian iterations are completed, the solution satisfies the split-paths, capacity, and door constraints. If at any point the heuristic cannot obtain a solution that satisfies a constraint, it stops and indicates that it cannot find a feasible solution.

In step 5, links are found that carry too few packages for a complete load and the packages are rerouted so that the capacity, split-paths, and door constraints continue to be satisfied. These constraints are included to improve the results of the trailer-assignment problem that is solved after the package-routing problem and that represents the true cost of a network design solution.

Finally, in step 6, underutilized sorting operations are examined to implement constraints (9). The packages of underutilized sorting operations are rerouted and the underutilized sorting operation for which the total cost of the solution decreases most is eliminated. The node representing the eliminated sorting operation is disabled and the algorithm starts over from step 2.

The routing decisions obtained by the package-routing heuristic algorithm outlined above are used as input in the trailer-assignment problem that is described next.

9.8. Trailer-Assignment Problem

Given the number of packages on each link obtained by solving the package-routing problem, the trailer-assignment problem determines the number and type of trailer combinations on each link of the network $G(N,E)$ that have enough combined capacity to transport all the packages on the link, balance by trailer type for each building, and have the least cost.

The trailer-assignment problem is described next. To solve the problem more efficiently, the network $G(N,E)$ can be modified into the network $G'(N',E')$ as follows. Each node $i \in N'$ represents a building, that is, all the nodes representing sorting operations in the same building are collapsed into one node. All the links that carry packages in the solution of the package-routing problem are included. Among the links in $G(N,E)$ that do not carry packages, only those that may be used to carry empty combinations for balancing are included so that $E' \subseteq E$. In particular, among parallel links between buildings that do not carry packages, only one is retained in G'. Still, the network G' generally has several parallel links between each pair of nodes, in which case link (i,j) is not unique. To avoid complicating the formulation and because it is easy to extend it to include parallel links, we will not include indexing to indicate parallel links, exactly as we did in formulation (3)–(20).

The trailer-assignment problem is formulated below on the network $G'(N',E')$ using constraints (12), (13), and (16) as well as the objective function (3) with some modifications.

9.9. Trailer-Assignment Formulation

$$\text{Min } Z(\mathbf{x}) = \Sigma_{(i,j)\in E'} \Sigma_{m\in M_{ij}} c_{ijm}\, x_{ijm} \tag{31}$$

subject to

$$\Sigma_{i\in N'} \Sigma_{m\in M_{ij}} \delta_{qm}\, x_{ijm} - \Sigma_{p\in N'} \Sigma_{m\in M_{ij}} \delta_{qm}\, x_{jpm} = 0 \qquad \forall\, j \in N',\, q \in F \tag{32}$$

$$\Sigma_{m\in M_{ij}} k_m\, x_{ijm} \geq \bar{v}_{ij} \qquad \forall\, (i,j) \in E' \tag{33}$$

$$x_{ijm} \geq 0 \text{ and integer} \qquad \forall\, i,j,m \tag{34}$$

Objective function (31) is the same as objective function (3). It does not include the first component because it is a constant since the values of the y_{ijkl} variables are known from the solution of the package-routing problem. Constraints (32) are the same as constraints (12) except that the summation over buildings is now unnecessary because a node represents a building. Constraints (33) are similar to constraints (13) where, as in the objective function, the number of packages, $\bar{v}_{ij}$, for all OD pairs on link (i,j) is now a constant. The variables w_{ijq} are not needed anymore and only the variables x_{ijm} are used, representing combinations with both full and empty trailers. So the trailer capacities k_q for $q \in Q$ are replaced by the trailer-combination capacities k_m for $m \in M_{ij}$.

The integer-programming (IP) problem (31)–(34) is still difficult to solve. If constraints (32) are removed, the problem without balancing is easy to solve because it breaks into many very small problems, one for each link. To take advantage of this characteristic, Lagrangian relaxation (Ahuja et al. 1993; Fisher 1985; Geoffrion 1974) is used to solve problem (31)–(34), dualizing the balancing constraints (32). A Lagrange multiplier λ_{jq}, unrestricted in sign, is used for each balancing constraint and the following Lagrangian dual problem is obtained.

$$\begin{aligned}\text{Min } Z(\mathbf{x},\lambda) &= \Sigma_{(i,j)\in E'} \Sigma_{m\in M_{ij}} c_{ijm}\, x_{ijm} \\ &\quad + \Sigma_{j\in N'} \Sigma_{q\in F} \lambda_{jq} \left(\Sigma_{i\in N'} \Sigma_{m\in M_{ij}} \delta_{qm}\, x_{ijm} - \Sigma_{p\in N'} \Sigma_{m\in Mij} \delta_{qm}\, x_{jpm}\right) \\ &= \Sigma_{(i,j)\in E'} \Sigma_{m\in M_{ij}} \left(c_{ijm} + \Sigma_{q\in F} \delta_{qm} (\lambda_{jq} - \lambda_{iq})\right) x_{ijm} \\ &= \Sigma_{(i,j)\in E'} \Sigma_{m\in M_{ij}} \bar{c}_{ijm}\, x_{ijm} \end{aligned} \tag{35}$$

subject to

$$\Sigma_{m\in M_{ij}} k_m\, x_{ijm} \geq \bar{v}_{ij} \qquad \forall\, (i,j) \in E' \tag{36}$$

$$x_{ijm} \geq 0 \text{ and integer} \qquad \forall\, i,j,m \tag{37}$$

where $\bar{c}_{ijm} = c_{ijm} + \Sigma_{q \in F}\ \delta_{qm}\ (\lambda_{jq} - \lambda_{iq})$ is the modified cost of moving trailer-combination type $m \in M_{ij}$ on link $(i,j) \in E'$ given the vector $\boldsymbol{\lambda}$.

Problem (35)–(37) decomposes into $|E'|$ subproblems, one for each link $(i,j) \in \text{E}'$. Each one of the subproblems is a kind of integer reverse knapsack problem, similar to the integer knapsack problem (Martello and Toth 1990; Nemhauser and Wolsey 1988; Chvátal 1983) and can be solved by similar algorithms. Each subproblem is very small, having $|M_{ij}|$ variables for link $(i,j) \in E'$. The solution obtained by solving the Lagrangian dual (i.e., all the reverse knapsack problems) does not necessarily balance trailer combinations at each node even for optimal $\boldsymbol{\lambda}$ and is not generally feasible for the original problem (31)–(34). A feasible solution is obtained heuristically by solving sequentially one minimum-cost-flow problem (Ahuja et al. 1993) for each trailer type that needs to balance. Balancing is achieved by adding empty trailers or replacing one trailer type with another one of larger capacity that is not yet balanced or does not need to balance.

A Lagrangian relaxation heuristic algorithm that solves the Lagrangian dual problem (35)–(37) is presented next. It uses subgradient optimization to compute the Lagrange multipliers $\boldsymbol{\lambda}$.

