Third Party Logistics Planning with Routing and Inventory Costs

Alexandra M. Newman
Division of Economics and Business
Colorado School of Mines
Golden, CO 80401

Candace A. Yano
Department of Industrial Engineering and Operations Research
and
The Haas School of Business
University of California
Berkeley, CA 94720-1777

Philip M. Kaminsky
Department of Industrial Engineering and Operations Research
University of California
Berkeley, CA 94720-1777

June 2001; revised June 2003, February 2004 and June 2004

Third Party Logistics Planning with Routing and Inventory Costs

Abstract

We address a scheduling and routing problem faced by a third-party logistics provider in planning its day-of-week delivery schedule and routes for a set of existing and/or prospective customers who need to make shipments to their customers (whom we call "end-customers"). The goal is to minimize the total cost of transportation and inventory while satisfying a customer service requirement that stipulates a minimum number of visits to each customer each week and satisfaction of time-varying demand at the end-customers. Explicit constraints on the minimum number of visits to each customer each week give rise to interdependencies that result in a dimension of problem difficulty not commonly found in models in the literature. Our model includes two other realistic factors that the third-party logistics provider needs to consider: the cost of holding inventory borne by end-customers if deliveries are not made "just-in-time" and the possibility of multiple vehicle visits to an end-customer in the same period (day).

We develop a solution procedure based on Lagrangian relaxation in which the particular form of the relaxation provides strong bounds. One of the subproblems that arises from the relaxation serves to integrate the impact of the timing of deliveries to the various end-customers with inventory decisions, which not only contributes to the strong lower bound that the relaxation provides, but also yields a mathematical structure with some unusual characteristics; we develop an optimal polynomial-time solution procedure for this subproblem. We also consider two variants of the original problem with more restrictive assumptions that are usually imposed implicitly in many vehicle routing problems. Computational results indicate that the Lagrangian procedure performs well for both the original problem and the variants. In many realistic cases, the imposition of the additional restrictive assumptions does not significantly affect the quality of the solutions but substantially reduces computational effort.

Keywords: Third-party logistics; vehicle scheduling; period vehicle routing problem; inventory routing problem; delivery scheduling

1 Introduction

Our research was motivated by a problem faced by a California-based third-party logistics (3PL) provider that offers shipping services in the form of full-truckload (or as-if-full-truckload) moves to its customers. Most of its customers are manufacturing firms that supply components to downstream manufacturers or finished goods to distribution centers, or distribution centers that supply large retail firms. Some of the manufacturing customers use the 3PL to provide transportation to support vendor-managed inventory (VMI) programs. For clarity, we use the term *customer* to refer to a purchaser of 3PL services and *end-customer* to refer to a customer's customers. Each customer needs to supply items to its end-customers to satisfy the end-customers' daily demands on or before their respective due dates. The 3PL provider has a contract with each of its customers to transport these goods, usually with a requirement on the minimum number of deliveries per week for each end-customer. Typically, the 3PL provider's customers would view more frequent delivery as an element of better customer service. Perhaps more importantly, the end-customers prefer more frequent deliveries to reduce their inventory holding costs, and the customers who are involved in VMI programs directly benefit from reduced inventory at the end-customers (if the customers own this inventory, as is common). We explicitly consider these factors in our model.

In this paper, we address the 3PL provider's problem of selecting routes to execute on each day of the week to service its existing and/or prospective customers. We consider the problem from the viewpoint of the 3PL provider, who is seeking a cost-efficient solution while satisfying customer service requirements that stipulate conditions such as frequency of delivery. Although the solution of this problem can be used for operational purposes, the 3PL management initially approached us seeking to estimate the cost of servicing a prospective customer in the context of preparing bids and/or negotiating the terms of a contract.

It is important to emphasize that the focus of this paper is on route selection and delivery quantity decisions, and not on the construction of candidate routes, because a typical 3PL provider has little difficulty generating a set of viable routes, taking into account the structure of the road network, traffic patterns, etc. Our primary concern is to find an effective solution procedure for our problem given a practical set of candidate routes.

The remainder of this paper is organized as follows. In the next section, we describe our problem in more detail and present a mathematical formulation. Section 3 contains a review of the related research literature. In Section 4, we discuss two restricted variants of our problem, and in Section 5, we present our proposed solution approach for the original problem and the two variants. Computational results are reported in Section 6, and Section 7 concludes the paper.

2 Model Description

Our research was motivated by a 3PL provider which, for reasons of material handling efficiency, usually requires its customers to palletize the goods to be shipped. For this reason and for ease of exposition, we assume that the volume of goods can be expressed in terms of a homogeneous unit, such as a standard pallet. In the formulation that follows, we assume that all customers use the *same* standard unit, but we only require that each customer's basic unit of shipment be sufficiently standardized that we do not have to address the "bin-packing" aspect of the truck loading problem.

We assume that the 3PL provider owns, leases, or otherwise controls a fleet of trucks and that the trucks are homogeneous. We also assume that the 3PL provider has sufficient trucks to service the selected routes. 3PL firms often have standing arrangements for rental vehicles when needed, and additional drivers are available except in unusual circumstances. Thus, although limitations due to the number of vehicles or drivers may exist, they do not play a major role in a 3PL provider's route planning decisions. When using our procedure to estimate the cost of servicing a new customer, the 3PL provider recognizes that additional vehicles may be needed, and generally would not want to constrain the number of vehicles *a priori*. In the concluding section, we explain how our approach can be generalized to accommodate non-homogeneous vehicles and the cost of rental trucks when required.

Our problem framework has many customers and many end-customers. To distinguish among origin-destination pairs, we use the term *job type* to refer to the movement of goods from a single customer location to a single end-customer location. Each "route" in our problem consists of a sequence of jobs types. To service a job type, a truck must arrive empty at a *customer*, pick up its goods, and deliver all of these goods to an *end-customer*. Thus, routes consist of (customer, end-customer, customer, end-customer,...) sequences, where for every customer, the end-customer is known. There are a number of reasons for assuming that routes have this structure. The first reason is that the transport capacity is sold to customers in full-truckload (or as-if-full-truckload) increments, and the vast majority of customers specify the minimum number of visits per week such that the transport capacity is reasonably well utilized (over 50% on the average). Second, many of the parts being transported are valuable and at risk of theft, so customers may require that the truck be sealed in transit or that the truck proceed directly to the end-customer. One example of such a part is a compact disc (CD) containing software. The manufacturing cost of the CD is negligible, but the significantly higher retail price makes it a target for thieves. Third, when using our model to estimate the cost of servicing a prospective customer, the 3PL provider may not be able to estimate in advance the opportunities for consolidation of loads that would facilitate the construction of appropriate candidate routes. This is especially true because our approach *determines* the shipment quantities rather than taking them as given. In the concluding section, we discuss how one can make heuristic adjustments to take advantage of consolidation opportunities *ex post*.

We emphasize that our motivating application precludes the need for focusing on the classical routing problem. The number of possible routes is severely constrained by geographical and practical considerations. For example, due to driver workday restrictions and the time required to load and unload goods, it is unlikely that a route would contain more than four or five (customer, end-customer) pairs. Thus, the 3PL provider can easily enumerate and determine the cost of all practical routes, and use these costs to select the best route containing any given subset of (customer, end-customer) pairs.

Under mild conditions, namely (i) if route costs are linear in distance (or time), (ii) the triangle inequality for driving distance (alternatively, time) holds, and (iii) the 3PL provider's objective is to minimize total route cost, it is optimal to service each job type as few days per week as possible (i.e., one day per week if the total weekly shipment quantity is less than or equal to one truckload, or the minimum number of trucks required to transport the load otherwise). Without additional economic incentives, the selected schedule will contain the minimal number of shipments for each customer. In order to lower their inventory holding costs, the end-customers desire as many shipments as possible. Indeed, they may desire just-in-time delivery of their daily demands. Thus, the customers may be willing to pay more to provide more frequent service for their end-customers. As a proxy for the customers' willingness to pay, we include the cost of holding inventory *at the end-customer*. Thus, we implicitly assume that end-customers are willing to reward customers for decreased holding costs, and that customers in turn are willing to reward the logistics provider. Alternatively, if the customer is providing VMI services and owns the inventory at the end-customer, the customer benefits directly from reduced inventory levels. We also impose a lower bound on the number of delivery days for each job type. Both the inventory holding costs and the lower bounds on delivery days encourage better service (i.e., smaller, more frequent deliveries). The goal is to choose the routes to execute each day (and thus implicitly the job types to service) and the delivery quantity for each job type to minimize the sum of route costs and inventory holding costs over a horizon of T periods. In doing so, we must meet demands on time at the end-customers and satisfy a lower bound on the number of delivery days for each end-customer. Of course, some of these decisions may be fixed in advance to represent the unchangeable portion of the existing schedule. A formulation follows.

