

Time Slot Management in Attended Home Delivery

Niels Agatz

Department of Decision and Information Science, Rotterdam School of Management, Erasmus University,
3062 PA DR Rotterdam, The Netherlands, nagatz@rsm.nl

Ann Campbell

Department of Management Sciences, Tippie College of Business, University of Iowa, Iowa City, Iowa 52242,
ann-campbell@uiowa.edu

Moritz Fleischmann

Logistics and Supply Chain Management, University of Mannheim Business School, 68131 Mannheim, Germany,
moritz.fleischmann@bwl.uni-mannheim.de

Martin Savelsbergh

CISRO Mathematics, Informatics, and Statistics, North Ryde, NSW 1670 Australia
martin.savelsbergh@csiro.au

Many e-tailers providing attended home delivery, especially e-grocers, offer narrow delivery time slots to ensure satisfactory customer service. The choice of delivery time slots has to balance marketing and operational considerations, which results in a complex planning problem. We study the problem of selecting the set of time slots to offer in each of the zip codes in a service region. The selection needs to facilitate cost-effective delivery routes, but also needs to ensure an acceptable level of service to the customer. We present a fully automated approach that is capable of producing high-quality delivery time slot offerings in a short amount of time. Computational experiments reveal the value of this approach and the impact of the environment on the underlying trade-offs.

Key words: e-grocery; home delivery; time slots; vehicle routing; continuous approximation; integer programming

History: Received: April 2008; revisions received: October 2009, August 2010. Published online in *Articles in Advance* December 20, 2010.

1. Introduction

The success of many e-commerce businesses hinges upon their ability to offer efficient and effective “last-mile” delivery. This task is especially challenging in attended home delivery, particularly for Internet grocers. The dramatic failures of e-grocery pioneers such as Streamline and Webvan underline the importance of a cost-effective delivery operation. Despite initial failures, the online grocery market has recovered and has grown spectacularly since 2003, showing an annual growth rate of 42% (Berning, Ernst, and Hooker 2005).

To provide a high service level and to avoid delivery failures as much as possible, it is customary in attended home delivery services for the company to offer the customer a choice of narrow delivery time slots. The specific time slot offering does not only impact the perceived customer service but also the expected delivery efficiency. In this paper, we consider the tactical problem of selecting the time slots to offer in each zip code of the delivery region. This selection needs to offer an acceptable level of service in each zip code and needs to be able to yield cost-effective daily

delivery routes. Once such a time slot schedule is put in place, there is the operational problem of managing the availability of the offered time slots. We will focus only on the tactical, not the operational problem.

Our research effort is motivated by operations at Albert.nl, the Internet channel of Albert Heijn, the largest grocery supermarket chain in the Netherlands and a subsidiary of Royal Ahold. Albert.nl is now the only full-scale Internet grocer in the Netherlands. Started in 2001, their delivery service is now available to around 60% of the country’s households. From dedicated warehouses, the company uses small vans to deliver a broad assortment of grocery products to each customer’s home. The time slot offering at Albert.nl is typical for those in the e-grocery industry (see, for example, Peapod.com and Tesco.com). The customer can choose a two-hour delivery slot from a menu of time slots available in his or her specific zip code. Each time slot has a corresponding delivery fee ranging from €4.95 to €8.95, depending on the day and time of delivery. On a day-to-day basis, when delivery capacity becomes tight, certain time slots in certain zip codes may be closed. Closing a time slot at

Albert.nl means that it is no longer available for the customer to choose (the slot turns “grey” on the web-site). Routes are planned for each morning and afternoon shift. After the order cutoff time for a shift has passed and thus all orders are known, the delivery routes for that shift are planned using a commercial vehicle routing package. The routing package properly takes delivery windows, dwell times, and vehicle capacities into account.

Albert.nl uses a differentiated time slot offering linked to the demand characteristics of each zip code. This means that a zip code with low demand potential receives only a few weekly time slots in an attempt to concentrate that demand, where high-demand zip codes receive a larger number of slots. The slots offered, regardless of the total number, must also exhibit a balance over the week between morning and afternoon time slots to ensure an acceptable level of service and choice to the customers. Determining the specific time slots for each zip code involves a careful trade-off between marketing and operational considerations. After deciding the number of morning and afternoon slots that will satisfy marketing needs, referred to hereafter as service requirements, the specific time slots are selected for each zip code to satisfy operational needs such as a low-cost delivery schedule. Because delivery vehicles may visit several zip codes during a single time slot and a delivery route spans multiple time slots, routing considerations should play a vital role in the construction of a time slot schedule. Assigning specific time slots to a zip code should not be done in isolation but should be considered jointly for neighboring slots. This results in a complex planning problem.

The time slot schedule employed at Albert.nl is created manually. Because of its complexity, this task takes several weeks to complete. Clearly, this is undesirable in an environment that continually changes as a result of a 30% annual growth rate. In particular, this manual process greatly limits their ability to assess and compare potential future scenarios. Therefore, Albert.nl is considering options for automating the time slot schedule generation. In this paper, we present an approach for making the time slot decision that applies not only to Albert.nl but also for many other companies offering attended delivery services.

More specifically, we address the following time slot management problem (TSMP). Given service requirements and average weekly demands for each zip code in the delivery region, determine the set of time slots to offer in each zip code so as to minimize expected delivery costs while meeting the service requirements. We explore a continuous approximation approach for the TSMP. This type of approach focuses on a realistic evaluation of a given time slot

schedule and then uses relatively simple optimization techniques to improve the schedule.

Our main contributions are

- the identification and explanation of a relevant planning concern in the e-grocery business that gives rise to challenging optimization problems;
- the development of a fully automated approach capable of producing high-quality time slot schedules much faster than the current manual process;
- a computational study analyzing the viability and merits of the approach; and
- a computational study investigating the impact of changes in the environment, such as increases in demand and demand variability, more relaxed or more stringent service requirements, and changes in the structure of the delivery time slot template.

The remainder of the paper is organized as follows. In §2, we summarize the relevant literature. In §3, we introduce notation and outline our modeling framework. In §4, we present our continuous approximation approach. In §5, we present an alternative integer programming based approach that will be used as a source of comparison in our computational experiments. In §6, we describe the design of our computational experiments and present the results. Finally, in §7, we summarize our main insights and discuss directions for future research.

2. Literature

The choice of time slots is one of many critical decisions in the design of an attended home delivery system, as discussed in Agatz et al. (2008a) and Agatz, Fleischmann, and van Nunen (2008b). Other important issues include the choice between store-based and warehouse-based delivery, inventory allocation, pricing, and return handling. Home delivery and the design and management of time slots has received little attention in the distribution and routing literature. Time slots are typically assumed to be exogenous information, e.g., specified by the customer or set by the sales and marketing department. The rise of e-commerce, which offers numerous opportunities for the use and active management of time slots, is changing this.

Much of the early research on “last-mile” delivery primarily examines the impact of different time slot lengths. For example, Punakivi and Saranen (2001) assess the impact of the time slot length on transportation costs for attended and unattended deliveries. Their results indicate substantial efficiency gains when time constraints are relaxed. Completely flexible, unattended delivery reduces costs by up to a third, relative to attended delivery with two-hour time slots. Lin and Mahmassani (2002) summarize the delivery policies employed by many online grocers in

the United States and use vehicle routing software to evaluate the performance of some of these policies on a few realistic instances of the problem. Both unattended and attended policies are considered along with different time slot lengths.