9.10. Lagrangian Relaxation Algorithm for the Trailer-Assignment Problem

1. Set the Lagrange multipliers $\boldsymbol{\lambda}$ to zero in the Lagrangian dual problem and initialize $\bar{Z}$ (upper bound, best known feasible solution of the original problem) to a high value.
2. Solve the Lagrangian dual problem with the latest values of $\boldsymbol{\lambda}$ (by solving a set of integer reverse knapsack problems) to obtain the optimal objective function value, $Z^*(\mathbf{x}^*, \boldsymbol{\lambda})$, for the given $\boldsymbol{\lambda}$.
3. Apply a heuristic approach to obtain a feasible solution of the problem and update $\bar{Z}$.
4. If the gap between $\bar{Z}$ and $Z^*(\mathbf{x}^*, \boldsymbol{\lambda})$ is small or some other criterion is satisfied (e.g., a set number of iterations is reached or no more improvement is expected), stop.
5. Compute new values of the Lagrange multipliers $\boldsymbol{\lambda}$ using subgradient optimization and go to step 2.

Step 3 above may be implemented only occasionally instead of at every iteration. Improvement heuristics can also be used to modify the solution in several ways. They can be used to combine single trailers into more efficient trailer combinations. They can also be used to find any cycles of trailer types that are either empty or can be replaced by trailer types that do not need to balance. If a whole cycle of trailers that results in improved cost is modified, balancing of trailers is maintained. Improvement heuristics can also be used to replace whole cycles of trailers of excess capacity with trailers of smaller capacity if the total cost is decreased.

A subgradient optimization algorithm (Ahuja et al. 1993; Crowder 1976) is used in step 5 to compute an improved Lagrange multiplier vector and is described below.

9.11. Subgradient Optimization Algorithm

Given an initial Lagrange multiplier vector $\boldsymbol{\lambda}^0$, the subgradient optimization algorithm generates a sequence of vectors $\boldsymbol{\lambda}^0, \boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \ldots$ If $\boldsymbol{\lambda}^k$ is the Lagrange multiplier already obtained, $\boldsymbol{\lambda}^{k+1}$ is generated by the rule

$$t_k = \frac{\rho_k(\bar{Z} - Z^*(\mathbf{x}^*, \boldsymbol{\lambda}^k))}{\Sigma_j\ \Sigma_q\ (\Sigma_i\ \Sigma_m\ \delta_{qm}\ x^k_{ijm} - \Sigma_p\ \Sigma_m\ \delta_{qm} x^k_{jpm})^2} \tag{38}$$

$$\boldsymbol{\lambda}^{k+1}_{jq} = \lambda^k_{jq} + t_k\ (\Sigma_i\ \Sigma_m\ \delta_{qm}\ x^k_{ijm} - \Sigma_p\ \Sigma_m\ \delta_{qm}\ x^k_{jpm}) \qquad \forall\ j \in N',\ q \in F \tag{39}$$

where t_k is a positive scalar called the step size and ρ_k is a scalar that satisfies the condition $0 < \rho_k \leq 2$. The denominator of equation (38) is the square of the Euclidean norm of the subgradient vector corresponding to the optimal solution vector $\mathbf{x}^k$ of the relaxed problem at step k. Often a good rule for determining the sequence ρ_k is to set $\rho_0 = 2$ initially and then halve ρ_k whenever $Z^*(\mathbf{x}^*, \boldsymbol{\lambda}^k)$ has not increased in a specific number of iterations. The costs are calculated with the new values of $\boldsymbol{\lambda}$. If any negative costs are obtained, the value of ρ is halved and the values of $\boldsymbol{\lambda}$ recomputed until all the costs are nonnegative or ρ is too small to continue iterating.

A feasible solution is obtained in step 3, using a minimum-cost-flow algorithm (Ahuja et al. 1993) sequentially for each trailer type that needs to balance. First, for each building the excess or deficit of trailers of each type that has to balance is computed. Then a minimum-cost-flow algorithm is applied for each trailer type that obtains the optimal movements of trailers from each node of excess to each node of deficit. This may be the movement of a single empty trailer, the movement of an empty trailer that gets attached to an already used trailer to make up a permitted combination, or the movement of an already used trailer replacing another trailer type of smaller or equal capacity that is not yet balanced or does not need to balance.

Lagrangian relaxation is often used within a branch-and-bound procedure. The exact branch-and-bound algorithm has large computational cost; also, the trailer-assignment problem is only part of a heuristic algorithm for the network design problem. For these reasons, the Lagrangian relaxation algorithm is applied only once, at the root of the branch-and-bound tree, to find a heuristic solution to the trailer-assignment problem.

9.12. Extensions of the Network-Design Problem

If the results of the network-design problem are intended not only for long-term planning but to actually modify the routing network, the solution must represent an incremental change from the currently used solution. Otherwise, any savings from an improved solution may be lost in modifying the sorting operations to accommodate the solution changes. To this end, both the package-routing and the trailer-assignment problem can be modified relatively easily to handle presetting some of the variables. For the package-routing problem, this means presetting whole or partial paths of specific OD pairs. A solution is obtained by preprocessing the input data to eliminate or modify OD pairs that are completely or partially preset and updating the input data. Similarly, for the trailer-assignment problem, particular combinations on links may be preset. After appropriate variables are fixed, the Lagrangian relaxation algorithm optimizes the remaining variables.

The network-design problem presented considers only packages moving on the ground and chooses only one transportation mode when travel times differ between sorting operations. The network-design problem can be extended to include all modes of transportation and all types of products with different levels of service requirements. Different modes may have different costs and travel times between sorting operations, permitting parallel use of modes with different travel times along the same routes. This extension complicates considerably an already difficult problem and is not described here any further.

10. DRIVER SCHEDULING

10.1. Tractor-Trailer-Driver Schedules

This section examines one approach for solving the tractor-trailer-driver-scheduling problem for a package-transportation company. Tractor-trailer combinations transport packages between terminals of a package-transportation company, as discussed in Section 9. This involves movement of both equipment and drivers. While balancing is the only constraint for equipment routing, the movement of drivers is more complex. The daily schedule of a tractor-trailer driver starts at his or her base location (domicile) where he or she returns at the end of his workday. A driver schedule consists of one or more legs.

A leg is the smallest piece of work for a driver and consists of driving a tractor-trailer combination from one terminal to another or repositioning trailers inside a terminal. Each leg is characterized by an origin terminal and a destination terminal, which may coincide. There is an earliest availability time at the origin terminal, a latest required arrival time at the destination terminal, and a travel time (or repositioning time) associated with a leg. A tractor-trailer combination that is driven from the origin terminal to the destination terminal of a leg must start no earlier than the earliest availability time at the origin and arrive at the destination no later than the latest required arrival time. At the destination terminal of a leg, a driver may drop his or her current tractor-trailer combination and pick up a new one to continue work on the next leg of his or her daily schedule, take a break, or finish work for the day.