Indices:

j: job type (specifies (customer, end-customer) pair)

r: truck route (specifies a sequence of stops)

t: time (day of week), $t = 1, ..., T$

Data:

c_r: total cost for executing route r

D_{jt}: demand of job type j on day t, expressed in standard units (e.g., pallets)

h_j: one-period inventory holding cost for one unit of demand for job type j

b_j: minimum number of days of service for job type j

α_{jr}: 1 if route r includes job type j; 0 otherwise

CAP: capacity of a vehicle in standard units (e.g., pallets)

Decision Variables:

z_{rt}: number of times route r is executed on day t

x_{jt}: number of units of job type j shipped on day t

v_{jt}: number of vehicles servicing job type j on day $t = \sum_r \alpha_{jr} z_{rt}$

y_{jt}: 1 if there is one or more delivery of job type j on day t; 0 otherwise (implicit decision)

I_{jt}: inventory of job type j remaining at the end-customer at the end of period t (implicit decision)

(P)

$$minimize \quad \sum_r \sum_t c_r z_{rt} + \sum_j \sum_t h_j I_{jt}$$

$s.t.$

$$\sum_t y_{jt} \geq b_j \quad \forall j \tag{1}$$

$$y_{jt} \leq v_{jt} \quad \forall j, t \tag{2}$$

$$v_{jt} \leq \sum_r \alpha_{jr} z_{rt} \quad \forall j, t \tag{3}$$

$$I_{jt} = I_{j,t-1} + x_{jt} - D_{jt} \quad \forall j, t \tag{4}$$

$$x_{jt} \leq CAP * v_{jt} \quad \forall j, t \tag{5}$$

$$y_{jt} \text{ binary} \quad \forall j, t$$

$$z_{rt}, v_{jt} \text{ non-negative integers} \quad \forall j, r, t$$

$$x_{jt}, I_{jt} \text{ non-negative} \quad \forall j, t$$

The first set of constraints ensures that each job type receives its minimum required days of service (or more). Without these constraints, a job type's "frequency of service" requirements may not be met, particularly if the end-customer's demands and holding costs are low, and the incremental cost of servicing the job type is high. The incremental cost of servicing a job type is high if the end-customer and/or the corresponding shipment origin is located far from the truck depot and/or from the other customers and end-customers. The second set of constraints ensures that a service day is credited only if the number of truck visits (for that job type) is one or greater. The third set of constraints defines the v_{jt} variables, and the fourth set of constraints represents the inventory balance equations, which preclude shortages. The fifth set of constraints limits the shipment quantity to the corresponding shipment capacity. Observe that constraints (3) are expressed as inequalities rather than equalities. Due to the structure of constraint sets (1), (2) and

(3), it is optimal for the v_{jt} values to be as large as possible; thus, constraints (3) are always satisfied as equalities. The inequality representation aids in our solution procedure, as we explain in more detail later.

The formulation above addresses the problem for a finite horizon. At the 3PL provider that motivated our work, the plan is expected to repeat periodically, usually weekly. For this reason, it is not essential for the system to start and end the week with zero inventory. We do require, however, that the plan be repeatable, and we therefore impose the constraints:

$$I_{jT} = I_{j0} \quad \forall j. \tag{6}$$

Of course, the model above does not explicitly represent all the possible complexities of real world problems. It can, however, be modified to capture at least some of these complexities, including:

- multiple truck types (the formulation is for a single truck type);
- constraints on the number of routes or the number of truck-hours available in a day (unconstrained here);
- delivery time constraints (unconstrained here; such constraints can be considered easily in the generation of routes);
- multi-day routes (single-day routes assumed here); and
- time-varying route and inventory holding costs (assumed time-invariant here).

Also, recall that we are addressing a problem in which each route consists of a series of one or more (pick-up, drop-off) operations, which reflects the usual mode of operation at the 3PL provider that motivated our research. Figure 1 shows an example of an allowable route. As a consequence of this assumption, truck capacity limits apply only to a single delivery. Of course, for some applications, goods from two or more job types may be loaded onto a truck simultaneously. Modifying our model to accommodate this problem variant would significantly increase the complexity of the model. (We discuss this issue further in the concluding section.)

INSERT FIGURE 1 HERE.

3 Literature Review

Our problem involves selecting routes for each day of the week and determining shipment quantities for each customer (within the capacity constraints defined by the selected routes) to satisfy demands that may vary by period. The latter decisions are similar to lot sizing decisions. The long history of research on deterministic *single-stage*, single-item lot sizing models begins with the seminal work of Wagner and Whitin (1958) for the uncapacitated model. Aggarwal and Park (1990), Federgruen and Tzur (1991), and Wagelmans et al. (1992) developed faster exact algorithms for the uncapacitated case. For the capacitated problem, Florian and Klein (1971) characterized the optimal solution for the case of constant capacity. Baker et al. (1978) developed algorithms for the case of time-varying capacity, and Love (1973) characterized optimal solutions when production and storage costs have a piecewise concave structure. Lippman (1969) analyzed the multiple setup cost case, where there is a fixed charge for each increment of capacity (such as one truckload). We discuss his results in more detail in Section 5.

Although we do not explicitly solve the routing problem, our problem contains features of both the *Period Vehicle Routing Problem (PVRP)* and the *Inventory Routing Problem (IRP).* The PVRP is a multi-period vehicle routing problem in which the decisions are the service day(s) for each customer and the vehicle routes for a service provider on each day. Very few PVRPs in the literature consider inventory or other costs associated with the selection of a particular day-of-week schedule. The emphasis is on minimizing routing costs and/or the number of required vehicles. The most common assumptions are that the number of visits during the horizon is fixed and that the delivery or pick-up quantity is the same for each visit. A few authors do allow for different numbers of visits and/or different delivery quantities. Russell and Gribbin (1991) allow an arbitrary allocation of the week's goods among a specified number of deliveries for each customer. Gaudioso and Paletta (1992) assume constant demand and equal delivery quantities, and impose spacing constraints between deliveries, without accounting for the cost of holding the inventory required to support such a delivery pattern while avoiding shortages. Similarly, Chao et al. (1995) explicitly account for the effect of time-varying demand and the delivery patterns on the quantities to be delivered, but do not consider inventory holding costs.

In contrast to the PVRP, IRPs more strongly emphasize the tradeoff between delivery and inventory-related costs. Typical objective functions include vehicle routing costs, inventory holding costs, and shortage costs. For articles on continuous-time problems with constant demand, see Dror and Trudeau (1996), Herer and Roundy (1997), Federgruen and Van Ryzin (1997), Viswanathan and Mathur (1997) and Chan et al. (1998), and references therein. For single-period problems with stochastic demands, see Federgruen and Zipkin (1984) and Federgruen et al. (1986). For multi-period problems with stochastic demands see Webb and Larson (1995), Herer and Levy (1997) and Bard et al. (1998) and the references therein. Finally, for continuous-time problems with stochastic demand, see Larson (1988), Dror and Trudeau (1996), and Qu et al. (1999). It should be noted that the vast majority of the multi-period problems and continuous time problems with stochastic demand assume that demand is stationary.

Several IRP papers address problems that are closely related to ours. Chien et al. (1989) consider a single-period model with deterministic demand in which the supplier has a limited quantity of the product, and the goal is to maximize revenue less transportation and shortage costs subject to supply and demand availability, and vehicle capacity constraints. Our problem is essentially a multi-period generalization of the Chien et al. model with additional costs for inventory and constraints on the delivery patterns. Dror and Levy (1986) consider a multi-period model in which the demand is constant but the required shipment quantity depends on the delivery day, and each customer is serviced no more than once during the time horizon. Bell et al. (1983) develop a model to minimize the cost of distributing industrial gases considering the forecasted inventory levels of the customers. Dror et al. (1985) study a finite-horizon problem in which each customer receives at most one delivery during the horizon and deliveries must occur before the customer is projected to deplete his supply. Dror and Levy (1986), and Dror and Trudeau (1996) consider generalizations and additional solution approaches for these models. Chandra (1993) addresses the joint problem of warehouse procurement decisions and delivery (routing) to retailers for multiple products over a finite horizon, and develops a heuristic for the problem. Chandra and Fisher (1994) examine a similar problem in which a production schedule, rather than a procurement schedule, must be decided. Metters (1996) examines the problem of coordinating delivery and sortation of mail when there are deadlines for the completion of sortation, and solves the problem using commercial optimization software. Carter et al. (1996) consider the problem of planning the delivery of multiple grocery items during multiple periods over a finite, repeating horizon. A delivery pattern must be selected for each customer, and inventory allocations and vehicle routes must be chosen on each day. Vehicle capacity, vehicle availability, route duration and delivery time window restrictions apply. They develop a heuristic procedure for solving this problem. As the size of the fleet is an important constraint in their motivating application, their procedure emphasizes smoothing vehicle use. A variant of our problem without constraints on the number of service days per week and with deliveries only (i.e., no intermediate stops for pickups) is addressed by Lee et al. (2003), who construct annealing heuristics and derive certain properties of the optimal solution for their problem.

In contrast to the vast majority of PVRP models, our model specifically accounts for effects of different delivery patterns on the inventory that must be held by the end-customer. In contrast to many IRP models, our model directly addresses a multi-period problem with time-varying demand that may need to be satisfied by more than one shipment during the horizon. Equally important is that our formulation of the problem permits an exact representation of route costs (versus a fixed cost per delivery, cf. Carter et al.), an exact representation of inventory costs incurred by the end-customer as a consequence of the delivery schedule, as well as constraints on the number of deliveries per week for each job type.