Another related stream of research focuses on the real-time management of time slots in attended home delivery. This work draws on revenue management concepts and applies them in a distribution setting. In this vein, both Bent and Van Hentenryck (2004) and Campbell and Savelsbergh (2005a) examine which deliveries to accept or reject. Their approaches exploit stochastic information about future requests to decide on requests under consideration. Bent and Van Hentenryck (2004) aim to maximize the number of accepted requests by controlling the time slots offered, but they do not consider rejecting an “expensive” delivery to preserve resources for more, future deliveries, in contrast with Campbell and Savelsbergh (2005a). Campbell and Savelsbergh (2005b) study the use of incentive schemes to reduce delivery costs by encouraging the selection of time slots that result in low-cost delivery routes. In contrast to the real-time management of time slots, our TSMP is tactical in nature and allocates time slots geographically prior to actual order intake. The resulting time slot schedules serve as a starting point for subsequent real-time adjustments.

Anticipating the impact of the offered time slots on delivery routes is crucial when solving the TSMP. This links the TSMP to the vast body of literature on vehicle routing with time windows. See Braysy and Gendreau (2005a, b) for an extensive review. In particular, the TSMP shares some aspects with dynamic and stochastic vehicle routing. In dynamic vehicle routing, as discussed in Jaillet and Odoni (1988), Bertsimas, Jaillet, and Odoni (1990), and Gendreau and Potvin (1998), not all inputs are known beforehand but become known concurrently with the determination of the solution. In stochastic routing, some components of the problem are stochastic, such as travel times (Kenyon and Morton 2003), demand (Dror, Laporte, and Trudeau 1989), or customers (Hvattum, Lokketangen, and Laporte 2006). Another problem with features similar to those of the TSMP is the period vehicle routing problem with service choice (PVRP-SC), where vehicle routes need to be constructed over a period of time and service frequency is a decision in the model (Francis, Smilowitz, and Tzur 2006; Francis and Smilowitz 2006).

Given the computational complexity of the vehicle routing problem alone, it is unrealistic to incorporate a full routing model in the tactical TSMP. Instead, routing costs resulting from time slot selections have to be approximated. One way is to model routing costs using the continuous approximation method

(Daganzo 1987a, b). This approach represents demand by continuous functions and assesses system-wide costs by aggregating over “local” cost estimates. For an overview of continuous approximation models in logistics, see Daganzo (2005) and Langevin, Mbaraga, and Campbell (1996). Another option is to explicitly model routing decisions but on a more aggregate level than a full vehicle routing problem. This idea is the basis for the alternative approach explored in §5.

3. Assumptions and Notation

The objective of the TSMP is to minimize the expected delivery costs for the customer orders ensuing from the offered delivery time slots. This requires assumptions on the demand response to a certain time slot offering. Throughout our analysis we make the following two key assumptions, where demand is measured in number of customer orders:

1. The expected weekly demand for each zip code is known and independent of the set of offered time slots.
2. The expected weekly demand for each zip code is divided evenly over the set of offered time slots.

The first assumption supposes that customers exhibit a certain degree of flexibility, such that changing the time slots offered does not result in lost sales. An empirical analysis of historical data from Albert.nl supports the validity of this assumption for moderate changes in the time slot offering (Agatz 2009). The second assumption implies that all time slots are equally popular. Differentiated delivery fees are introduced specifically for this reason. They counterbalance differences in time slot popularity and smooth demand. Historical data from Albert.nl show the effectiveness of differentiated delivery fees and indicate that the equal time slot popularity assumption is reasonable.

We further assume that all orders are of the same size in terms of the number of totes. The size of an order impacts the number of orders that fit in a delivery vehicle. We conducted a few computational experiments in which we varied order sizes and found that it had little or no impact on the results.

The marketing-imposed service requirements limit the set of feasible time slot offerings. Our models treat these service requirements as exogenous. The service requirements employed at Albert.nl specify for each zip code the required number of available delivery slots in each shift, e.g., one on Monday morning and two on Tuesday afternoon. As a result of these specific service requirements and the above demand assumptions, the problem decomposes by shift. Thus, separate problems need to be solved for each shift.

Our objective function considers fixed costs as well as variable costs, incorporating both distance-related

Table 1 Notation

\mathcal{Z} :	Set of zip codes (indexed by i)
\mathcal{T} :	Set of time slots (indexed by s)
\mathcal{V} :	Set of vehicles
e_i :	Expected demand for zip code i for each time slot
Q :	Vehicle capacity (in terms of the number of orders that can be accommodated)
f :	Vehicle fixed cost
c^n :	Cost per mile for vehicle and driver usage when driving within a zip code and within a time slot
c^z :	Cost per mile for vehicle and driver usage when driving between different zip codes within a time slot
c^t :	Cost per mile for vehicle and driver usage when driving between zip codes in consecutive time slots
c^0 :	Cost per mile for vehicle and driver usage when driving to or from the depot
d_{ij} :	Distance between centers of zip codes i and j
τ :	Dwell time per order

and time-related costs. We do not take into account delivery fees and solely focus on minimizing expected delivery costs.

In Table 1, we introduce notation that is common to the proposed continuous approximation approach as well as the alternative integer programming based approach that will be used for comparison. Note that e_i is the weekly demand for zip code i divided by the number of time slots that need to be offered in zip code i .

4. Continuous Approximation Approach

At the heart of the continuous approximation approach is a model for estimating the delivery cost of a given time slot schedule for a set of zip codes. Based on this delivery cost estimate, a local search is performed to iteratively improve the time slot schedule. The original continuous approximation approach proposed by Daganzo (1987a) does not consider time slots and distinguishes two components of a vehicle tour, namely the stem distance between the depot and the delivery area and the distance between consecutive stops within the delivery area. To account for time slots, we expand these ideas and distinguish four components of vehicle tours (see Figure 1):

- d^0 : the stem distance to or from the depot to a stop in a time slot;
- d^n : the distance between stops within a zip code within a time slot;
- d^z : the distance between stops in different zip codes within a time slot; and
- d^t : the distance between stops in consecutive time slots.

The corresponding travel times are denoted by h^0 , h^n , h^z , and h^t .