In this section, we assume that the legs are already determined and given. We want to generate driver schedules by combining legs. An acceptable schedule must meet specific work rules, which specify the minimum number of hours that a driver must be paid for a day's work (usually 8 hours) and the maximum length of a workday (usually 10 hours). In addition, a driver must return to his or her domicile every day and a workday must incorporate breaks of specified duration at specified times. For example, a lunch break must last 1 hour and be scheduled between 11:00 am and 2:00 pm.

Another consideration in the generation of driver schedules is the availability of packages for sorting within a terminal. During a sorting operation, loads should arrive at such a rate that the facility is not kept idle. If packages arrive late, the facility will be underutilized during the early stages of sorting while the facility capacity may be surpassed later. For this reason, the duration of each sorting operation is divided into equal time intervals, say, half-hour intervals. Driver schedules need to be generated in such a way that volume availability is satisfied, that is, a minimum number of packages arrives at each sorting facility by the end of each time interval.

The driver-scheduling problem has a short or intermediate planning horizon. It is usually solved regularly once or twice a year and the obtained schedules are bid by the drivers, based on seniority.

The problem may also be solved if changes in volume occur that render the existing schedules inadequate. If decisions about driver schedules are made at the local level and the schedules are bid separately by region, the problem is solved locally in a decentralized fashion.

10.2. Driver-Scheduling Problem

The problem of generating driver schedules can be defined as follows. Given a set of legs with time windows and travel times, generate driver schedules that assign work to drivers so that all the loads are moved within their time windows, the total work assigned to each driver meets given work rules, volume availability meets or exceeds sorting capacities, and the total cost of the schedules is as low as possible.

The problem is defined on an underlying network. The terminals are represented by nodes of the network and each leg is represented by an arc connecting the terminal of origin to the terminal of destination. Time windows on the nodes correspond to the earliest departure and latest arrival times at the terminals.

The cost of a schedule is a combination of both time and distance in addition to fixed costs because it is computed based on driver wages, cost of fuel, and vehicle depreciation. Feasible schedules can be generated using deadheading—that is, a driver can drive a tractor without a trailer to reposition himself or herself for the next leg. Tractor-only movements are used sparingly in a good set of schedules.

The driver-scheduling problem is formulated mathematically as a set-partitioning problem, a special type of integer-programming (IP) problem (Nemhauser and Wolsey 1988; Bradley et al. 1977).

10.2.1 Notation

Parameters

J = set of schedules (columns)

I_{leg} = set of legs (rows)

I_{dom} = set of domiciles (rows)

I_{sort} = set of sort intervals (rows)

c_j = cost of column j

a_{ij} = 1 if column j contains leg i; 0 otherwise

b_{ij} = 1 if column j has $i \in I_{\text{dom}}$ as its domicile; 0 otherwise

k_i^{lo} = minimum number of times domicile i must be used

k_i^{hi} = maximum number of times domicile i can be used

g_{ij} = number of packages contributed by column j to sort interval i

u_i = minimum number of packages required for sort interval i

Decision variables

x_j = 1 if column j is selected; 0 otherwise

10.3. Set-Partitioning Formulation with Side Constraints

$$\text{Min } \Sigma_{j \in J}\, c_j\, x_j \tag{40}$$

subject to

$$\Sigma_{j \in J}\, a_{ij}\, x_j = 1 \qquad \forall \text{ leg } i \in I_{\text{leg}} \qquad \text{(set-partitioning constraints)} \tag{41}$$

$$\Sigma_{j \in J}\, b_{ij}\, x_j \geq k_i^{\text{lo}} \qquad \forall \text{ domicile } i \in I_{\text{dom}} \qquad \text{(lower-domicile constraints)} \tag{42}$$

$$\Sigma_{j \in J}\, b_{ij}\, x_j \leq k_i^{\text{hi}} \qquad \forall \text{ domicile } i \in I_{\text{dom}} \qquad \text{(upper-domicile constraints)} \tag{43}$$

$$\Sigma_{j \in J}\, g_{ij}\, x_j \geq u_i \qquad \forall \text{ sort interval } i \in I_{\text{sort}} \qquad \text{(volume-availability constraints)} \tag{44}$$

$$x_j = 0 \text{ or } 1 \qquad \forall \text{ schedule } j \in J \qquad \text{(binary constraints)} \tag{45}$$

The objective function (40) minimizes the total cost of schedules. Constraints (41) are the set-partitioning constraints ensuring that each leg (row) is used by only one schedule (column). Constraints (42) and (43) are the domicile constraints and ensure that the lower and upper bounds for the selection of domiciles are met. According to work rules, each particular terminal must be used as a domicile a number of times within a given range. Constraints (44) are the volume-availability constraints and ensure that the required volume of packages is available for each sort interval. Constraints (45) are the binary constraints.

The presented formulation is a set-partitioning problem with additional side constraints. For a freight transportation company with 10,000 tractors and 15,000 drivers in the continental United States, the problem formulated above is too large to be solved for the whole country. Often, however, driver-scheduling decisions are made at the local level and the problem is broken naturally into smaller problems that are solved locally by each region.

It is sometimes difficult to obtain even a feasible solution of the IP problems formulated in (40)–(45). If the feasible region contains few feasible solutions or if there are errors in the data that make it difficult or impossible to find a feasible solution, the set-partitioning formulation (40)–(45) can be changed into a set-covering formulation (Nemhauser and Wolsey 1988; Bradley et al. 1977) with soft domicile and volume-availability constraints to help guide the cleaning of data and the solution process. Slack and surplus (auxiliary) variables are added to the equations and inequalities (41)–(45) and incorporated into the objective function (40) with very high costs. The following additional notation is introduced:

d_i = very high cost for auxiliary variable of row i.
s_i^+ = surplus variable for row i; if positive, it indicates the number of units that the original constraint is below its right-hand-side.
s_i^- = slack variable for row i; if positive, it indicates the number of units that the original constraint is above its right-hand-side.