None of the articles cited above accounts for all of the factors and constraints that we consider. For this much more general and accurate representation of real-world, multi-period shipment problems involving both pickup and delivery with days-of-service constraints that 3PL providers are facing, our major contribution is developing an approach that produces near-optimal solutions relatively quickly.

It is important to highlight the complications introduced by the minimum days-of-service constraints. Our problem exhibits strong links across multiple periods, not only because of the inventory costs induced by the day-of-week delivery pattern for an individual customer, but because the incremental cost of servicing a customer and the difficulty of satisfying that customer's days-of-service constraint depend upon the delivery patterns of the other customers. Although the *economic* interactions noted above arise in the IRP and some versions of the PVRP, in those contexts, they tend to induce "soft" constraints that can often be negotiated via earlier shipments (and the associated inventory costs). On the other hand, the "hard" days-of-service constraints in our problem create structural linkages that cause the (general) integral route selection decisions to play a stronger role in the solution of our problem.

4 Problem Variants

In addition to (P) formulated in Section 2, we examine two restricted versions that may be applicable in many problem environments. These restricted problems are not only realistic but can be solved using the same solution framework as that described in the next section and with less computational effort. As our discussion proceeds, we will explain why the problems are easier to solve. Here, we present the motivation for the restrictions and the related changes in the problem formulations.

Variant 1:

In the first problem variant, we impose the constraint that each job type receives at most one visit per day. In this case, at most a partial truckload could be shipped ahead of schedule on a given day. Such a constraint would be imposed in practice if the customer insists on a low-inventory, almost-just-in-time solution.

The formulation changes as follows:

• The z_{rt} and v_{jt} variables are now binary.

• The y_{jt} variables are now equivalent to the v_{jt} variables and can be removed from the formulation by substituting v_{jt} wherever y_{jt} appears and removing redundant constraints (e.g., (2)).

• We add the constraint

$$\sum_r \alpha_{jr} z_{rt} \leq 1 \quad \forall j, t.$$

If the demand on a single day exceeds one truckload, then the job type can be replaced by multiple ("dummy") job types with the same physical origin and destination, where each of these job types has demand of up to one truckload per day. (Of course, it is most economical to subdivide the goods into as few truckloads as possible.) In this case, for each job type j whose demand exceeds a truckload on day t, we define J_{jt} as the set of corresponding "dummy" job types and rewrite the constraint that defines visits as:

$$y_{jt} \leq \sum_{i \in J_{jt}} \sum_r a_{ir} z_{rt}$$

Although this constraint has a different form than the constraints in the original formulation, it has the same structure as certain constraints in the formulation on which our solution procedure is based. Consequently, we can easily handle demands exceeding a truckload for any job type on any day.

Variant 2:

The second problem variant does not restrict the number of visits for each job type; it simply permits us to execute each route at most once on each day. The motivation for this constraint is the very small likelihood of needing, much less choosing, the same route more than once on the same day. Such a need would arise only if several job types that could comprise a relatively efficient route could all benefit from receiving more than one truckload of good on the same day.

The only required change in the formulation is to make the z_{rt} variables binary.

5 Solution Approach

The set partitioning problem, an NP-hard problem (Garfinkel and Nemhauser 1969), is a special case of (P), which implies that (P) is NP-hard. To see this, consider the special case of our problem in which inventory costs are ignored and inventory non-negativity constraints are not enforced except at the end of the horizon. In this case, it is optimal to service each job type with as few vehicles as possible, and without regard to the day of the week, so each "week" can be regarded as a single time period. Each job type requiring more than one truckload in a week is replaced by an appropriate number of "dummy" jobs, in the same way as in Variant 1. The z_{rt} become binary rather than general integer variables. The routes are defined for the set of "dummy" job types and the a_{jr} values are defined accordingly. With these redefinitions and appropriate simplifications of the objective function and constraints, the problem reduces to the set partitioning representation of the single-period (standard) vehicle routing problem.

The inventory non-negativity constraints and inventory costs link the periods, thereby creating a problem that is more difficult than a set partitioning problem. Indeed, as we report in more detail later, even small instances of (P) are impractical to solve using commercial software. This necessitates the development of a solution procedure that takes advantage of the structure of our problem. We first present a solution approach for the general problem, and then explain how the procedure should be modified for the two problem variants.

We propose a Lagrangian approach to the problem in which constraints (2) are relaxed using multipliers μ_{jt} (≥ 0) and constraints (3) are relaxed using multipliers λ_{jt} (≥ 0). Observe that expressing constraints (3) as inequalities allows us to dualize them using non-negative Lagrange multipliers. This, in turn, provides for a more meaningful interpretation of the λ_{jt} values, and, as we observed in preliminary computational studies, a more stable solution procedure.

Because we have relaxed constraints (2), the introduction of constraints (7) and (8), shown below, which are redundant in the original problem, provides a stronger formulation for the Lagrangian procedure. Constraints (7) ensure that a service day is credited only if at least one appropriate route is selected, while constraints (8) ensure that the total number of truck visits is large enough to satisfy each job type's demand.

$$y_{jt} \leq \sum_{r} \alpha_{jr} z_{rt} \quad \forall j, t \tag{7}$$

$$\sum_{r} \sum_{t} \alpha_{jr} z_{rt} \geq \lceil CAP^{-1} \sum_{t} D_{jt} \rceil \quad \forall j \tag{8}$$

Relaxing constraints (2) and (3) and adding constraints (7) and (8) yields two subproblems for fixed λ_{jt} and μ_{jt} values:

(P1)

$$minimize \quad \sum_{r} \sum_{t} (c_r - \sum_{j} \lambda_{jt} \alpha_{jr}) z_{rt} + \sum_{j} \sum_{t} \mu_{jt} y_{jt}$$

$s.t.$

$$\sum_{t} y_{jt} \geq b_j \quad \forall j$$

$$y_{jt} \leq \sum_{r} \alpha_{jr} z_{rt} \quad \forall j, t$$

$$\sum_{r} \sum_{t} \alpha_{jr} z_{rt} \geq \lceil CAP^{-1} \sum_{t} D_{jt} \rceil \quad \forall j$$

$$y_{jt} \text{ binary} \quad \forall j, t$$

$$z_{rt} \text{ non-negative integers} \quad \forall r, t$$

and (P2)

$$minimize \quad \sum_{j} \sum_{t} h_j I_{jt} + \sum_{j} \sum_{t} (\lambda_{jt} - \mu_{jt}) v_{jt}$$

$s.t.$

$$I_{jt} = I_{j,t-1} + x_{jt} - D_{jt} \quad \forall j, t$$

$$x_{jt} \le CAP * v_{jt} \quad \forall j, t$$

$$v_{jt} \text{ non-negative integers} \quad \forall j, t$$

$$x_{jt}, I_{jt} \text{ non-negative} \quad \forall j, t$$

Subproblem (P1), the routing subproblem, is a variant of the PVRP with lower bounds on the number of service days and on the number of truck visits over the horizon for each job type. In the objective function, there is an adjusted cost for each route that accounts for the value of that route's shipping capacity in meeting the needs of customers on that route on that day, as well as additional "costs" corresponding to satisfying the service-day requirements of the customers. The "standard" PVRP includes only terms containing the z_{rt} variables, and consequently, is much simpler. Problem (P1) is NP-hard for the same reasons as (P), via the same reduction.

Observe that without the added valid inequalities (7) and (8), the optimal solution to (P1) would be to identify, for each j, the b_j smallest values of μ_{jt} and to set the corresponding values of y_{jt} to 1 (otherwise set y_{jt} to zero) and to set z_{rt} to 1 if its coefficient (reduced cost) in the objective function is negative (otherwise set z_{rt} to zero). Such a solution does not ensure consistency between the selected routes and the days of service, and provides no guarantee that the number of routes is sufficient to handle each job type's weekly demand. Consequently, the "solution" is far from being feasible and its cost is a very loose bound on the actual cost. The inclusion of constraints (7) and (8) provides substantially better bounds, primarily because the resultant routing solution is feasible: the constraints on number of service days are satisfied and there are sufficient routes to accommodate the required freight flows.

Subproblem (P2), the shipment scheduling subproblem, is a capacitated lot sizing problem with multiple setup costs, one for each capacitated vehicle. Because it is a subproblem in a relaxation with a form that allows some of the (adjusted) setup costs to be negative, when viewed as a stand-alone problem, (P2) could have an unbounded objective. However, in our problem context, despite these negative setup costs, we must devise a method that provides a strong bound in order to solve the original problem. It is from this vantage point that we analyze (P2) and develop an optimal polynomial-time solution procedure for it.

5.1 Analysis of (P2)

The second subproblem is separable by job type. Thus, in the remainder of this section, we consider the problem for a single job type and omit the job type subscript. Lippman (1969) studies a class of multiple setup cost problems that includes ours as a special case. He shows that there exists an optimal solution consisting of regeneration intervals. (A regeneration interval is a set of consecutive periods with zero initial and terminal inventory and with all intermediate periods having positive inventory.) Thus, the strategy is to find the optimal solution for each potential regeneration interval, then to find the best combination of regeneration intervals using a shortest path algorithm.