In line with the continuous approximation methodology, we estimate the distance values for these

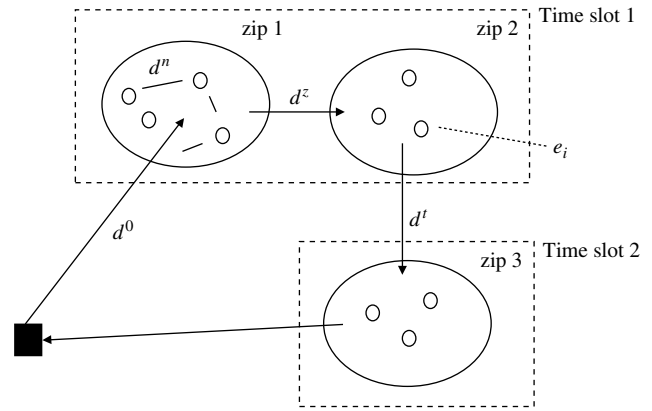


Figure 1 Routing Components

components of a vehicle tour based on local data. We estimate the distance values from the perspective of each zip-code time slot combination (i, s) in the time slot schedule. These local estimates are then used to compute a local estimate of the distance per order (excluding the stem distance to and from the depot) as follows:

$$DPO(i, s) = \frac{1}{n_{is}} \left[n_{is}^t \left(\frac{n_{is}^t - 1}{n_{is}^t} d_{is}^t + n_{is}^z \left(\frac{n_{is}^z - 1}{n_{is}^z} d_{is}^z + (n_{is}^n - 1) d_{is}^n \right) \right) \right],$$

where

- n_{is} denotes the estimated number of orders per route,
- n_{is}^t denotes the estimated number of time slots covered by a route,
- n_{is}^z denotes the estimated number of zip codes visited in time slot on a route, and
- n_{is}^n denotes the estimated number of orders delivered in a zip code in a time slot on a route.

These n -values are again estimated based on local data from the perspective of a given zip-code time-slot (i, s) .

By multiplying the distance per order with the cost per kilometer, we obtain a local estimate for the distance-related costs (excluding stem distances). This, in turn, yields a local estimate of the cost per order (excluding stem costs and fixed vehicle costs):

$$CPO(i, s) = \frac{1}{n_{is}} \left[n_{is}^t \left(\frac{n_{is}^t - 1}{n_{is}^t} d_{is}^t c^t + n_{is}^z \left(\frac{n_{is}^z - 1}{n_{is}^z} d_{is}^z c^z + (n_{is}^n - 1) d_{is}^n c^n \right) \right) \right].$$

We obtain an estimate of the expected delivery cost associated with a given time slot schedule by multiplying the expected number of orders e_i for a zip-code time slot combination (i, s) with the cost per order

$CPO(i, s)$, aggregating over all zip codes and time slots, and adding stem costs and fixed vehicle costs:

$$\text{expected delivery cost} \approx \sum_{i \in \mathcal{Z}} \sum_{s \in \mathcal{T}} e_i CPO(i, s) + d^0 c^0 + fv,$$

where d^0 is the estimated total stem distance and v is the estimated number of vehicles required. Note that we sum over all zip codes and all offered time slots for that zip code. Also note that we omit labor costs for the dwell times because these costs are constant and independent of the offered schedule.

4.1. Estimating Vehicle Tour Components—Part I

In this subsection, we discuss how the vehicle tour components are estimated from the perspective of a given zip-code time slot combination (i, s) . We start with the vehicle tour components used in the calculation of the cost per order.

4.1.1. Distance Between Stops Within a Zip Code Within a Time Slot (d_{is}^n). Statistical analysis of recent routing data at Albert.nl shows that the driving distance between two customers within the same zip code is similar across zip codes; we use d^n to represent this distance. The same is true for the travel time between two customers within the same zip code, so we use h^n to represent this travel time.

4.1.2. Distance Between Stops in Different Zip Codes Within a Time Slot (d_{is}^z). Daganzo (2005) approximates the distance between two consecutive stops of a route through a region with a slowly varying demand density δ by $k/\sqrt{\delta}$, where k is a dimensionless constant that is independent of the region shape. We apply this approach on a zip code level and consider the density of open zip codes in time slot s in the neighborhood of i . Let \mathcal{Z}_i be a collection of zip codes in the neighborhood of zip code i (including zip code i itself). In our calculations, we take \mathcal{Z}_i to be the set of zip codes within a given maximum distance from the center of zip code i . Let δ_{is} be the density of open zip codes in \mathcal{Z}_i during time slot s . If a_j denotes the surface area of zip code j and \mathcal{F}_{js} denotes whether zip code j is open during time slot s ($\mathcal{F}_{js} = 1$) or whether it is closed during time slot s ($\mathcal{F}_{js} = 0$), then

$$\delta_{is} = \frac{\sum_{j \in \mathcal{Z}_i} \mathcal{F}_{js}}{\sum_{j \in \mathcal{Z}_i} a_j}.$$

The estimate of d_{is}^z is then $k/\sqrt{\delta_{is}}$. We impose an upper bound \bar{d}^z on this estimate to handle very low densities δ_{is} . In our calculations, we set \bar{d}^z equal to twice the average diameter of a zip code. Thus, $d_{is}^z = \min(k/\sqrt{\delta_{is}}, \bar{d}^z)$.

The above calculation of δ_{is} does not take into account that there may be open zip codes with an expected demand per slot e_i smaller than one. To

account for this situation, we modify δ_{is} to be a weighted density:

$$\delta_{is} = \frac{\sum_{j \in \mathcal{Z}_i} \min(1, e_j) \mathcal{F}_{js}}{\sum_{j \in \mathcal{Z}_i} a_j}.$$

This can be interpreted as treating values of e_i smaller than one as probabilities of a demand occurrence.

4.1.3. Distance Between Stops in Consecutive Time Slots (d_{is}^t). We take a similar approach when computing d_{is}^t , but using zip code densities in preceding and succeeding time slots. We set $d^{\text{pre}} = d^n$ if time slot $s - 1$ is open for zip code i , i.e., for consecutive open time slots we simply use the average distance between stops within a zip code. Otherwise, we set $d^{\text{pre}} = \min\{\bar{d}^z, k/\sqrt{\delta_{i, s-1}}\}$, where $\delta_{i, s-1}$ is the density of open zip codes in \mathcal{Z}_i in time slot $s - 1$. Analogously, $d^{\text{suc}} = d^n$ if time slot $s + 1$ is open for zip code i and $d^{\text{suc}} = \min\{\bar{d}^z, k/\sqrt{\delta_{i, s+1}}\}$ otherwise, where $\delta_{i, s+1}$ is the density of open zip codes in \mathcal{Z}_i in time slot $s + 1$. Finally, we set $d_{is}^t = d^{\text{pre}}$ (d^{suc}) if s is the first (last) time slot of the shift and $d_{is}^t = (d^{\text{pre}} + d^{\text{suc}})/2$ otherwise.

4.1.4. Number of Stops in a Zip Code in a Time Slot on a Route (n_{is}^n). The number of stops in a zip code in a time slot is limited by the demand e_i and by the length of the time slot l . Let ν^n be the maximum number of stops that can be made during a time slot of length l and recall that τ denotes the dwell time. Then $l - h_{is}^t = \nu^n \tau + (\nu^n - 1)h_{is}^n$ and thus $\nu^n = (l - h_{is}^t + h^n)/(\tau + h^n)$, where we subtract h_{is}^t from l to account for travel time between consecutive time slots. The number of stops in a zip code during a time slot n_{is}^n is therefore $\min\{e_i, \nu^n\}$.

4.1.5. Number of Zip Codes Visited in a Time Slot on a Route (n_{is}^z). Analogously, the number of zip codes visited in a time slot is limited by the number of neighboring open zip codes n^{open} and by the length of the time slot l . Let ν^z denote the maximum number of zip codes that can be visited during a time slot of length l . Then $l - h_{is}^t = \nu^z [n_{is}^n \tau + (n_{is}^n - 1)h_{is}^n] + (\nu^z - 1)h_{is}^z$, and thus $\nu^z = (l - h_{is}^t + h_{is}^z)/[n_{is}^n \tau + (n_{is}^n - 1)h^n + h_{is}^z]$. We then set $n_{is}^z = \min\{n^{\text{open}}, \nu^z\}$.