10.4. Set-Covering Formulation with Soft Constraints

$$\text{Min } \Sigma_{j\in J}\, c_j\, x_j + \Sigma_{i\in I_{\text{leg}}} d_i s_i^- + \Sigma_{i\in I_{\text{dom}}} d_i s_i^+ + \Sigma_{i\in I_{\text{dom}}} d_i s_i^- + \Sigma_{i\in I_{\text{sort}}} d_i s_i^+ \tag{46}$$

subject to

$$\Sigma_{j\in J}\, a_{ij}\, x_j - s_i^- = 1 \qquad \forall \text{ leg } i \in I_{\text{leg}} \qquad \text{(set-covering constraints)} \tag{47}$$

$$\Sigma_{j\in J}\, b_{ij}\, x_j + s_i^+ \geq k_i^{\text{lo}} \qquad \forall \text{ domicile } i \in I_{\text{dom}} \qquad \text{(soft lower-domicile constraints)} \tag{48}$$

$$\Sigma_{j\in J}\, b_{ij}\, x_j - s_i^- \leq k_i^{\text{hi}} \qquad \forall \text{ domicile } i \in I_{\text{dom}} \qquad \text{(soft upper-domicile constraints)} \tag{49}$$

$$\Sigma_{j\in J}\, g_{ij}\, x_j + s_i^+ \geq u_i \qquad \forall \text{ sort interval } i \in I_{\text{sort}} \qquad \text{(soft volume-availability constraints)} \tag{50}$$

$$x_j = 0 \text{ or } 1 \qquad \forall \text{ schedule } j \in J \tag{51}$$

$$s_i^+ \geq 0 \qquad \forall \text{ row } i \tag{52}$$

$$s_i^- \geq 0 \qquad \forall \text{ row } i \tag{53}$$

The high value of the costs d_i of the slack and surplus variables prevents their inclusion in a solution with positive values, if this is possible. Constraints (47) resemble set-partitioning constraints but they can also be considered as set-covering constraints, that penalize overcoverage. If some slack or surplus variables have positive values in a solution, they may help identify reasons for not obtaining a true feasible solution or they may remain in the solution for as long as necessary in an iterative solution process that will be discussed later.

10.5. Column-Generation Methodology

Each one of the regional problem formulations of a large freight transportation company may have up to 2000 legs (2400 rows altogether) that can be combined into possibly billions of feasible schedules. Even if the binary constraints (51) are relaxed, it is impossible to solve the resulting linear program (LP) with all the columns included. Instead, a standard decomposition method for large LPs called column generation is used (Bradley et al. 1977; Chvátal, 1983).

When the binary constraints (51) are replaced with the following bounding constraints

$$0 \leq x_j \leq 1 \qquad \forall \text{ schedule } j \in J$$

a column generation approach refers to the resulting LP problem that includes all the possible columns as the master problem. The corresponding LP problem that includes only a subset of the columns is called the restricted master problem.

Solving the LP involves pricing out each column using the dual variables or shadow prices associated with the restricted master problem. Sometimes this pricing-out operation can be formulated as a known problem (e.g., a shortest-path problem) and is called a subproblem. The solution of the subproblem produces a column, not yet included in the restricted master problem, that prices out best. For a minimization problem, if the best new column obtained has negative reduced cost, its inclusion in the restricted master problem will improve the solution. Column generation iterates solving the subproblem, adding a new column with negative reduced costs to the restricted master problem, and solving the new restricted master problem until no more columns can be obtained with negative reduced costs. At that point the master problem has been solved optimally since all billions of possible schedules have been examined implicitly.

When the generation of columns cannot be structured as a known optimization problem, they must be generated explicitly. Usually, this approach results in solving the master problem heuristically because optimality is guaranteed only if all the feasible columns are examined. The schedules for the set-partitioning or set-covering problems presented above are obtained explicitly. The nature of the breaks imposed by work rules, the possible existence of additional local work rules in some regions, the presence of legs that carry priority loads and need to be included as early as possible in a schedule, and other local characteristics make it very difficult to generate schedules by solving a structured subproblem.

The schedules obtained from the solution of the LP problem by column generation may be fractional and therefore unacceptable. To obtain feasible schedules, an IP problem needs to be solved. An iterative heuristic approach for solving the driver-scheduling problem is presented next.

10.6. Iterative Process for Solving the Driver-Scheduling Problem with Column Generation

1. Start with a feasible solution. Set up an LP that consists of the columns of the feasible solution.
2. Solve the LP (restricted master problem). If the LP cost is low enough or a given number of iterations is reached, go to step 4.
3. Using the LP shadow prices, generate up to a given number of good new schedules and add them to the LP. If the number of columns exceeds a given maximum, remove columns with the worst reduced costs. Go to step 2.
4. Solve the IP.

In step 1, the algorithm starts with a feasible solution, which can be obtained from the currently used set of schedules. These schedules are broken apart to obtain the set of legs for the problem. In step 2, an LP problem is solved that consists of the current set of schedules and shadow prices are obtained for the rows. In step 3, the algorithm generates more schedules with reduced costs less than a set value using the shadow prices. The new schedules are added to the LP, omitting duplicate schedules. Because the LP is a minimization problem, columns with negative reduced costs will reduce the objective function value. More than one column is added to the LP at each iteration. For this reason, columns with low positive reduced costs may also be accepted. Such columns are also valuable in solving the IP problem in step 4. If the total number of columns in the restricted master problem exceeds a preset maximum number, the columns with the worst (highest) reduced costs are deleted. Steps 2 and 3 are repeated until an LP cost is obtained that is below a preset cutoff value or until a given number of iterations is reached. The resulting IP problem is solved in step 4.

In actual applications, the IP problem at step 4 of the previous algorithm obtained from column generation turned out to be a very difficult problem to solve with existing IP solvers. There were several reasons for this difficulty. The optimal LP solution at the beginning of step 4 included too many fractions, the problem structure exhibited massive degeneracy, and there were too many alternative optimal solutions. It was even difficult for IP solvers to obtain feasible solutions for large problems. Because of these difficulties, the following heuristic approach has been used that combines column generation with solution of the IP problem.

10.7. Integrated Iterative Process for Solving the Driver-Scheduling Problem

1. Start with a feasible solution. Set up an LP that consists of the columns of the feasible solution.
2. Solve the LP (restricted master problem). If the LP cost is low enough or a given number of iterations is reached, go to step 4.
3. Using the LP shadow prices, generate up to a given number of good new schedules and add them to the LP. If the number of columns exceeds a given maximum, remove columns with the worst reduced costs. Go to step 2.
4. Select a small number of columns with the highest fractional values, say 8. Using their legs as seeds, generate a small number of additional schedules, say 300, add to the LP, and solve it.
5. Select a small number of the highest fractional schedules, say 8. Restrict them to be integers and solve the resulting mixed-integer-programming (MIP) problem. If all variables in the solution of the current restricted master problem are integral, stop. Otherwise, set the selected columns to their integer solution values permanently, update the formulation, and go to step 2.

Steps 1, 2, and 3 are the same as before. In step 4, a preset number of columns with the highest fractional values are selected. The actual number used is set by experimentation. The legs making up these columns are used as seeds to generate a small number of additional columns that are added to the LP. The LP is solved and a small number of columns with the highest fractional values are selected and restricted to be integral. The resulting MIP problem is solved to optimality using an MIP solver. If all the columns of the restricted master problem have integral values in the solution, the algorithm stops. If some fractional values are still present in the solution, the selected columns to be integral are set permanently to their integer solution values and eliminated from the formulation and the column-generation phase starts again. In applications, up to about 40,000 columns were included in the restricted master problem during the iterative solution process.