Lippman also shows that there exists an optimal solution such that:

$$I_{t-1}(x_t \mod CAP) = 0.$$

In other words, in each period, either entering inventory is zero or the shipment quantity is a multiple of a full truckload (or both). Although not explicitly stated in his paper, a further implication of this result is that *within a regeneration interval*, only the first period can have a partial-truckload shipment, as all of the remaining periods must have $I_{t-1} > 0$. As such, if all trucks have the same capacity (as in our problem), we know exactly how many vehicles are sent within the regeneration interval.

Lippman's result is based on the assumption that the setup costs are non-negative, and his result characterizes the shipment quantities but does not explicitly state how many trucks should be sent. Of course, when setup costs are positive, it is optimal to send as few trucks as possible to accommodate the shipment quantity in each period. Lee (1989) and Anily and Tzur (2002), among others, have studied variants of this lot-sizing problem in which multiple capacitated shipments (of arbitrary quantities) are allowed in each period, but all of these models have the implicit assumption of positive setup costs. Pochet and Wolsey (1993) study the special (restrictive) case in which the batch size must be some integer multiple of some basic batch size, but they, too, assume that setup costs must be positive.

Our second subproblem has the unusual and distinctive characteristic that some of the (adjusted) setup costs may be negative, and it is this characteristic that necessitates a different solution approach. The approaches in the literature cannot be applied directly to our problem because they do not allow for the combination of negative and time-varying setup costs. Both of these aspects arise in our second subproblem.

In our problem, it may be optimal to send extra trucks, including some that are completely empty. To avoid an unbounded solution, we impose the constraint

$$v_t \leq \lceil \sum_t D_t / CAP \rceil, \quad \forall t$$

which simply limits the number of trucks in any period to the number that would be required to service all of the demand in a single period. Let $\bar{v} = \lceil \sum_t D_t / CAP \rceil$. (We later obtain stronger bounds on v, but for the purposes of our present analysis, this particular upper bound is useful.) It is clear that $v_t^* = \bar{v}$ for periods in which the corresponding coefficients are negative. The problem is now to determine how to use this "free capacity" and how to make shipments in the remaining periods. We show that this modified problem (with constraints on v_t) has the same property as that derived by Lippman. We then show how to construct an optimal solution (both the truck schedule and the shipment quantities) for this problem. For ease of exposition, let K_t denote the coefficient associated with v_t.

Proposition 1: For a setup cost structure of the form $K_t v_t$ where some of the K_t values may be negative, the optimal solution satisfies:

$$I_{t-1}(x_t \quad mod \quad CAP) = 0.$$

Proof: Suppose, to the contrary, that we have an optimal solution, x_t^*, in which $I_{t-1} > 0$ and $x_t^* \quad mod \quad CAP > 0$. The latter condition implies there is excess transportation capacity in period t. There exists some $\epsilon = min\{CAP - (x_t^* \quad mod \quad CAP), I_{t-1}\} > 0$ such that we can ship, in period t, ϵ units that had been shipped in some prior period, at a minimum savings of $h\epsilon$. This contradicts the optimality of the original solution. □

Observe that the possibility of trucks being sent empty does not change the implication of Proposition 1 with respect to the timing of a partially-filled truck. Thus, only the first period in a regeneration interval can have a partial truck shipment. The complication in our problem is that we do not know in advance how many trucks will be sent in a regeneration interval. However, we do know how many non-empty trucks will be sent.

Consider a solution for a regeneration interval consisting of periods a through b, constructed as follows:

Algorithm A2:

Step 1. For $t = a, ..., b$, set

$$x_t = \sum_{k=a}^{b} D_k - CAP\lfloor(\sum_{k=t+1}^{b} D_k/CAP)\rfloor - \sum_{k=a}^{t-1} x_k$$

Step 2. For $t = a+1, ..., b$,

$$n^* = \arg\max_{a \le n \le t}\{K_t - CAP * h * (t-n) - K_n\}$$

$$S = \max_{a \le n < t}\{K_t - CAP * h * (t-n) - K_n\}$$

If $S > 0$

set $x_{n^*} = x_{n^*} + x_t$

set $x_t = 0$

Step 3. For $t = a, ..., b$, set

$$v_t = \lceil x_t/CAP \rceil$$

Step 4. For $t = a, ..., b$,

if $K_t < 0$ and $v_t < \bar{v}$, set $v_t = \bar{v}$.

In Step 1, the shipment quantity is set so that the fractional truckload is shipped in the first period and beyond this, just enough full truckloads are shipped in each period so that demand is satisfied on time. This tentative solution can be viewed as the schedule that is as close to just-in-time as possible while retaining properties of the optimal solution.

In Step 2, we determine, for each period, the best earlier period into which we could shift full truckloads. If the savings is positive, we shift all relevant truckloads. In Step 3, we set the truck variable v_t equal to the minimum number of trucks necessary to ship the quantity x_t. Finally, in Step 4, we identify periods with $K_t < 0$ in which the maximum number of trucks is not yet assigned, and set the corresponding v_t values to their upper bounds. Note that Steps 2, 3, and 4 retain properties of the optimal solution.

Proposition 2: Algorithm A2 produces an optimal solution (x_t^* and v_t^*) for a given regeneration interval.

Proof: We first note that by Proposition 1, in some optimal solution for the regeneration interval, all shipments except possibly that in the first period of the interval are integer multiples of truck capacity. In this proof, we restrict our attention to schedules for which the property in Proposition 1 holds. Thus, Step

1 of the algorithm assigns shipments so that the minimum number of non-empty trucks, and indeed, the *only possible number of non-empty trucks* under the property in Proposition 1, are scheduled. Moreover, the schedule constructed in Step 1 is such that each full truckload is scheduled as late as possible. Thus, the only feasible changes entail moving full truckloads to earlier periods. Recall that the maximum number of trucks in each period ($\bar{v}$) is sufficient to ship the entire demand during the regeneration interval. Thus, for any moves of full truckloads to earlier periods, we can consider the best time to dispatch each individual non-empty truckload independently. Now, by construction of the algorithm, we cannot move an entire truckload from one period to another while reducing costs (cf. Step 2). Had it been possible to reduce costs by removing a truck with $K_t > 0$ and shifting the load into a period with $K_t \leq 0$, that shift would have been implemented in Step 2. Furthermore, by Step 4 of the algorithm, we cannot reduce the total cost by adding empty trucks to the solution. Thus, Algorithm A2 produces an optimal solution for the regeneration interval. □

We note that regeneration intervals can be considered independently. Thus, the optimal solution for each possible regeneration interval can be determined using this algorithm, and a shortest path algorithm can be used to select the optimal set of regeneration intervals.

Observe that Steps 1, 3 and 4 have linear time complexity and Step 2 has $O(T^2)$ complexity. Thus the computation of the costs for the $O(T^2)$ regeneration intervals has $O(T^4)$ complexity. The shortest path problem (to find the best combination of regeneration intervals) has $O(T^2)$ complexity. Consequently, the overall procedure has $O(T^4)$ complexity. Observe, however, that the computations and comparisons are extremely simple.

We note that for any time interval that could be covered by a single regeneration interval, there may be an alternate dominant solution consisting of a set of shorter regeneration intervals. Such dominant solutions are identified by the shortest path procedure. Thus, it is only necessary for our procedure to find an optimal solution that satisfies the regeneration interval property for the time interval under consideration. Among all such optimal solutions, we restrict our search to those satisfying an established property of the optimal solution within a regeneration interval. By doing so, we are able to construct a very efficient solution procedure for (P2).

Because of its structure, (P2) is easy to solve and produces a much stronger bound than its LP relaxation, and thereby provides a significant contribution to strengthening the overall lower bound. In addition, as we will see later, including the fixed charge costs (via the dual variables) and the vehicle capacity constraints in this subproblem also provides relatively fast convergence of the Lagrangian procedure because the solutions of (P1) and (P2) are more consistent than they would be without these considerations in (P2).

5.2 Solution Procedure for (P)

Recall that in order to solve (P), we propose a Lagrangian approach in which we relax two sets of constraints – those ensuring that each service day is correctly accounted for, and those defining the v_{jt} variables. Correspondingly, we associate a set of multipliers with each set of relaxed constraints. We employ variants of the subgradient optimization method to update the multipliers at each iteration. (See the Appendix for details.) For each set of multipliers, we first solve (P1) and (P2) optimally to provide a lower bound, which we update if it has improved. We then construct a feasible solution by using the z^*_{rt} values from (P1), computing

$$v_{jt} = \sum_r \alpha_{jr} z^*_{rt} \quad \forall j, t$$

and substituting the values of v_{jt} (as fixed quantities) in (P2). We then solve (P2), which is a linear program when the v_{jt} values are fixed.