4.1.6. Number of Time Slots Covered on a Route (n_{is}^t). The number of time slots covered on a delivery tour is limited by the number of time slots in the shift $|\mathcal{T}|$ and by the number of orders that can be accommodated by the delivery vehicle Q . Therefore, we set $n_{is}^t = \min\{|\mathcal{T}|, Q/(n_{is}^n n_{is}^z)\}$.

4.1.7. Number of Stops on a Route (n_{is}). Using the previously estimated parameters, the number of stops on a delivery tour is $n_{is}^t n_{is}^z n_{is}^n$.

4.2. Estimating Vehicle Tour Components—Part II

In the previous subsection, we have shown how we compute local estimates for the vehicle tour components used in the computation of the local estimate of the cost per order. What remains to be shown is how to estimate the number of vehicles required (v) and the stem distance (d^0).

4.2.1. Number of Vehicles (v). We start with the estimate of the number of vehicles required. The number of vehicles required for serving the total demand of a shift depends on the vehicle capacity and on the number of orders that can be delivered in a time slot, i.e., on physical capacity and on available time. We address each of these constraints separately.

We have local estimates for the number of orders per route (n_{is}). We take a demand-weighted average over these local estimates to obtain a global estimate of the number of orders per route. Dividing the expected demand for a shift by this global estimate of the number of order per route gives an estimate of the number of routes r . More precisely, let \mathcal{Z}^s denote the set of open zip codes in time slot s . Then we have

$$r = \frac{\sum_{i \in \mathcal{Z}^s, s \in \mathcal{T}} e_i}{\sum_{i \in \mathcal{Z}^s, s \in \mathcal{T}} (e_i / \sum_{i \in \mathcal{Z}^s, s \in \mathcal{T}} e_i) n_{is}} = \frac{(\sum_{i \in \mathcal{Z}^s, s \in \mathcal{T}} e_i)^2}{\sum_{i \in \mathcal{Z}^s, s \in \mathcal{T}} e_i n_{is}}.$$

Similarly, we estimate the number of vehicles required to serve all orders in a given time slot v_s , based on the expected demand per time slot and the estimated number of stops per time slot, with

$$v_s = \frac{(\sum_{i \in \mathcal{Z}^s} e_i)^2}{\sum_{i \in \mathcal{Z}^s} e_i n_{is}^z n_{is}^n}.$$

The number of vehicles required is the maximum of the number of routes r and the number of vehicles required to serve all orders in a time slot v^s . This is represented by $v = \max\{r, \max_{s \in \mathcal{T}} v_s\}$.

4.2.2. Stem Distance (d^0). Next, we consider the (total) stem distance d^0 . We start by estimating d_s^0 , the expected stem distance traveled by a vehicle making its first delivery in time slot s . This is done by taking the average distance between the depot and the open zip codes in time slots s :

$$d_s^0 = \frac{\sum_{j \in \mathcal{Z}_i} \mathcal{J}_{js} \times d_{0j}}{\sum_{j \in \mathcal{Z}_i} \mathcal{J}_{js}}.$$

Of course, the expected stem distance traveled by a vehicle making its last delivery in time slot s is the same. Next, we use the estimates of the number of vehicles required in each time slot (v_s) to estimate the number of vehicles that start their trip in time slot s and to estimate the number of vehicles that end their trip in time slot s . Let $v_0 = 0$ and $v_{|\mathcal{T}|+1} = 0$, then we estimate the (total) stem distance as

$$d^0 = \sum_{s=0}^{|\mathcal{T}|} (\max\{0, v_{s+1} - v_s\} d_{s+1}^0 + \max\{0, v_s - v_{s+1}\} d_s^0).$$

4.2.3. Handling Overlapping Time Slots. A further complication arises if time slots are overlapping, as is the case at Albert.nl. In our calculations, we handle this situation by creating adjusted nonoverlapping slots. To this end, we apportion the length of the overlapping part of two windows to both of the individual windows, based on their expected demand volumes. For example, consider two two-hour windows that overlap for one hour. Furthermore, assume that the total expected demand from all open zip codes in the first time slot is twice as large as the total expected demand in the second time slot. Then, we create a first time slot of size 1.66 hours and a second slot of size 1.33 hours.

4.3. Solution Method

The discussion above shows how to estimate the expected costs of a given time slot allocation. The next step is to optimize the time slot allocation based on this evaluation. Note that this optimization problem is nonlinear and nonconvex. Therefore we use a simple greedy iterative improvement heuristic. We can start from any feasible time slot schedule. Observe that finding a feasible schedule for the TSMP is not difficult because any schedule that satisfies the service requirements is feasible. In our experiments, we begin with the schedule currently in place at Albert.nl. Next, we determine for each zip code the time slot allocation for that zip code that results in the minimum expected delivery cost (keeping the time slot allocation for all other zip codes fixed) by complete enumeration. We then adjust the current time slot schedule by implementing the time slot allocation for the zip code that achieves the minimum expected delivery cost. We repeat this process as long as there is a reduction in expected delivery costs greater than some threshold or until a maximum number of iterations is reached.

5. Integer Programming Alternative

The continuous approximation model presented in the previous section does not model the operational-level routing decisions explicitly. Instead, it aims to reflect them implicitly in the employed cost approximations. In this section, we complement this approach with a model that deals with the embedded routing component of the TSMP in a more explicit way and we will compare the results obtained by the continuous approximation model with the results of this model in our computational experiments. As with the continuous approximation model, the objective is to construct a time slot schedule that satisfies the service requirements and minimizes delivery costs.

Intuitively, one could model the routing costs by introducing variables that represent, for each vehicle, the sequence of stops at various zip codes within

each time slot. However, because vehicles visit multiple time slots and may visit multiple zip codes within a time slot, the embedded routing component makes the integer program very hard to solve. Even finding feasible solutions in a reasonable amount of time is quite difficult with modern, sophisticated, and powerful integer programming solvers.

Therefore, we approximate the routing costs using a seed-based scheme (Fisher and Jaikumar 1981). Even though the number of stops a vehicle makes during a shift may be large, the number of time slots in a shift is small; no more than four in the case of Albert.nl. Solving routing problems with at most four “stops” is much easier. This cost approximation is expressed graphically in Figure 2. The dashed circles surround the customers served in a single time slot with the center dot representing the seed. With the use of seeds, we can approximate the routing costs for a vehicle by the sum of two costs: (1) the cost of a route through the seeds associated with each time slot plus (2) a simple estimate of the costs incurred when visiting the customers within each single time slot based on the distance to the seed.

Our integer program has the following variables to represent the different decisions and associated costs:

- z^v is one if vehicle v is used and zero otherwise;
- y_i^{vs} is one if zip code i is the seed for vehicle v for time slot s and zero otherwise;
- x_i^{vs} is one if vehicle v visits zip code i in time slot s and makes a delivery and zero otherwise;
- X_{ij}^{vs} is the distance from zip code i to zip code j if vehicle v serves customer i in time slot s and zip code j is the seed for time slot s and zero otherwise;
- Y_{ij}^{vs} is the distance between the seed of time slot $s - 1$ and the seed of time slot s for vehicle v if zip code i is the seed of time slot $s - 1$ and zip code j is the seed of time slot s and zero otherwise;
- u_i^s is one if a particular time slot s is offered for a particular zip code i and zero otherwise; and
- q_i^{vs} is the portion of the demand of zip code i in time slot s that is served by vehicle v .