10.8. Generation of Schedules

Approaches for generating new schedules are discussed in this section. Several techniques can be used to obtain new schedules explicitly, guided by the LP shadow prices. Each leg is associated with a shadow price in the LP solution, which indicates how expensive it is to schedule this leg. The generation of schedules focuses on the expensive legs to provide more alternative schedules that include them and drive the LP cost down. New schedules improve the LP solution if they have negative reduced costs. Usually the cutoff value is set higher than zero to include schedules with low positive costs that may combine well with the rest of them because columns are not added one at a time to the LP.

New schedules are generated probabilistically. Each available leg is assigned a probability of selection proportional to its shadow price. The list of legs is then shuffled using the assigned probabilities of selection as weights.* This means that legs with high shadow prices are more likely to be at the top of the shuffled list. Each leg in the shuffled list is used as a seed sequentially to generate up to a given number of legs.

Starting with a seed, schedules are generated using three different approaches: one based on depth-first search (Aho et al. 1983), a second approach that generates a given number of schedules in parallel, and a third method that recombines existing schedules to produce new and better ones. The schedules are generated by limited complete enumeration, that is, all the schedules that can be generated starting with a particular seed are generated, limited by an upper bound when too many combinations exist. Tractor-only movements for repositioning drivers are also used in the generation of feasible schedules. A partial schedule is accepted only if a complete feasible schedule can be generated from it, including appropriate work breaks. When a maximum total number of columns is obtained, the process stops.

The depth-first-search approach starts with a seed and adds legs sequentially until a complete schedule is obtained. Then a new schedule is started using either the same seed or the next seed in the list. All the feasible successors of a leg are shuffled based on their shadow prices as weights and used to obtain the next leg of a partial schedule. The parallel approach generates several schedules simultaneously by adding each one of its feasible successors to the current partial schedule.

The recombination approach identifies feasible schedules that are generated by removing one or more consecutive legs from one feasible schedule and replacing them with one or more legs from another schedule. Cycles of leg exchanges are then identified that produce new schedules of lower costs. In actual applications, the recombination approach tends to produce columns that drive the iterative solution process much more quickly toward a good overall result than when only columns obtained by the other two approaches are included.

10.9. Beyond Algorithms

Good algorithms that capture well the complexities of real-world problems are a big step towards achieving efficiency using optimization techniques. They are rarely, however, sufficient by themselves. The perfect algorithm is useless if it is not actually used, if it not used properly, or if the solution is not implemented. This is particularly important in the transportation of goods, which is a labor-intensive industry, and where the implementation of an optimization system may involve and affect a large number of people.

Often, a big challenge, beyond the development of good algorithms, is defining the correct problem to solve, finding the necessary data, setting up tools to extract the needed data, correcting and validating the data, having the model used correctly, and obtaining acceptance of the model and its results. A successful, decentralized application needs the involvement of the users, who must be able and willing to use it correctly and apply the results.

*This is exactly like regular shuffling except for the probabilities of selection that are not uniform. An ordered list of legs is obtained by randomly selecting one leg at a time from an original list of legs, based on the weights.

To obtain good results using the algorithms described in this chapter, correct data need to be used. Obtaining good, correct data is often a difficult, expensive, and time-consuming task. Errors in the data need to be identified and corrected easily for the system to succeed. The driver-scheduling problem is especially sensitive to the values of the time windows and cost input. An interface needs to be available that handles the cleaning of the data as well as the generation of the input to the algorithms in an appropriate format and that after a solution is obtained produces useful reports.

If a computer system is put in place for the first time to replace a manual system, the data needs of the computer system may be much larger than those of the manual system. Most optimization models obtain solutions by comparing explicitly or implicitly large numbers of alternatives for which data must be available. Manual systems, on the other hand, often obtain a new solution by modifying an existing one, that is, they examine very few alternatives and need fewer input data. The difference in input data also means that input-data needs for solving an existing application often have to be redefined for computer models. If good input data cannot be obtained for the computer model, its use cannot be expected to improve results.

The algorithms described previously are used to solve difficult MIP problems. Large computers and expensive commercial LP and IP solvers need to be used, which are usually available only in centralized locations. The interface that cleans the data and obtains the input can be implemented locally when the driver-scheduling system is applied separately by region.

A system like the one described has been deployed successfully by United Parcel Service using two different platforms. The data preparation is implemented on local computers available at all company locations. The optimization algorithms are solved remotely on large computers at another location using the company's intranet. The fact that two different platforms are used is hidden from the users, who are informed by e-mail about the status of a submitted job at various stages and get all the reports on the intranet. This kind of application is now possible because of the increase in computer power that permits the solution of difficult optimization problems in reasonable time and because of the evolution of the Internet that provides the tools supporting a remote implementation.

To improve the probability of success, user training is very important for both actual use of the system and interpretation of results. Optimization systems sometimes do not include some difficult characteristics of a real-life problem. In cases like this, interpretation of results, modification of the solution, or simply selection among alternative solutions is very important. A pilot implementation is also very helpful.

An important element for the success of the system is the existence of appropriate support to the users in applying the system, interpreting the results, and making appropriate decisions. This is especially important when the system results differ from current practices and need to be understood and their validity accepted or when they need to be slightly modified. A team of experts in both the computer system and the operational problem that is solved, who are accepted by the users and speak the same business language, needs to be available to assist the users, especially when the system is first deployed.

The optimization model described minimizes total cost of schedules, which often results in a solution using fewer drivers than those used in the current solution. How such a solution is implemented depends on company–labor agreements, which may differ among companies and regions. In the short term, available drivers beyond the number needed by the solution may be put on an ''on-call'' list and asked to come to work only if another driver is sick or a particular need arises.

It is generally very important to get the final users involved early on in the development process. In this way, the developer makes sure that the correct problem is solved and user needs are met as much as possible. Including the user in the development process early on increases the probability of acceptance of the model.

For the implementation of results, one other very significant factor for success concerns the reward structure of a company. If an optimization model minimizes cost but the user is not rewarded directly for implementing a solution that minimizes cost, the solution results are unlikely to be used. If a model minimizes the number of drivers but the decision maker is not rewarded for using fewer drivers, he is unlikely to jeopardize the goodwill of the people working with him by making a big effort to change the status quo.

11. QUALITY IN TRANSPORTATION

Companies that specialize in the transportation of goods must manage their costs, growth, and quality in order to remain competitive. However, without the appropriate measures and the systems to support performance measurement, it is practically impossible to manage any transportation system. In measuring quality in the freight-transportation industry several questions arise. First, of course, is the definition of quality itself. In this industry, quality is viewed differently at different steps in the transportation process. Shippers have different requirements from those of the receivers. Different internal processes have different views of quality and its measurement. However, we can summarize these requirements into five categories:

1. *Damages:* Was the shipment damaged in the process?
2. *On-time performance:* Were all service guarantees met?
3. *Accuracy:* Was the shipment delivered to the correct destination?
4. *Shipment integrity:* Were all the items in a shipment delivered together?
5. *Information integrity:* Was the information associated with a shipment available at all times?