We observed that solutions constructed by the method described above often result in excess truck movements, i.e., the v_{jt} values are larger than necessary to handle the resulting shipment quantities. Therefore, we also construct another feasible solution by taking the solution for (P2) and checking its feasibility with respect to the customer service constraints. If the solution satisfies these constraints, for each day of the week, we solve the associated problem (P1). Of course, if the solution for (P1) exactly satisfies the shipping requirements from (P2), the solution is optimal and there is no need to re-solve the routing subproblem. Although we could make incremental changes to the routes from (P1) (i.e., eliminate excess stops), in exploratory tests, such a method did not consistently produce good solutions.

We compute the objective value of the feasible solution and update the upper bound if the newly-constructed solution has a better objective than the current upper bound. The multipliers are then updated and the process is repeated until optimality is achieved, or the best feasible solution is within some tolerance of the lower bound, or until the step size reaches virtually zero (thereby precluding any significant improvement in the objective function value).

Because we allow a repeating schedule rather than restricting the solution to one with zero initial and ending inventories, constraints (8), which are retained in (P1), are sufficient to ensure that the z_{rt} values from (P1) will yield a feasible solution for (P2). If one is solving a finite horizon problem, then it may be necessary to impose lower bound constraints on $\sum_{k=1}^{t} \sum_{r} \alpha_{jr} z_{rk}$ for $t = 1, ..., T$ and for all j to ensure that the timing of trucks allows for a feasible solution of (P2) with the v_{jt} values implied by (P1).

5.3 Modification for Problem Variants

Variant 1:

In this case, we only need to impose the tighter of the customer service (number of visit days) constraint or the constraint related to total delivery capacity for each job type j. Noting that there exists an optimal solution such that $v_{jt} = \sum_{r} \alpha_{jr} z_{rt}$, we can eliminate the v_{jt} variables entirely. With these simplifications, (P1) becomes

(P1′)

$$minimize \ \sum_{r} \sum_{t} (c_r - \sum_{j} \lambda_{jt} \alpha_{jr}) z_{rt}$$

s.t.

$$\sum_{r} \sum_{t} \alpha_{jr} z_{rt} \geq \ \max \{b_j, \lceil CAP^{-1} \sum_{t} D_{jt} \rceil\} \quad \forall j$$

$$\sum_{r} \alpha_{jr} z_{rt} \leq 1 \quad \forall j, t$$

$$z_{rt} \text{ binary} \quad \forall r, t$$

The only change in (P2) is that the v_{jt} variables are now binary, so the problem becomes a lot sizing problem with standard (binary) setups. Recall, however, that because (P2) is derived from a relaxation, the setup costs may be negative.

Variant 2:

In this case, the only changes are that the z_{rt} variables are binary. The same solution procedure can be used, but the constraint space is much smaller. Next, we discuss ways to further limit the search space.

5.4 Bounding the v_{jt} Values

In this subsection, we describe methods to obtain upper and lower bounds on the values of v_{jt} and consequently also on $\sum_r \alpha_{jr} z_{rt}$ for the original problem and for Variant 2. The upper bounds on $\sum_r \alpha_{jr} z_{rt}$ also have obvious implications for the individual z_{rt} values in the general model where the z_{rt} values may be greater than 1. These bounds and their derivations are intuitive and we state them without proof.

Simple Bounds

The simple bounds are based on the observation that for customer j, service must occur on at least b_j days. Thus, the most that one would ship on a single day is the sum of demands on the $T - (b_j - 1)$ days with the greatest demands, as the demands on all of the other days would be shipped "just-in-time." This bound may be quite loose, but it is easy to compute and does not differ by day of week.

Cost-Based Bounds

The cost-based bounds recognize the economic tradeoffs between the "setup" (transportation) and holding costs. We can derive an upper bound on the number of times job type j is serviced on day t as follows: First, we let the setup cost on day t be equal to a lower bound on the smallest incremental cost of servicing job j, which can be determined by finding the least expensive way to insert job type j into any (already-generated) executable route. Then, we let the setup cost for all other days be equal to an upper bound, for example, that derived from serving job type j alone. (Note that this route is always feasible.) With these bounds on setup costs, we solve the associated lot sizing problem with multiple setups. The number of trucks on day t in the solution of this problem is a tentative upper bound on $v_{jt} = \sum_r \alpha_{jr} z_{rt}$. In other words, we would not service job type j on day t any more times than if transportation costs were as cheap as possible on day t and as expensive as possible on the other days. This would be a valid bound if we did not have a constraint on the number of service days. To account for this constraint, we note that if the initial upper bound is equal to zero, it is economically unfavorable to service that job type on that day. To incur the minimum penalty while contributing to the days of service, we would service that job type at most once on that day. Thus, we can, without loss of optimality, set the upper bound on v_{jt} equal to 1 in this case.

Similarly, we can find lower bounds on the v_{jt} values by setting the setup cost on day t equal to an upper bound and the cost on the other days equal to a lower bound and solving the same lot sizing problem. In this case, the days-of-service constraint does not necessitate any adjustment.

Other Bounds

Observe that an upper bound on v_{jt} can be used as an upper bound on all z_{rt} such that $\alpha_{jr} = 1$. Such bounds may be useful when the upper bound on v_{jt} is small (e.g., 1) and the corresponding z_{rt} values would otherwise be constrained only by much larger values obtained from the simple bounds described above.

The analysis can be taken a step further by noting that another upper bound on z_{rt} is:

$$\min_j \{\text{upper bounds on } v_{jt} \text{ such that } \alpha_{jr} = 1\}.$$

In other words, the maximum number of times we would select a route is the minimum among the upper bounds on v_{jt} for the locations on the route. Bounds of this type may be useful when many locations have small upper bounds on v_{jt}. We did not implement this type of bound because of the computational effort required for the large number of routes in our problems.

6 Computational Results

We perform a series of computational tests in order to evaluate the effectiveness of our algorithm on the original problem and on the two problem variants. Before describing our computational study, it is important to point out that preliminary computational tests showed that both (i) the bounds described in the previous section and (ii) constraint sets (7) and (8) that are redundant in (P) but not redundant in (P1) are critical in finding solutions quickly. Without them, our procedure is not efficient, and the standard implementation of CPLEX applied to (P) is rarely able to find feasible solutions, even for problems of modest size. We report results in which both solution approaches, i.e., our Lagrangian approach and applying CPLEX to (P), are afforded the benefits from these additional valid inequalities. We next detail problem generation, and then discuss results.

6.1 Problem Generation and Execution of Algorithms

We generate a variety of problem instances for our computational study. Each problem instance has a five day time horizon. For each customer, the minimum number of days of service in a week is randomly generated from a U[1,3] distribution. We chose this range for the minimum days of service because problems with a four-day minimum service requirement are easier to solve and a five-day delivery requirement eliminates day-of-week decisions altogether.

The depot is located at the center of a 100 “mile” x 100 “mile” area. The location of each customer and end-customer is determined by randomly generating (i.i.d.) horizontal and vertical coordinates from a U[0,100] distribution. All customer locations are thus distinct, so there is a one-to-one relationship between customers and job types. Once customer and end-customer locations are generated, we use Euclidean distances and assume that transportation costs are linear in the travel distance (normalized to $1 per “mile,” which is roughly equal to the true variable cost for many 3PL providers). All trucks are assumed to have a capacity of 20 units (e.g., pallets).

Each customer’s daily inventory holding cost per unit is generated from a U[0.5,5] distribution. This range of inventory holding costs corresponds to goods whose value may be as much as approximately $20,000 per (full) truckload. For such goods, less-than-daily delivery may be warranted, necessitating day-of-week decisions.

To generate candidate routes, we initially generated all combinations of 1, 2, 3, or 4 job types. In typical applications involving the transport of components to manufacturers and finished goods from distribution centers to retailers, the combination of transit times between customers and end-customers and enroute loading and unloading time limits the number of customers that a single vehicle can service to about 4 job types in a typical work shift, especially in congested urban areas.

Before solving the traveling salesman problem (TSP) for each combination, we apply a filter which eliminates those combinations for which a very loose lower bound on total route time exceeds a 7.5 hour work-day. We assume an average driving speed of 40 miles per hour (which is similar to the value used by regional delivery companies in major metropolitan areas), and a total of 30 minutes for loading, unloading and waiting time associated with one delivery. For those combinations that pass the filter, we solve the TSP (by enumeration) and eliminate the combination if the route time for the optimal TSP solution exceeds the 7.5 hour threshold, or retain the best TSP routing if the route time is below the threshold. However, we retain all single-job-type routes, even if they exceed the route time threshold, to ensure that a feasible solution exists.

We solve one set of (“small”) problems with 25 customers and their respective end-customers. The problem sizes in this set are limited in order to allow us to compare our heuristic solutions with those obtained from commercial software. The second set contains (“large”) problems with 50 customers and their respective end-customers. All computations are performed on a Sunblade 1000 with 1 GB RAM. CPLEX

7.0 is utilized for (P) and its variants, as well as for the subproblems in the Lagrangian procedure. We use AMPL as the matrix generator for the solver, and as the scripting language for the Lagrangian procedure. In all instances, we record only the *solve* time required.