One zip code will serve as the seed (or representative) for the customers served in a particular time

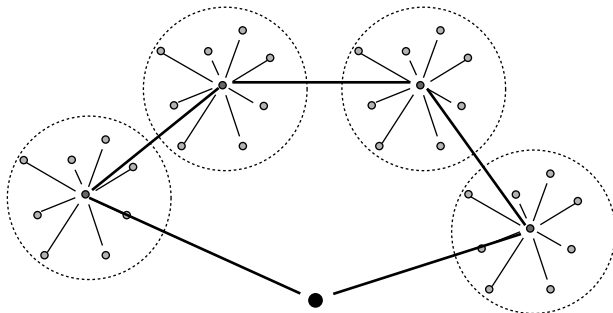


Figure 2 Routing with Seeds

slot/vehicle combination, and this decision is represented by the y variable. The set of zip codes served in a particular time slot/vehicle combination is captured by the x variables. The estimate of the cost to serve a set of zip codes in a time slot is based on the sum of the distances between the zip codes and the seed, which are represented by the X variables. The cost of visiting the seeds associated with a vehicle, i.e., the route through the seeds, is captured by the Y variables. Note that the desired time slot schedule is implied by the x variables. Slot s is offered in zip code i if and only if $x_i^{vs} = 1$ for some v .

Given the parameters defined in Table 1 and these variables, the objective function can be expressed as

$$\begin{aligned} \min \sum_{v \in V} f \cdot z^v + \sum_{i \in Z, j \in Z, v \in V, s \in \mathcal{T}: s > 1} c^t Y_{ij}^{vs} \\ + \sum_{i \in Z, j \in Z, v \in V, s \in \mathcal{T}} c^n X_{ij}^{vs} + \sum_{j \in Z, v \in V, s \in \mathcal{T}} c^0 X_{0j}^{vs}. \end{aligned}$$

The first term represents the cost for using vehicles/drivers. The second term represents the cost to travel between seeds. The last two sums represent the cost to serve zip codes from seeds and the depot from seeds, respectively. Note that, as in the continuous approximation model, we omit labor costs for the dwell times because they are constant.

The continuous approximation model presented in §4 evaluates the cost of a *given* time slot schedule. Here, though, the model needs not only to capture the costs but also needs to define what makes a schedule feasible. To do so, we must include some new parameters:

- r_i : the number of slots that must be open for zip code i in a shift, i.e., the service requirement for i ,
- t^n : the travel time between customers in the same zip code, and
- t^t : a multiplier to convert the distance between zip codes into a travel time between zip codes.

Next, we describe the constraints that define feasible solutions. First, we examine the linkage between the X and Y variables and the x and y variables. For the X variables, the linkage is defined as follows:

$$X_{ij}^{vs} \geq d_{ij}(x_i^{vs} + y_j^{vs} - 1) \quad i \in Z, j \in Z, v \in V, s \in \mathcal{T}.$$

The above is valid because it forces X_{ij}^{vs} to have nonzero value d_{ij} only if both x_i^{vs} and y_j^{vs} equal one; otherwise, the right-hand side has a value less than or equal to zero. In the same way, we can define the linkage for the Y variables with

$$Y_{ij}^{vs} \geq d_{ij}(y_i^{vs-1} + y_j^{vs} - 1) \quad i \in Z, j \in Z, v \in V, s \in \mathcal{T}: s > 1.$$

Note that the above is only used for $s > 1$ because the cost to travel to the first seed from the depot will be handled by the X variables.

The next set of constraints enforces that every time slot has a seed,

$$\sum_{i \in Z} y_i^{vs} = z^v \quad v \in V, s \in \mathcal{T}.$$

Note that enforcing that every time slot has a seed does not create unnecessary travel costs in the case where a vehicle serves no customers in a time slot, because the seed can be chosen to be the same as the seed of the previous (or subsequent) time slot.

We allow multiple vehicles to visit the same zip code in the same time slot. However, we should make sure that multiple visits to a particular zip code in the same time slot do not count toward the zip code's service requirements. Therefore, we use the binary variable u_i^s in

$$\begin{aligned} \sum_{s \in \mathcal{T}} u_i^s &= r_i \quad i \in Z, \\ \sum_{v \in V} x_i^{vs} &\geq u_i^s \quad i \in Z, s \in \mathcal{T}. \end{aligned}$$

Recall that we have assumed that the demand in a zip code is the same in every time slot offered. If multiple vehicles serve a zip code during a particular time slot, then the zip code's demand e_i must be distributed over the different vehicles. This is defined by the following constraints:

$$\begin{aligned} \sum_{v \in V} q_i^{vs} &\geq e_i u_i^s \quad i \in Z, s \in \mathcal{T}, \\ q_i^{vs} &\leq \min(e_i, Q) x_i^{vs} \quad v \in V, i \in Z, s \in \mathcal{T}. \end{aligned}$$

The next set of constraints enforces that the demand served by a single vehicle does not exceed the vehicle capacity Q :

$$Qz_v \geq \sum_{i \in Z, s \in \mathcal{T}} q_i^{vs} x_i^{vs} \quad v \in V.$$

To properly estimate the routing costs, especially the travel time to and from the depot, we force the depot to be served in the first and last time slot of a particular delivery period, i.e., time slot 1 and time slot \hat{s} :

$$\begin{aligned} x_0^{v1} &= z^v \quad v \in V, \\ x_0^{v\hat{s}} &= z^v \quad v \in V. \end{aligned}$$

Recall from the continuous approximation model that the maximum number of deliveries that a vehicle can make is constrained not only by the vehicle capacity but also by the length of each time slot. Enforcing this time constraint is one of the most complex parts of the integer programming model. The situation is further complicated by the fact that the number and the length of the time slots can vary by day and by

shift. Finally, there is a need to distinguish the first time slot, "in between" time slots, and the last time slot. Recall that travel from the depot and back to the depot can occur outside of the time windows, so it does not need to be incorporated within these time window length constraints. As an example, let l_1 represent the width of the first time slot. Then the constraints to limit the time spent on deliveries during the first time slot is

$$\begin{aligned} \sum_{i \in Z, j \in Z} t^t X_{ij}^{v1} + \sum_{i \in Z} (\tau e_i + t^n(e_i - 1)) x_i^{v1} \\ + 0.5 \sum_{i \in Z, j \in Z} t^t Y_{ij}^{v2} \leq l_1 \quad v \in V. \end{aligned} \quad (1)$$

The first term captures the travel time to the seed and the travel time between the zip codes visited during the time slot. The second term in constraint (1) captures the stop time at a customer served in a zip code as well as the travel time between customers served in a zip code. The final term in constraint (1) captures the travel time from the seed of this time slot to the seed of the next time slot. The 0.5 in the final term reflects the fact that half of this travel time is allocated to this time slot and the other half is allocated to the next time slot. Constraints for the "in between" and final time slots are structured similarly.