The primary objective of the transportation-planning activity is to design processes that maintain high levels of performance in all five categories. Let's explore these requirements further:

- *Damages:* Of all the quality factors discussed, damages are perhaps one of the most important indicators of quality in both the receiver's and the shipper's view. When the freight-transportation company damages the merchandise being moved, the shipper, the receiver, and the transportation company itself are affected. Costs associated with insurance, returns, product replacement, and lost productivity are all a result of damaged goods. All transportation processes must be designed to prevent damaging the goods. Usually, every step in the transportation process has procedures to measure the number of damages created in any period of time. These procedures are used to establish accountability practices and to identify process-improvement opportunities.
- *On-time performance:* Freight-transportation companies compete on the basis of service performance and cost. In order to support the needs of the complex supply chains that exist today, high levels of on-time delivery reliability are expected from the transportation company. Several external and internal factors can have a direct impact on on-time delivery performance. External factors such as weather, traffic conditions, and subcontractor labor relations can have a direct impact on the ability of the transportation company to meet service commitments. With proper planning, transportation companies manage to minimize the impact of some of these factors. On the other hand, the one internal factor that will always have a negative impact on on-time delivery is lack of planning or, quite simply, poor planning. If the organization is not prepared to handle seasonal variations in pickup and delivery volumes or does not have contingency plans to deal with unexpected events, on-time delivery performance will be affected.
- *Accuracy:* Delivering to the correct destination is expected every time for every shipment. However, there are instances in which the transportation system fails to satisfy this basic requirement. Two major causes contribute to this type of service failure: missing or incorrect information and inadequate planning. For example, when the wrong address is attached to a shipment, the probability of making delivery mistakes increases substantially. Today, transportation companies offer a variety of services aimed at providing continuous shipment tracking and improved information quality. As indicated before, labor is usually the highest cost variable in the profitability equation of a transportation company. Labor is also the primary driver of quality in these organizations. When companies fail to develop staffing and training plans properly, delivery accuracy and reliability will be impacted.
- *Shipment integrity:* Receivers expect to receive all the items (e.g., packages) in a shipment on the same day and at the same time. It is the responsibility of the freight-transportation company to maintain the integrity of all shipments. The transportation system must be capable of using shipment information in its different processes to ensure the integrity of every shipment.
- *Information integrity:* As indicated earlier in this chapter, the information about a shipment is as important, in today's supply chains, as the movement of the shipment itself. Since shipment information is offered to shippers and receivers as a value-added service, the effectiveness with which this information is provided to them must be monitored and measured. Systems to effectively capture, store, and provide shipment information are critical in today's freight transportation business models. The freight-transportation industry will continue to be an information-based industry. Therefore, maintaining high levels of information accuracy and integrity will continue to be an important measure of performance.

Now that some basic measures have been defined, let's look into the quality-improvement process. Like other industries, freight-transportation companies make use of well-established quality-improvement techniques. Without a clear understanding of the facts and figures that affect the quality of the services offered, management cannot control and improve the processes involved. Remember that there are four continuous phases in the quality improvement process: Plan, Do, Check, and Act. Together, these phases are known as the Deming circle. Transportation processes must be designed and performance targets must be defined. As activities are completed throughout the different processes, regular checks must take place as the processes are monitored and controlled. The information gathered from these checks must be used to improve the process continuously.

12. TECHNOLOGY

12.1. Vehicle Routing

For solving transportation problems, the use of computers plays an important role in the development of models, schedules, network plans, and delivery and pickup routes. As described in previous sections, the complexity and magnitude of the transportation problems discussed in this chapter require extensive computation times. Making transportation decisions is a complex process. Dispatch managers in package-delivery companies must assess a variety of factors before making dispatch decisions. Some of those factors were discussed in previous sections. They include vehicle capacity, time windows, demand fluctuations, labor productivity, and the dynamic changes in pickup and delivery characteristics of customers, geographies, and traffic conditions.

Vehicle-routing problems fall into one of three basic segments: routing of service vehicles, passenger vehicles, and freight vehicles (Hall and Partyka 1997). Service vehicles are used to support jobs in the field and are generally used by service technicians. In this type of problem, the primary constraints are service time, time windows, and travel time. Because there is no major variation in the merchandise carried by any given vehicle, capacity constraints are not included in the formulation of service routes. Passenger-transportation services such as bus service face an additional constraint: capacity. The size of the vehicle determines the number of passengers that can be safely carried from one point to another. Freight vehicles are also constrained by their capacity. When completing pickups, a vehicle may run out of capacity at a certain customer location. At this point, the dispatcher must either dispatch another vehicle to the customer's location to complete service or ask the current driver to return to the depot, obtain an empty vehicle, and return to the customer's location to complete the service. It is clear from this example that the decision made by the dispatcher can have different cost and service implications and may affect more than one customer. A few years ago, the only way to make these dispatch decisions effectively was through human knowledge and experience. The ability of the dispatcher to make numerous decisions based on a few pieces of information could make or break the dispatching process. Today, things have changed. With the increasing implementation of technologies such as the geographic information systems (GIS) and global-positioning systems (GPS), dispatchers have new tools that automate and improve the vehicle-routing process.

Routing pickup-and-delivery vehicles does not end with the development of routes prior to the drivers' departure from the depot. Once drivers have left the depot, in-vehicle communications and route-information systems offer mechanisms not only to improve their performance but to meet on-demand customer requests. When dispatchers have the ability to communicate routing instructions and customer requests to drivers in the field, the opportunities for improving the overall efficiency of a dispatch plan increase substantially. However, the initial development of an efficient and effective dispatch plan is still critical.

Several software vendors have developed vehicle-routing software. In 1997, Hall and Partyka surveyed several vendors in order to compare the characteristics of their respective software systems. Table 4 presents an extract of this survey. The complete survey appeared in the June 1997 issue of *OR/MS Today.*

12.2. Information Gathering and Shipment Tracking

As indicated earlier in this chapter, the information associated with the goods being moved is as important as the transportation process itself. Today, transportation companies use a variety of tools to track, manage, and control the movement of goods from pickup to delivery. In addition, sophisticated electronic devices are being used by drivers not only to record the status of deliveries and pickups, but also to keep track of vehicle usage, time cards, and sales information.

12.3. New Trends: Intelligent Transportation Systems (ITS)

The transportation community has turned to the deployment of intelligent transportation systems (ITS) to increase the efficiency of existing highway, transit, and rail systems. One of the key variables in the vehicle-routing models described above is travel time. With the use of information from ITS, dispatchers can make better decisions. The U.S. Department of Transportation (DOT) has indicated that "ITS uses advanced electronics and information technologies to improve the performance of vehicles, highways, and transit systems. ITS provides a variety of products and services in metropolitan and rural areas."