For each problem instance, we solve the original version of the problem (P), as well as Variants 1 and 2. We first generate bounds on the v_{jt} values, as described in Section 5.4, to be applied in the original version and in Variant 2. We execute our Lagrangian procedure as well as CPLEX applied to each variant of (P), utilizing all relevant bounds on the v_{jt} values. We employ an optimality tolerance of 2% for both procedures, and, because preliminary results indicate that the quality of the CPLEX solutions for the original problem and for Variant 2 do not improve significantly after several hours, we impose a time limit of 4 hours on both the 25-customer and the 50-customer problems. The four hour time limit allows for a reasonable tradeoff in both solution procedures between optimality and solution time. In executing the Lagrangian procedure, we terminate it when either the optimality tolerance or the time limit is reached, or, additionally, when the step size becomes zero to within the precision of the computer (precluding significant improvement in the objective function value), whichever comes first. (See the Appendix for details of CPLEX parameter settings and numerical implementation issues.)

We discuss parameters specific to the problem sets, along with computational results, below.

6.2 25-Customer Problems

For this set of 10 problems, we generate demand for each customer and each day from a truncated Normal ($\mu = 10, \sigma = 3$) distribution, rounded to the nearest integer. For these problems, it is unlikely that more than one vehicle will visit a customer on a given day in a good solution (i.e., it is unlikely that $v_{jt} > 1$). Consequently, even Variant 1 would not be overly constraining for these problems. For this set of problems, after applying our filter, the number of remaining routes is about 7,000, corresponding to between 8,000 and 34,000 binary variables for Variants 1 and 2 (and a few hundred integer variables for Variant 2), and a corresponding number of *integer* variables for the original problem. The problems contain about 500 constraints.

In Table 1, we report the objective values for the problem variants and for the two solution procedures, along with the corresponding optimality gaps (best objective from procedure/lower bound from procedure - 1). Where optimality gaps are not reported, the gap is less than 2%. For Variant 1, both procedures solve

Prob.	Variant 1		Variant 2		Original Problem (P)		Benchmark
No.	CPLEX	Lagrangian	CPLEX	Lagrangian	CPLEX	Lagrangian	
Objective Function Values (% Gap)							
1	8302 †	8353	8483 (5.0%)	8351	8390 (3.9%)	8298 (4.0%)	10701
2	7778	7816	7842 (5.6%)	7782	8001 (7.4%)	7793 (8.3%)	8962
3	7581	7574	7889 (10.2%)	7588	7838 (9.6%)	7617 (7.8%)	9282
4	9015	9054	9757 (14.7%)	9074	9710 (14.3%)	9054 (6.2%)	11621
5	7535	7587	7672 (7.3%)	7582	7782 (8.7%)	7577 (7.7%)	9263
6	7091	7126	7298 (9.3%)	7149	7366 (10.1%)	7160 (7.4%)	8727
7	7011	7049	7172 (8.1%)	6979	7138 (7.6%)	6979 (4.5%)	8864
8	8244	8336 (2.3%)	8509 (10.4%)	8274	8422 (9.5%)	8287 (7.6%)	10348
9	8288	8291	8587 (11.4%)	8303	8825 (13.7%)	8264 (5.3%)	9869
10	6792	6829	6860 (5.8%)	6827	6878 (6.0%)	6806 (5.0%)	8526
Solution Times (seconds)							
1	130	312	*	164	*	421	
2	1300	599	*	925	*	1021	
3	860	176	*	225	*	660	
4	47	21	*	103	*	205	
5	1500	570	*	294	*	571	
6	2600	324	*	1371	*	1033	
7	1000	825	*	406	*	1183	
8	980	874	*	301	*	472	
9	310	189	*	79	*	284	
10	3100	2164	*	604	*	1381	

† If gap percentage is not reported, gap is less than 2%.

* Indicates that CPLEX reached the time limit of 4 hours.

Table 1: Objective values, corresponding gaps and solution times for 25-customer problems

all 10 problems to within approximately 2% of their respective lower bounds fairly quickly.

Variant 2 and the original problem are more difficult to solve. For Variant 2, the Lagrangian procedure identifies solutions within 2% of their respective lower bounds for all 10 scenarios in less than 25 minutes of CPU time, whereas after 4 hours of computing, CPLEX applied to (P) identifies solutions with optimality gaps of between 5% and 15%. For the original problem, the Lagrangian procedure identifies solutions within 5% of their respective lower bounds for 3 of the problems and within 8.3% for all 10 problems before the step size becomes virtually zero. CPLEX applied to (P) identifies a solution within 5% of its lower bound for only one problem, within 10% of its lower bound for 7 of the 10 problems, and within 14.3% for the remaining problems. Although the Lagrangian solutions have optimality gaps of up to 2% for Variant 2 and up to 8.3% for the original version of the problem, for each of the 10 problems, the Lagrangian procedure finds a solution that ranges from one-half of one percent to more than 7% percent better than the solution identified by CPLEX applied to (P), with an average improvement of approximately 2-3%. In addition to providing better solutions, the Lagrangian procedure consumes, on average, less than 5% of the CPU time for our (fine-tuned) CPLEX implementation on the corresponding instances of Variant 2 and the original problem.

For the distribution from which we generated demands, the $v_{jt} \leq 1$ and $z_{rt} \leq 1$ constraints present in Variant 1 are unlikely to affect the optimal solution. The imposition of these constraints reduces the search space so the Variant 1 problems require considerably less CPU time than the other problem variants; thus, Variant 1 can be solved with a straightforward implementation of CPLEX applied to (P). Thus, where it is reasonable to assume that $v_{jt} \leq 1$ in an optimal solution, the application of Variant 1 appears to be a practical alternative.

A typical 3PL provider would not explicitly consider the consequences of its selected route schedule on the cost of holding inventory at the end-customer. But the 3PL provider has the opportunity to offer tangible value to the party who bears the cost of holding inventory by offering more frequent service, and this value can translate into additional revenue for the 3PL provider. For this reason, we are interested in the consequences of ignoring inventory holding costs. To make this assessment, we use CPLEX applied to (P) (although, in principle, we could have used either procedure) to solve Variant 1 with the inventory holding costs set to zero. This version of the model is identical to the standard PVRP with the usual assumptions that (i) at most one visit is made to each end-customer (drop-off location) each day and thus also (ii) each route is executed at most once each day. This models the situation faced by the "traditional" 3PL provider, who optimizes his own costs (while ignoring those of his customers). To tabulate the full cost of this solution, we add the consequent inventory costs. We treat this total cost as a benchmark which we then compare to optimal or near-optimal solutions to estimate the system-wide benefit of explicitly considering customer inventory costs. Observe that whether or not the 3PL provider explicitly considers end-customer inventory holding costs when determining a delivery schedule, the end-customers *will* incur these costs directly, in addition to indirectly incurring the cost of transportation from the 3PL provider.

The benchmark objective values are about 15% to 30% greater than the corresponding values from

Variant 1 (the most constrained version of the problem), suggesting that accounting for inventory costs leads to significantly better solutions if the 3PL provider is currently ignoring inventory costs. Even if the 3PL provider considers inventory costs indirectly by offering customers the possibility of better service at a higher price, there may still be opportunity from using more accurate "value pricing," particularly for customers with expensive goods. In view of the thin profit margins in the trucking industry, even a portion of a 15% to 30% gap is likely to be large enough to have a significant effect on the bottom line.

6.3 50-Customer Problems

Our primary reason for solving 50-customer problems is to demonstrate that problems of the sizes observed in some practical applications can be solved by our procedure. Large logistics providers typically subdivide their customers into geographical districts, and/or according to the type of vehicle required, e.g., standard, refrigerated, extra shock protection (used for sensitive electronic goods), small vehicles (for narrow roads or hilly terrains). The corresponding problems would then also be separable by these job type categories.

Recall that one of our goals in constructing the Lagrangian procedure is to develop a viable method of solving problems for which the usual PVRP assumption of $v_{jt} \leq 1$ might be unnecessarily restrictive. Because only highly correlated demands among customers on relatively "efficient" routes (with little deadheading) would lead to $z_{rt} > 1$, we generate customer demands in such a way that we could test Variant 2 and our original problem for cases with some individual demands exceeding a truckload. We solve 10 problems with 50 (customer, end-customer) pairs. Demands are generated from a truncated Normal ($\mu = 20, \sigma = 5$) distribution, rounded to the nearest integer.

After applying the route filter, these problem instances contain about 60,000 routes. In general, these problems contain between 250,000 and 600,000 binary variables (and a few hundred integer variables) for Variant 2, and a corresponding number of *integer* variables for the original version of the problem. The problems contain between 1000 and 1500 constraints, on average.