To handle overlapping time windows, we add a constraint that assures that the total length of the combined slots is not exceeded. As an example, consider the time slots 8 A.M.–10 A.M. and 9 A.M.–11 A.M. (referred to below by 1 and 2, respectively) and assume that the 8 A.M.–10 A.M. time slot is the first time slot of the shift and that the 9 A.M.–11 A.M. time slot is not the last time slot of the shift. Let the total time period covered by the time slots be denoted by l , which is three hours in this example. We add the following constraint:

$$\begin{aligned} \sum_{i \in Z, j \in Z} t^t (X_{ij}^{v1} + X_{ij}^{v2}) + \sum_{i \in Z, j \in Z} t^t (Y_{ij}^{v1} + 0.5 Y_{ij}^{v2}) \\ + \sum_{i \in Z} (\tau e_i + t^t(e_i - 1)) x_i^{v1} + \sum_{i \in Z} (\tau e_i + t^t(e_i - 1)) x_i^{v2} \leq l. \end{aligned}$$

Last, we add symmetry breaking constraints. These constraints are not necessary for the correctness of the model, but help the integer programming solver find better solutions faster. These symmetry breaking constraints force the lower indexed vehicles to be the ones with the largest number of stops:

$$\sum_{i \in Z, s \in \mathcal{T}} x_i^{vs} \leq \sum_{i \in Z, s \in \mathcal{T}} x_i^{v-1s} \quad v \in V: v > 1.$$

When only a subset of the vehicles is used, there are many equivalent solutions. By imposing

$$z^v \leq z^{v-1} \quad v \in V: v > 1,$$

the number of equivalent solutions is also significantly reduced.

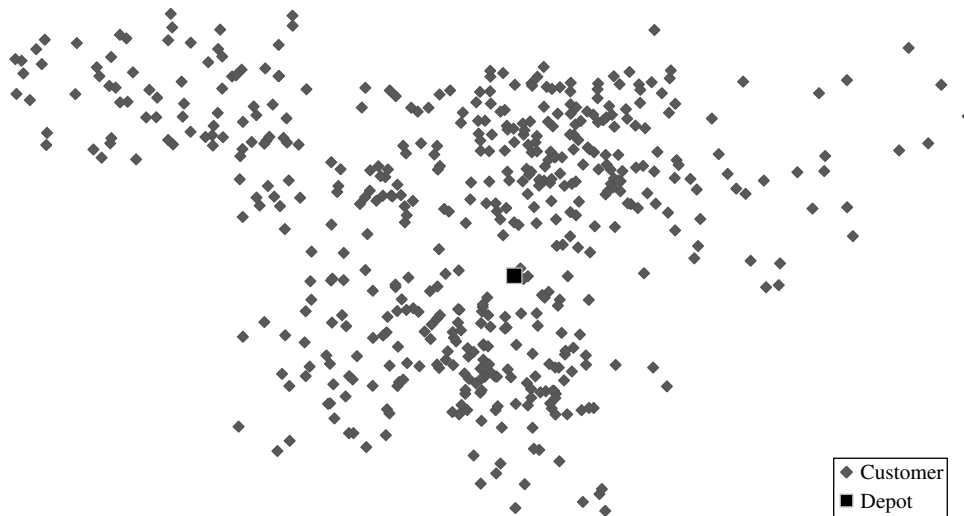


Figure 3 Geographical Distribution in Nijmegen Region

6. Computational Experiments

In this section, we report on the results of a computational study to evaluate the performance of the proposed models and to analyze the impact of various characteristics of the environment on their performance. Recall that the TSMIP is a tactical planning problem with an embedded routing problem on the operational level. We evaluate alternative time slot schedules through simulation. To this end, we generate multiple demand realizations for each schedule and determine the corresponding detailed routing costs using a commercial routing package. To evaluate the impact of characteristics of the environment on the time slot schedules, we vary individual characteristics and analyze the results.

For our study, we use real-life data from Albert.nl. Specifically, we focus on the Nijmegen region, a subset of the service area of Albert.nl consisting of 30 three-digit zip codes with varying demand densities and covering a total area of approximately 1,000 km². Figure 3 shows the geographic distribution of demand in the region. The area is served through a delivery hub in the city of Nijmegen. We consider a typical morning shift and a typical afternoon shift, because they differ in terms of total length, number of time slots, and time slot overlap (see Figure 4).

Furthermore, we use the current service requirements at Albert.nl for the Monday morning and the

Tuesday afternoon. (We also performed simulations for other weekdays and obtained similar results.) The employed service requirements imply that in the morning schedule 24 zip codes have to receive one time slot, three zip codes have to receive three time slots, and three zip codes do not receive any time slot. In the afternoon schedule, 25 zip codes have to receive one time slot and five zip codes have to receive two time slots.

In our experiments we evaluate the following time slot schedules:

- CA: The schedule produced by the continuous approximation approach;
- IP: The schedule produced by the alternative integer programming approach;
- ALBERT: The schedule that is currently in place at Albert.nl;
- ALL: The schedule in which all time slots are made available in all zip codes where Albert.nl currently offers at least one time slot;
- NO-SLOT: The schedule in which a single time slot spanning the entire shift is made available in all zip codes where Albert.nl currently offers at least one time slot.

The schedules ALL and NO-SLOT are included to provide bounds on the performance and to provide insights into the cost-service trade-off. Note that these two schedules do not use the service requirements of

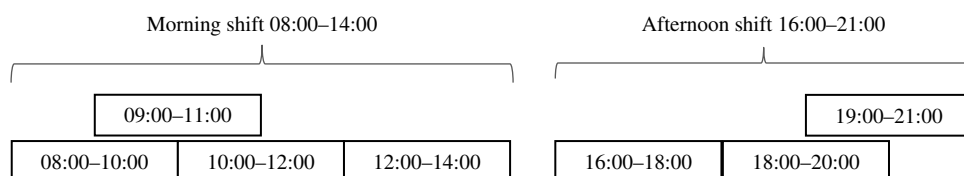


Figure 4 Albert.nl Time Slots

Albert.nl. Instead, the schedule ALL offers the highest level of customer service, and the schedule NO-SLOT offers the lowest level of customer service. The service levels will be reflected in the delivery costs, so the NO-SLOT schedule serves as a lower bound, albeit a weak lower bound, and the ALL schedule serves as an upper bound.

We supplement the comparison outlined above with values derived from the simulation results of a few randomly generated feasible time slot schedules. More precisely, we randomly generate five feasible time slot schedules, perform a simulation for each of them, and average the statistics obtained from the simulation runs. In our results tables, the resulting values will be labeled with RAND-ASGN.

For each problem instance, we generate 20 random demand instances for our simulations as follows. We start with a list of 448 customer addresses in the Nijmegen area. For each demand instance, we randomly select a number of addresses from this list for each zip code, based on the expected number of orders per time slot in that zip code. Specifically, we decide randomly and independently for each address whether or not to select it. We set the probability of selection of an address equal to the ratio between the expected number of orders in the zip code to which that address belongs and the total number of addresses in that zip code. Therefore, the total number of orders varies across demand realizations, with an average equal to the number of orders that was used as input to the optimization models.