As ITS evolves from pure research, limited prototyping, and pilot projects into routine usage, decision makers at "the corporate, state, regional, and local levels seek reliable information about the contribution that ITS products can make toward meeting the demand for safe and efficient movement of people and goods." Literature indicates that substantial benefits have already been realized in areas such as accident reduction, travel-time savings, customer service, roadway capacity, emission reduction, fuel consumption, and vehicle stops. Greater benefits are predicted with more extensive

TABLE 4 Routing Software Survey

		Solvable Problem Size			Routing				
Product	Publisher	Number of Stops	Number of Vehicles	Number of Terminals	Real-Time Routing	Daily Routing	Route Planning	GIS Product Interface	Special Features
GeoRoute	Kositzky & Associates, Inc.	4600	512	256	N	Y	Y	GeoWhiz	GeoRoute works for local delivery as well as over-the-road applications. Options for multidepot and redisptach are available.
GeoRoute 5	GIRO Enterprises, Inc.	Unlimited	Unlimited	Unlimited	N	Y	Y	ArcInfo, MapInfo	Software supports both point-to-point and street-by-street operations, as well as mixed requirements.
Load Manager	Roadnet Technologies, Inc.	N/A	N/A	N/A	N	N	N		
LoadExpress Plus	Information Software, Inc.	Unlimited	500	Unlimited	N	Y	Y	Proprietary, can work with data from all GIS systems	LoadExpress is a simple, powerful, and flexible choice for building and optimizing routes, scheduling deliveries, and analyzing distirbution patterns.
Manugistics Routing & Scheduling	Manugistics, Inc.	Unlimited	Unlimited	Unlimited	Y	Y	Y	Xeta, Rockwell, Cadec, Autoroach	Resource management—allows management of driver, tractor and trailer schedules, provides information on equipment requirements.
OVERS	Bender Management Consultants	10,000	1000	100	Y	Y	Y	Map Objects	Can optimize routes across multiple time periods, respect space/time constraints, optimize number and location of terminals and service areas.
RIMMS	Lightstone Group, Inc.	Unlimited	Unlimited	Unlimited	Y	Y	Y	ESRI shape files	Configurable by users across multiple industries, including both scheduling model and screen cosmetics. Interfaces via ODBC drivers.

ROADNET 5000	Roadnet Technologies, Inc.	Unlimited	Unlimited	Unlimited	N	Y	N	GDT maps	
RoadShow for Windows	ROADSHOW International, Inc.	8000	Unlimited	Unlimited	Y	Y	Y	GDT, Etak, MapInfo and proprietary software	ROADSHOW calculates cost-effective solutions based on actual costs incorporating specific information supplied by the user.
RoutePro	CAPS LOGISTICS	HW-based	HW-based	HW-based	Y	Y	Y	Etak, Horizons Technology, GDT, PCMiler	Customizable through fourth-generation macro language and ability to call functions from other languages.
RouteSmart Neighborhood	RouteSmart Technologies	Unlimited	Unlimited	1+ Intermediates	N	N	Y	GIS Plus (DOS version), ArcInfo	Meter-reading system, handles walking, driving, and combination routes.
RouteSmart Point-to-Point	RouteSmart Technologies	Unlimited	Unlimited	1+ Intermediates	N	Y	Y	ArcView version 3.0	
Routronics 2000	Carrier Logistics	Unlimited	Unlimited	Unlimited	N	Y	N	MapInfo	Routronics 2000 has been developed as a complete customer-service routing and dispatch system with interfaces to wireless communications.
SHIPCONS II	Insight, Inc.	Unlimited	Unlimited	Unlimited	N	Y	Y	GDT, MapInfo, Etak	Cost-based, integer optimization; user-configurable screens; Ad Hoc Report Writer; Digital Geography; Shipment Rater for TL and LTL carriers.
Taylor II	F&H Simulations, Inc.	1000	100	1000	N	N	Y		2 and 3D animation; Windows 95 and Windows NT; design of experiments; curve fitting of raw data (advanced statistics module).
Territory Planner	Roadnet Technologies	Unlimited	Unlimited	Unlimited	N	N	Y	GDT maps	
TESYS	Inform Software Corporation	3000	1000	500	Y	Y	N	CDPD/GPS/RF	
TransCAD	Caliper Corporation	Unlimited	Unlimited	Unlimited	Y	Y	Y	Maptitude, GIST, can work with data from all GIS systems	Toolkit of OR methods including min-cost network flow, transportation problem, and various optimal location methods.

Source: R. W. Hall and J. G. Partyka, ‘‘On the Road to Efficiency,’’ *OR/MS Today,* June 1997, pp. 38–47. Reprinted with permission.

deployment of more mature products. Freight-transportation companies face new constraints and challenges not only in meeting service commitments but in remaining competitive and cost effective while meeting governmental regulations. The use of ITS offers new opportunities to use information in the development of routes and schedules.

The ITS program is sponsored by the DOT through the ITS Joint Program Office (JPO), the Federal Highway Administration (FHWA), and the Federal Transit Administration (FTA).

ITS, formerly known as the intelligent vehicle highway systems (IVHS), were created after the Intermodal Surface Transportation Efficiency Act (ISTEA) of 1991 was established. ISTEA helped authorize larger spending for transit improvement. In January 1996, then Secretary of Transportation Frederico Peña launched Operation TimeSaver, which seeks to install a metropolitan intelligent transportation infrastructure in 75 major U.S. cities by 2005 to electronically link the individual intelligent transportation systems, sharing data so that better travel decisions can be made.

A projected $400 billion will be invested in ITS by the year 2011. Approximately 80% of that investment will come from the private sector in the form of consumer products and services.

The DOT has defined the following as the components of the ITS infrastructure:

- *Transit fleet management:* enables more efficient transit operations, using enhanced passenger information, automated data and fare collection, vehicle diagnostic systems, and vehicle positioning systems
- *Traveler information:* linked information network of comprehensive transportation data that directly receives transit and roadway monitoring and detection information from a variety of sources
- *Electronic fare payment:* uses multiuse traveler debit or credit cards that eliminate the need for customers to provide exact fare (change) or any cash during a transaction
- *Traffic signal control:* monitors traffic volume and automatically adjusts the signal patterns to optimize traffic flow, including signal coordination and prioritization
- *Freeway management:* provides transportation managers the capability to monitor traffic and environmental conditions on the freeway system, identify flow impediments, implement control and management strategies, and disseminate critical information to travelers
- *Incident management:* quickly identifies and responds to incidents (crashes, breakdowns, cargo spills) that occur on area freeways or major arteries
- *Electronic toll collection:* uses driver-payment cards or vehicle tags to decrease delays and increase roadway throughput
- *Highway–rail intersection safety systems:* coordinates train movements with traffic signals at railroad grade crossings and alerts drivers with in-vehicle warning systems of approaching trains
- *Emergency response:* focuses on safety, including giving emergency response providers the ability to pinpoint quickly the exact location of an incident, locating the nearest emergency vehicle, providing exact routing to the scene, and communicating from the scene to the hospital

The use of information from all of these system components will enhance the planner's ability in designing efficient transportation networks and delivery routes. In addition, as this information is communicated to the drivers, they will also have the capability of making better decisions that will enhance customer satisfaction and reduce overall costs.