Results for the 50-customer problems appear in Table 2. Where optimality gaps are not reported, the gap is less than 2%. CPLEX applied to (P) fails to find a feasible solution within 4 hours of CPU time for 9 out of the 10 problem instances (for both Variant 2 and the original problem). On the other hand, the Lagrangian procedure identifies a solution within 2% of optimality in 9 of the 10 cases for Variant 2, and in the tenth case, the optimality gap is only 2.3%. The Lagrangian procedure also identifies solutions within 2%

	Variant 2				Original Problem P			
	CPLEX		Lagrangian		CPLEX		Lagrangian	
Prob. No.	obj. value (% gap)	time (sec.)	obj. value (% gap)	time (sec.)	obj. value (% gap)	time (sec.)	obj. value (% gap)	time (sec.)
1	—	*	21106†	10700	—	*	21145 (7.9%)	*
2	23876 (5.7%)	*	23673	4760	24121 (6.7%)	*	23612 (5.5%)	*
3	—	*	23770	13800	—	*	26640(18.0%)	*
4	—	*	22300	12400	—	*	21038	9660
5	—	*	21455 (2.3%)	*	—	*	22480 (14.7%)	*
6	—	*	20323	11900	—	*	20327 (7.2%)	*
7	—	*	22526	7410	—	*	21499	7540
8	—	*	22535	13400	—	*	22129 (6.1%)	*
9	—	*	18432	2000	—	*	18432	3220
10	—	*	23443	5910	—	*	23397 (6.0%)	*

* Indicates that the procedure reached the time limit of 4 hours.

† If gap percentage is not reported, gap is less than 2%.

Table 2: Objective values, corresponding gaps and solution times for 50-customer problems

of optimality for 3 of the 10 cases of the original problem. In the remaining 7 cases of the original problem, the Lagrangian procedure provides solutions that are generally within 8% of the corresponding lower bound, but the gaps range up to 18%. We allowed the Lagrangian procedure to run to termination (i.e., until the step size equals virtually zero) for the two problems that have large (> 10%) gaps at the 4 hour time limit, and found that at termination, solutions within 7% of the respective lower bounds were achieved.

Solving the original problem with 50 customers is evidently quite difficult, but the Lagrangian procedure reliably finds what appear to be good feasible solutions, while, for the vast majority of problem instances, CPLEX applied to (P) is unable to find any feasible solution.

A portion of the difficulty of solving 50-customer problems is due to the large number of routes. In practical applications with several dozen job types, a 3PL provider would rarely consider including 50,000+ routes as we have done in our computational study. In practice, routes would be eliminated due to factors other than route time, so the usable set would be much smaller. Consequently, problems with more job types and proportionally fewer routes are within the range of what could be solved in practice.

Although the Lagrangian procedure produces excellent solutions in most cases, it could also be used to provide strong bounds in a branch-and-bound framework if one desired to use an enumerative procedure to find better solutions. The Lagrangian procedure could be executed differentially at various nodes in the branch-and-bound tree to take best advantage of its flexibility.

Overall, the Lagrangian procedure appears to be a promising approach, especially for solving these difficult problems in which v_{jt} may exceed 1, where there are strong interactions among the decisions across both locations and time periods.

7 Summary and Conclusions

We have modeled a multi-customer, multi-period delivery scheduling problem faced by a third-party logistics provider in which routes must be selected, and delivery quantities must be decided while satisfying constraints on the number of customer visits during a specified horizon. We have developed a solution procedure based on Lagrangian relaxation to minimize the total cost of transportation and inventory.

In this paper, we focus on the route selection and delivery quantity decisions because a typical 3PL provider has little difficulty generating a practical set of candidate routes, taking into account the structure of the road network, traffic patterns, etc. Our primary concern was to find an effective solution procedure given a good set of candidate routes. We constructed a relaxation that provides strong bounds, owing largely to the combination of the following: (i) the identification of additional valid inequalities that "tighten" the formulation and the relaxation, (ii) the economic structure of the relaxation in which one of the subproblems integrates the impact of the *timing* of deliveries to the various end-customers with the inventory decisions, and (iii) upper bounds on the values of certain decision variables that are derived from the solutions to variants of certain subproblems. The subproblem mentioned in (ii) above has the unusual feature of potentially negative setup costs (for some values of the Lagrange multipliers) and we develop an optimal polynomial-time solution procedure for it.

We also consider two variants of the problem in which we impose one or both of the constraints implicitly assumed in much of the literature. The weaker of the two constraints permits a route to be used at most once each day, and the stronger constraint limits the number of routes servicing each customer each day to at most one. Computational results indicate that the Lagrangian procedure performs well on difficult problem instances for which it is ineffective to simply apply CPLEX to (P). The results also suggest that the imposition of the additional simplifying constraints does not significantly affect the quality of the solutions when it is unlikely that two trucks will be sent to a single customer on the same day in an optimal solution, and that the resulting problems require much less computational effort to solve. When demand is such that more than one stop per day is required at a customer (i.e., Variant 2 or (P) is appropriate), the Lagrangian procedure obtains very good solutions fairly quickly. More notably, the Lagrangian procedure produces very strong bounds, and thus may be valuable within a branch-and-bound procedure.

Several generalizations can be handled with no modification or only minor modifications to our approach. Time varying costs require no change in the solution procedure. Constraints on route duration and delivery and pick-up time windows can be considered in the route generation routine. Heterogeneous truck types can be handled by generating routes applicable to each truck type. If a job type can be serviced by more than one type of vehicle, then our algorithm for (P2) cannot be used directly, but because this subproblem is separable by job type, it can be solved using commercial software with a concomitant increase in the CPU time. If truck availability imposes practical limitations and rental vehicles are available, it would be possible to add the cost of a rental vehicle to each route and solve the problem in the usual way. The ability to avoid rental costs for the routes covered by the 3PL provider's own vehicles would create a "sunk" benefit in the model (i.e., it would appear as a non-controllable "cost" in the objective function that would not actually need to be paid), and the rental costs for all additional vehicles would be properly accounted for.

Multi-day routes can be handled with a modification to the formulation to account for the actual day of delivery and the extra cost of inventory due to goods in transit. Also, allowing multiple shipments to be loaded onto a truck at the same time can be handled, in principle, for small, pre-defined sets of job types that typically have small shipments. Considerable "bookkeeping" effort in the route generation scheme may be required, as well as a change in the solution method for P2. (The revised version of P2 could still be solved easily using commercial software.) Of course, if the solution generated by our procedure allows for consolidation of loads for customers that happen to be on the same route, then any such route can be modified to take advantage of such opportunities if they reduce costs.

The Lagrangian procedure may be enhanced by devising more effective multiplier adjustment methods especially designed for this problem, or other methods for constructing feasible solutions from the Lagrangian solutions. Also, if the problem contains a particularly large number of potential routes, it may be possible to generate only a subset of the routes *a priori* and to utilize a column generation-based approach to construct other economically viable routes. Further research is needed to explore the implications of such a strategy.

Recall that subproblem (P2) already is separable by customer, so it is easily solved for large numbers of customers. It may be necessary, however, to devise a more efficient solution method for subproblem (P1).

Further research is also needed to consider more rigid constraints on allowable day-of-week combinations (e.g., MWF or Tu-Th) and the possibility of backorders, and to handle uncertainty in demand and transit times.

References

Aggarwal, A. and J. K. Park (1990). Improved Algorithms for Economic Lot-Size Problems. Working Paper, Laboratory for Computer Science, MIT, Cambridge, MA.

Anily, S. and M. Tzur (2002). Shipping Multiple Items by Capacitated Vehicles–An Optimal Dynamic Programming Approach. Working paper, Faculty of Management, Tel Aviv University, Tel Aviv, Israel. To appear in *Transportation Science*.

Baker, K. R., P. Dixon, M.J. Magazine and E.A. Silver (1978). An Algorithm for the Dynamic Lot-Size Problem with Time-Varying Production Capacity Constraints. *Management Science* **24** (16), 1710-1720.

Bard, J.F., L. Huang, P. Jaillet and M.A. Dror (1998). A Decomposition Approach to the Inventory Routing Problem with Satellite Facilities. *Transportation Science* **32** (2), 189-203.

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

Camerini, P., L. Fratta and F. Maffioli (1975). On Improving Relaxation Methods by Modified Gradient Techniques. *Mathematical Programming Study* **3**, 26-34.

Carter, M.W., J.M. Farvolden, G. Laporte and J. Xu (1996). Solving an Integrated Logistics Problem Arising in Grocery Distribution. *INFOR* **34** (4), 290-306.

Chan, L.M., A. Federgruen and D. Simchi-Levi (1998). Probabilistic Analyses and Practical Algorithms for Inventory-Routing Models. *Operations Research* **46** (1), 96-106.

Chandra, P. (1993). A Dynamic Distribution Model with Warehouse and Customer Replenishment Requirements. *Journal of the Operational Research Society* **44** (7), 681-692.

Chandra, P. and M.L. Fisher (1994). Coordination of Production and Distribution Planning. *European Journal of Operational Research* **72** (3), 503-517.

Chao, I.M., B.L. Golden and E.A. Wasil (1995). A New Heuristic for the Period Traveling Salesman Problem. *Computers and Operations Research* **22** (5), 553-565.

Chien, T.W., A. Balakrishnan and R.T. Wong (1989). An Integrated Inventory Allocation and Vehicle Routing Problem. *Transportation Science* **23** (2), 67-76.

Dror, M., M. Ball and B. Golden (1985). A Computational Comparison of Algorithms for the Inventory Routing Problem. *Annals of Operations Research* **4**, 3-23.