For 20 different streams of random numbers, this results in 20 different random sets of addresses. We use the same sets for each schedule, in order to reduce variability across schedules to obtain more comparable results. Using this approach, demand realizations for different time slot schedules differ only in terms of the time slots associated with each order. For example, if time slot schedule CA has slots 2 and 4 open and time slot schedule IP has slots 3 and 4 open, then the same customer addresses have to be visited in slots 2 (CA) and 3 (IP) and in slots 4 (CA) and 4 (IP). For the ALL schedules, we determine the time slot associated with an order by means of a uniform random draw. For the NO-SLOT schedule, all orders are associated with a single time slot spanning the entire shift.

The continuous approximation model is implemented in Excel with the heuristic implemented in VBA. The time slot schedules were produced on an Intel Pentium M 1.6 GHz machine. The heuristic takes about two minutes to converge. The integer programming model is implemented in AMPL with CPLEX 9.0 as the solver. The time slot schedules were produced on an Intel Pentium D CPU 3.20 GHz ($\times 2$) machine. A limit of eight hours of CPU time was imposed.

The limit was always reached without proving optimality. We used Shortec version 7.3.2.1, the routing tool from ORTEC (<http://www.ortec.com>), to construct the delivery routes and determine associated costs for each demand realization. Simulation experiments were run on an Intel Xeon 3.0 GHz machine and take approximately 25 minutes of computing time for 20 demand realizations of a time slot schedule.

6.1. Model Comparison

The first set of experiments is aimed at assessing differences in performance, if any, of the different time slot schedules. We compute the following statistics to compare the different time slots schedules: the average cost, the average number of vehicles used, the average distance, and the average total time (where the averages are taken over the 20 demand realizations). For convenience, we also display the percentage cost savings relative to ALL and provide the widths of the 95% confidence intervals for these savings. For confidentiality, the cost figures are normalized by setting the cost for ALL for the morning shift to 100.

The results of the simulation experiments can be found in Tables 2 and 3. We observe that the gap between the upper bound (ALL) and the lower bound (NO-SLOTS) in terms of average cost is substantial. The absence of time slots in NO-SLOTS allows for more cost-effective delivery routes but, of course, is inconvenient for the customers as they have to be home during the entire morning or afternoon. This illustrates the core trade-off between service and delivery cost and is in line with the results of Punakivi and Saranen (2001) and Lin and Mahmassani (2002). Time slot schedules provide a means for managing

Table 2 Schedule Comparison—Morning

	No. of vehicles	Time	Distance	Cost	Savings (%)	CI (%)
ALL	5.3	1,399	443	100	—	
RAND-ASG	4.8	1,348	408	94	5.6	0.9
ALBERT	4.5	1,361	384	92	7.5	2.3
CA	4.3	1,328	377	90	9.9	1.5
IP	4.6	1,310	393	91	8.9	1.7
NO-SLOTS	3.9	1,210	329	82	18.4	1.2

Table 3 Schedule Comparison—Afternoon

	No. of vehicles	Time	Distance	Cost	Savings (%)	CI (%)
ALL	5.4	1,352	444	98	—	
RAND-ASG	5.1	1,310	417	94	4.3	1.1
ALBERT	4.6	1,266	400	89	8.9	1.4
CA	4.5	1,277	385	89	9.5	2.2
IP	4.6	1,260	386	89	9.6	2.2
NO-SLOTS	3.9	1,171	329	80	19.0	1.4

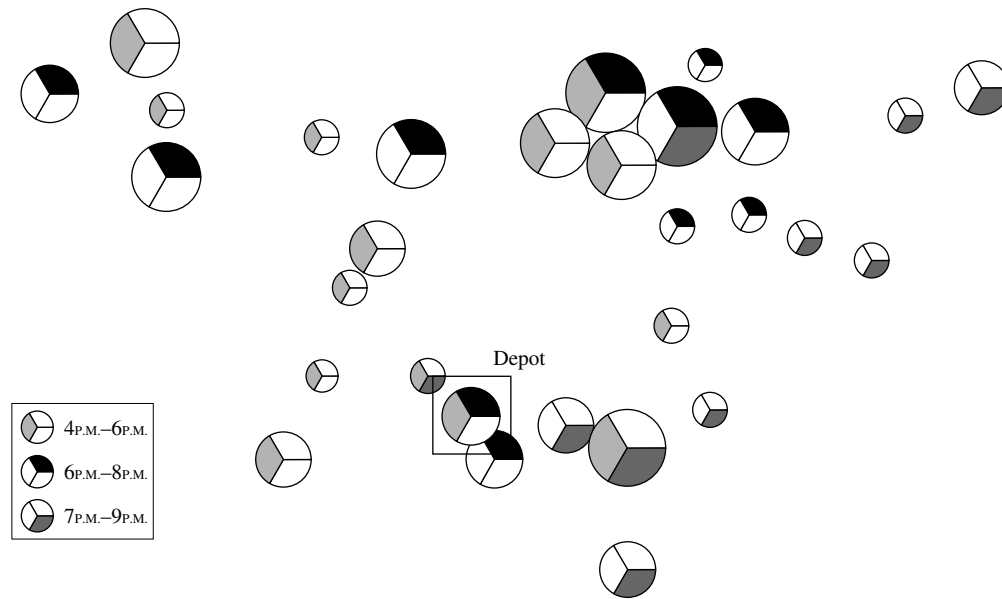


Figure 5 Afternoon Solution CA

this trade-off by concentrating demand while still providing acceptable service levels.

Next, we observe that the morning and afternoon schedules currently in place at Albert.nl (ALBERT) are improved on average by both CA and IP but that the savings are not statistically significant. Albert’s current schedule seems to perform quite well. We also see that randomly constructed time slot schedules (RAND-ASGN) already yield a noticeable improvement over simply offering all time slots (ALL) but do not achieve the quality of Albert.nl’s current time slot schedule and that of the time slot schedules produced

by our optimization models. We also observe, not surprisingly, that the average number of vehicles required in NO-SLOT is less than the average number of vehicles required by the optimized schedules, which in turn is less than the average number of vehicles required by ALL (and RAND-ASGN). This again reflects the trade-off between efficiency and service.