For additional information, visit the DOT's website on ITS: http://www.its.dot.gov/.

Acknowledgments

The authors wish to thank Professor Mark S. Daskin of Northwestern University and Professor Ching-Chung Kuo of Pennsylvania State University for their constructive comments. They also thank their colleagues Ranga Nuggehalli, Doug Mohr, Hla Hla Sein, Mark Davidson, and Tai Kim for their assistance. They also thank Dr. Gerald Nadler of the University of Southern California for all his support.

REFERENCES

Aho, A. V., Hopcroft, J. E., and Ullman, J. D. (1983), *Data Structures and Algorithms*, Addison-Wesley, Reading, MA.

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993), *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, NJ.

Bartholdi, J., and Platzman, L. (1988), "Heuristics Based on Spacefilling Curves for Combinatorial Problems in Euclidean Space," *Management Science*, Vol. 34, pp. 291–305.

Bertsimas, D., Jaillet, P., and Odoni, A. (1990), "A Priori Optimization," *Operations Research*, Vol. 38, pp. 1019–1033.

Bradley, S. P., Hax, A. C., and Magnanti, T. L. (1977), *Applied Mathematical Programming*, Addison-Wesley, Reading, MA.

Chvátal V. (1983), *Linear Programming*, W.H. Freeman, New York.

Crowder, H. (1976), "Computational Improvements for Subgradient Optimization," *Symposia Mathematica*, Vol. 19, pp. 357–372.

Daskin, M. S. (1995), *Network and Discrete Location: Models, Algorithms, and Applications*, John Wiley & Sons, New York.

Desrosiers, J., Dumas, Y., Solomon, M. M., and Soumis, F. (1995), "Time Constrained Routing and Scheduling," in *Network Routing*, Vol. 8 of *Handbooks in Operations Research and Management Science,* M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, Eds., North-Holland, Amsterdam, pp. 35–139.

Fisher, M. L. (1985), "An Applications Oriented Guide to Lagrangian Relaxation," *Interfaces*, Vol. 15, No. 2, pp. 10–21.

Fisher, M., and Jaikumar, R. (1981), "A Generalized Assignment Heuristic for Vehicle Routing," *Networks*, Vol. 11, pp. 109–124.

Garey, M., and Johnson, D. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, New York.

Geoffrion, A. M. (1974), "Lagrangian Relaxation for Integer Programming," *Mathematical Programming Study*, Vol. 2, pp. 82–114.

Gillet, B., and Miller, L. (1974), "A Heuristic Algorithm for the Vehicle Dispatching Problem," *Operations Research*, Vol. 22, pp. 340–349.

Glover, F. (1989), "Tabu Search—Part I," *ORSA Journal on Computing*, Vol. 1, pp. 190–206.

Glover, F. (1990), "Tabu Search—Part II," *ORSA Journal on Computing*, Vol. 2, pp. 4–32.

Hall, R. W., and Partyka, J. G. (1997), "On the Road to Efficiency," *OR/MS Today*, June, pp. 38–47.

Hall, R., Du, Y., and Lin, J. (1994), "Use of Continuous Approximations within Discrete Algorithms for Routing Vehicles: Experimental Results and Interpretation," *Networks*, Vol. 24, pp. 43–56.

Jaillet, P. (1988), "A Priori Solution of a Traveling Salesman Problem in Which a Random Subset of the Customers Are Visited," *Operations Research*, Vol. 36, pp. 929–936.

Jaillet, P., and Odoni, A. (1988), "The Probabilistic Vehcile Routing Problem," in *Vehicle Routing: Methods and Studies*, B. Golden and A. Assad, Eds., North-Holland, Amsterdam, pp. 293–318.

Kontoravdis, G., and Bard, J. (1995), "A GRASP for the Vehicle Routing Problem with Time Windows," *ORSA Journal on Computing*, Vol. 7, pp. 10–23.

Lawler, E., Lenstra, J., Rinnooy Kan, A., and Shmoys, D., Eds. (1985), *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, John Wiley & Sons, New York.

Lin, S., and Kernighan, B. (1973), "An Effective Heuristic Algorithm for the Traveling Salesman Problem," *Operations Research*, Vol. 21, pp. 498–516.

Martello, S., and Toth, P. (1990), *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley & Sons, New York.

Minieka, E. (1978), *Optimization Algorithms for Networks and Graphs*, Marcel Dekker, New York.

Nemhauser, G. L., and Wolsey, L. A. (1988), *Integer and Combinatorial Optimization*, John Wiley & Sons, New York.

Potvin, J., and Rousseau, J. (1993), "A Parallel Route Building Algorithm for the Vehicle Routing and Scheduling Problem with Time Windows," *European Journal of Operational Research*, Vol. 66, pp. 331–340.

Potvin, J., and Rousseau, J. (1995), "An Exchange Heuristic for Routing Problems with Time Windows," *Journal of the Operational Research Society*, Vol. 46, pp. 1433–1446.

Potvin, J., Kervahut, T., Garcia, B., and Rousseau, J. (1996), "The Vehicle Routing Problem with Time Windows—Part I: Tabu Search," *INFORMS Journal on Computing*, Vol. 8, pp. 158–164.

Rochat, Y., and Taillard, E. (1995), "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing," *Journal of Heuristics*, Vol. 1, pp. 147–167.

Russell, R. (1995), "Hybrid Heuristics for the Vehicle Routing Problem with Time Windows," *Transportation Science*, Vol. 29, pp. 156–166.

Savelsbergh, M. (1985), "Local Search in Routing Problems with Time Windows," *Annals of Operations Research*, Vol. 4, pp. 285–305.

Savelsbergh, M. (1990), "An Efficient Implementation of Local Search Algorithms for Constrained Routing Problems," *European Journal of Operational Research*, Vol. 47, pp. 75–85.

Savelsbergh, M. (1992), ‘‘The Vehicle Routing Problem with Time Windows: Minimizing Route Duration,’’ *ORSA Journal on Computing*, Vol. 4, pp. 146–154.

Solomon, M. (1987), ‘‘Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints,’’ *Operations Research*, Vol. 35, pp. 254–265.

Solomon, M., Baker, E., and Schaffer, J. (1988), ‘‘Vehicle Routing and Scheduling Problems with Time Window Constraints: Efficient Implementations of Solution Improvement Procedures,’’ in *Vehicle Routing: Methods and Studies*, B. Golden and A. Assad, Eds., North-Holland, Amsterdam, pp. 85–105.

Taillard, E., Badeau, P., Gendreau, M., Guertin, F., and Potvin, J. (1997), ‘‘A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows,’’ *Transportation Science*, Vol. 31, pp. 170–186.

Thangiah, S., Osman, I., and Sun, T. (1995), ‘‘Metaheuristics for Vehicle Routing Problems with Time Windows,’’ Technical Report, Computer Science Department, Slippery Rock University, Slippery Rock, PA.