Dror, M. and L. Levy (1986). A Vehicle Routing Improvement Algorithm Comparison of a "Greedy" and a Matching Implementation for Inventory Routing. *Computers and Operations Research* **13** (1), 33-45.

Dror, M. and P. Trudeau (1996). Cash Flow Optimization in Delivery Scheduling. *European Journal of Operational Research* **88**, 504-515.

Federgruen, A. and G. van Ryzin (1997). Probabilistic Analysis of a Combined Aggregation and Math Programming Heuristic for a General Class of Vehicle Routing and Scheduling Problems. *Management Science* **43** (8), 1060-1078.

Federgruen, A. and P. Zipkin (1984). A Combined Vehicle Routing and Inventory Allocation Problem. *Operations Research* **32** (5), 1019-1037.

Federgruen, A., G. Prastacos and P.H. Zipkin (1986). An Allocation and Distribution Model for Perishable Products. *Operations Research* **34** (1), 75-82.

Federgruen, A. and M. Tzur (1991). A Simple Forward Algorithm to Solve General Dynamic Lot Sizing Models with n periods in $O(n \log n)$ or $O(n)$ Time. *Management Science* **37** (8), 909-925.

Florian, M. and M. Klein (1971). Deterministic Production Planning with Concave Costs and Capacity Constraints. *Management Science* **18** (1), 12-20.

Garfinkel, R.S. and G.L. Nemhauser (1969). The Set Partitioning Problem: Set Covering with Equality Constraints. *Operations Research* **17**, 848-856.

Gaudioso, M. and G. Paletta (1992). A Heuristic for the Periodic Vehicle-Routing Problem. *Transportation Science* **26** (2), 86-92.

Held, M., P. Wolfe and H. Crowder (1974). Validation of Subgradient Optimization. *Mathematical Programming* **6**, 62-88.

Herer, Y.T. and R. Levy (1997). The Metered Inventory Routing Problem, an Integrative Heuristic Algorithm. *International Journal of Production Economics* **51** (1, 2), 69-81.

Herer, Y. and R. Roundy (1997). Heuristics for a One-Warehouse Multiretailer Distribution Problem with Performance Bounds. *Operations Research* **45** (1), 102-115.

Larson, R.C. (1988). Transporting Sludge to the 106-mile Site: An Inventory/Routing Model for Fleet Sizing and Logistics System Design. *Transportation Science* **22** (3), 186-98.

Lee, C.-G., Y.A. Bozer and C.C. White, III (2003). A Heuristic Approach and Properties of Optimal Solutions to the Dynamic Inventory Routing Problem. Working paper, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, Canada.

Lee, C.-Y. (1989). A Solution to the Multiple Set-Up Problem with Dynamic Demand. *IIE Transactions* **21** (3), 266-270.

Lippman, S. (1969). Optimal Inventory Policy with Multiple Set-up Costs. *Management Science* **16** (1), 118-138.

Love, S.F. (1973). Bounded Production and Inventory Models with Piecewise Concave Costs. *Management Science* **20** (3), 313-318.

Metters, Richard D. (1996). Interdependent Transportation and Production Activity at the United States Postal Service. *Journal of the Operational Research Society* **47** (1), 27-37.

Pochet, Y. and L.A. Wolsey (1993). Lot-Sizing with Constant Batches: Formulation and Valid Inequalities. *Mathematics of Operations Research* **18** (4), 767-785.

Qu, W.W., J.H. Bookbinder, and P. Iyogun (1999). An Integrated Inventory-Transportation System with Modified Periodic Policy for Multiple Products. *European Journal of Operational Research* **115**, 254-269.

Russell, R.A. and D. Gribbin (1991). A Multiphase Approach to the Period Vehicle Routing Problem. *Networks* **21** (7), 747-765.

Viswanathan, S. and K. Mathur (1997). Integrating Routing and Inventory Decisions in One-Warehouse Multiretailer Multiproduct Distribution Systems. *Management Science* **43** (3), 294-312.

Wagelmans, A., S. Van Hoesel and A. Kolen (1992). Economic Lot Sizing: An $O(n \log n)$ Algorithm that Runs in Linear Time in the Wagner Whitin Case. *Operations Research* **40**, Suppl. No. 1, S145-S156.

Wagner, H.M. and T.M. Whitin (1958). Dynamic Version of the Economic Lot Size Model. *Management Science* **5** (1), 89-96.

Webb, I.R. and R.C. Larson (1995). Period and Phase of Customer Replenishment–A New Approach to the Strategic Inventory/Routing Problem. *European Journal of Operational Research* **85** (1), 132-148.

Appendix: Computational Implementation Issues

CPLEX

For all executions of the CPLEX software on the original problem and its variants, we use strong branching, i.e., the branching variable is selected whose resolution is most likely to yield the greatest improvement in the objective function value. This setting provides the best overall performance.

Lagrangian Procedure

Within the Lagrangian procedure, we employ variants of the standard subgradient optimization method to update the multipliers. For Assumption 1, we use the variant of the subgradient procedure (Held et al. 1974) described in Camerini et al. (1975). For Variant 2 and the original problem, we use a version in which the scale factor is halved if the lower bound has not improved after 5 iterations. We also update the multipliers using a weight of 0.35 on the slack from the prior iteration and a weight of 1 on the slack from the current iteration.

To solve subproblem (P2), which can be solved easily using CPLEX (thus obviating the need for the special-purpose algorithm developed in Section 3.1), we use the default branch-and-bound algorithmic settings, including the default optimality tolerance of 0.0001.

Subproblem (P1) is more difficult to solve than (P2). For the original formulation and for Variant 2, we use the solution from the prior iteration as a "warm start" for the next iteration. We do not utilize the CPLEX-generated cuts because we observed that they do not provide much benefit relative to the CPU effort. We also select the CPLEX parameter setting that emphasizes optimality over feasibility. We solve each subproblem to within 5% of optimality or stop after 1000 seconds, whichever occurs sooner. For Variant 1, we similarly use the solution from the prior iteration as a "warm start" for the next iteration and we turn off the CPLEX-generated cuts. We use a search strategy in which the branching variable is selected based on "pseudo-reduced" costs, i.e., estimates of the change in the objective from rounding a fractional variable to the nearest integer; the branching node is selected based on the best integer objective that can be achieved from solving the subproblem corresponding to all nodes eligible for selection. We solve each subproblem to within 1% of optimality. The ease with which these subproblems are solved in contrast to those in Variant 2 and the original problem obviates the need for a time limit.

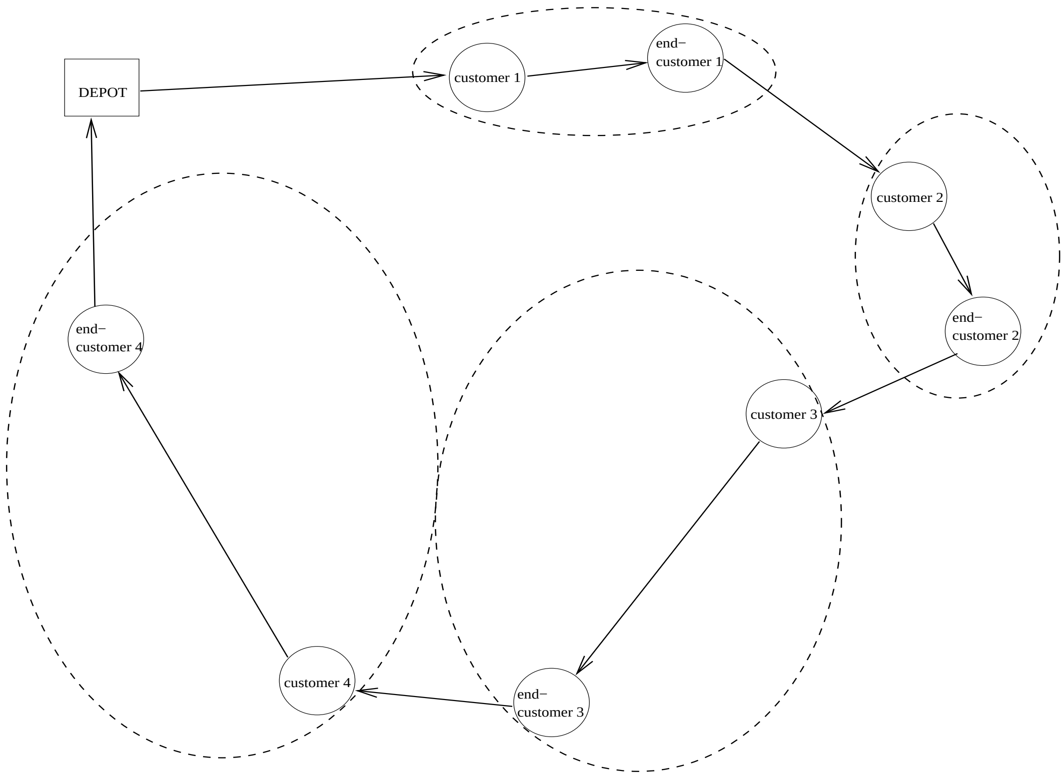

Figure 1: Depiction of an example route