In addition to comparing performance differences, it is interesting to compare the actual time slot schedules. Figures 5 and 6 show the afternoon time slot schedules produced by the continuous approximation model and the integer programming model,

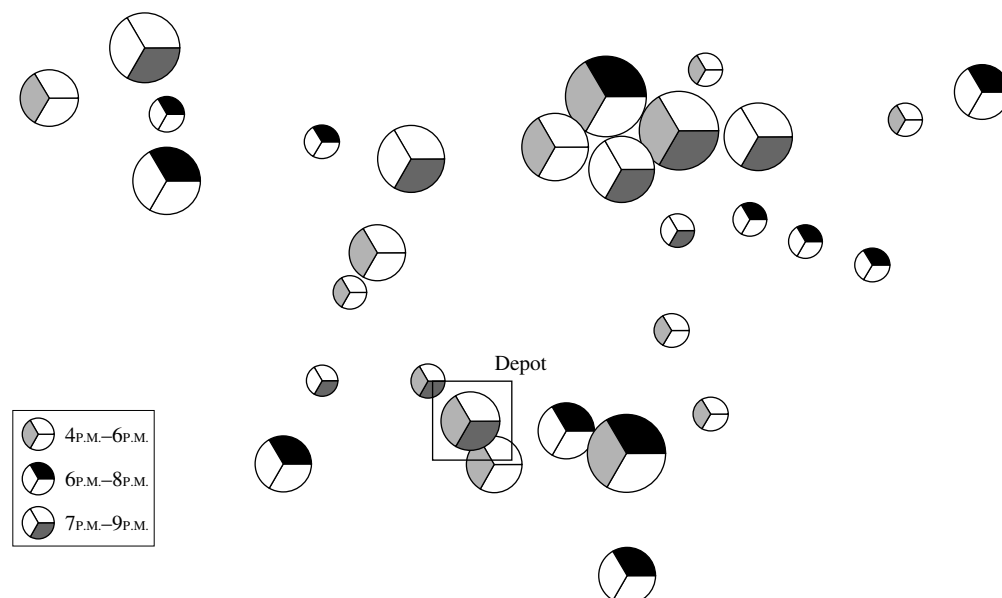


Figure 6 Afternoon Solution IP

respectively. For each zip code, the figure shows the assigned time slot(s). We use the size of the circle to reflect the number of expected orders for the zip code.

We observe that the distinct modeling paradigms of the CA and the IP approach are reflected in different characteristics of the resulting schedules. In Figure 5, we see that the continuous approximation model assigns the same time slot to zip codes that are geographically clustered together. This is especially the case for the zip codes that receive the late afternoon 19.00–21.00 time slot. In contrast, we see in Figure 6 that the integer programming model does not necessarily cluster geographically close zip codes, but forms “vehicle routes.” A possible explanation for the observed behavior is the fact that the continuous approximation approach aims to increase the demand density within a time slot for each zip code. Therefore, it tends to construct large zip code clusters for each time slot. Because the integer programming model explicitly considers the individual vehicle routes, it is more concerned with the zip code groups that can be visited by individual vehicles over the entire time period. From an application perspective, these differences are interesting, but the CA approach yields much more practical run times.

We conclude from the above experiments that our CA model achieves our main objective, namely, supporting the design of effective time slot schedules in real-life scenarios. By automating the generation of time slot schedules the models significantly reduce the planning effort, compared to the current practice of manual planning as well as the alternative IP-based approach. In what follows, we consider what-if analyses of many different scenarios and discuss the insights that they yield.

6.2. Changes in the Environment

As the e-grocery channel is still a relatively new distribution channel, understanding the impact of characteristics of the environment on its performance is of enormous value to companies experimenting with it. Therefore, the second part of our computational study focuses on these issues. All of the percentage differences reported are relative to the costs found in Tables 2 and 3 for the same schedule type.

6.2.1. Demand. The e-grocery market is experiencing huge annual growth rates. Therefore, it is interesting to study the impact of an increase in demand on the resulting time slot schedules and their performance. We consider a scenario in which the service requirements remain unchanged, but every zip code in the existing service area is experiencing a 30% increase in demand. The results can be found in Tables 4 and 5. (We omit time slot schedule ALBERT from these and further experiments as it was not designed for these environments.)

Table 4 Demand Increase—Morning

	No. of vehicles	Time	Distance	Cost	Δ Cost (%)
ALL	6.6	1,734	522	123	23.00
RAND-ASG	5.9	1,677	485	116	23.40
CA	5.5	1,618	451	110	22.22
IP	5.8	1,664	465	114	25.27
NO-SLOTS	5.0	1,511	386	101	23.17

Table 5 Demand Increase—Afternoon

	No. of vehicles	Time	Distance	Cost	Δ Cost (%)
ALL	6.7	1,665	519	120	22.45
RAND-ASG	6.2	1,607	485	114	21.28
CA	5.9	1,570	467	110	23.60
IP	6.0	1,574	462	111	24.72
NO-SLOTS	4.8	1,459	386	98	22.50

We observe a cost increase of a little over 20% for a demand increase of 30%, which demonstrates, as expected, that there are economies of scale. This supports the importance of growth in order for a company to become more profitable.

6.2.2. Vehicle Capacity. An important consideration when setting up any delivery operation is the vehicle fleet and mix. As Albert.nl wants to operate a homogeneous fleet of vehicles, we focus on the impact of vehicle capacity. We consider two scenarios: one in which larger vehicles are used, i.e., the vehicle capacity is increased by 25%, and one in which smaller vehicles are used, i.e., the vehicle capacity is decreased by 25%. The results can be found in Tables 6 and 7.

Not surprisingly, a capacity reduction results in a cost increase whereas a capacity expansion results in a cost decrease. What is interesting, however, is that the cost penalty of a capacity reduction is much stronger than the cost benefit of additional capacity.

Table 6 Vehicle Capacity Change—Morning

	No. of vehicles	Time	Distance	Cost	Δ Cost (%)
–25%					
ALL	5.7	1,433.4	456	103	3.00
RAND-ASG	5.4	1,402.1	433	100	6.38
CA	5.3	1,414.8	414	99	10.00
IP	5.3	1,435.3	428	101	10.99
NO-SLOTS	5.2	1,282.4	370	91	10.98
+25%					
ALL	5.3	1,394.9	440	100	0.00
RAND-ASG	4.7	1,342.1	407	94	0.00
CA	4.1	1,287.6	371	87	–3.33
IP	4.5	1,328.5	394	91	0.00
NO-SLOTS	3.5	1,183.7	311	78	–4.88

Table 7 Vehicle Capacity Change—Afternoon

	No. of vehicles	Time	Distance	Cost	Δ Cost (%)
-25%					
ALL	5.5	1,355	444	99	1.02
RAND-ASG	5.3	1,329	428	96	2.13
CA	4.9	1,319	418	94	5.62
IP	5.1	1,324	416	95	6.74
NO-SLOTS	4.7	1,220	361	86	7.50
+25%					
ALL	5.4	1344	437	97	-1.02
RAND-ASG	5.0	1,303	412	93	-1.06
CA	4.5	1,271	401	89	0.00
IP	4.7	1,256	383	89	0.00
NO-SLOTS	3.8	1,165	327	79	-1.25

This suggests a decreasing marginal value of capacity. To understand this, note that the physical vehicle capacity is only one of the factors that limits the number of orders that a vehicle can deliver. In particular, recall from our models that time constraints, based on the length of a time slot and of a shift, form another constraining factor. The above results suggest that these time constraints become more relevant with increasing vehicle capacity.

The extent to which vehicle capacity is binding depends on the shift and on the service level. The optimization-based time slot schedules and the NO-SLOT time slot schedule appear to be more sensitive to capacity changes because they can better exploit capacity, thus achieving a higher utilization. Moreover, the morning shift seems to be more sensitive to a reduction in vehicle capacity (10% versus 7%), most likely because it covers a period of six hours, as compared to five hours in the afternoon, whereby time constraints are less tight.

6.2.3. Service Level. A key challenge for e-grocers is to find a proper trade-off between service level and delivery costs. A higher service level typically comes at the expense of higher delivery costs. We clearly see this when comparing the simulations results for the NO-SLOTS schedules with those for the schedules that use the service requirements of Albert.nl and with those of the ALL slots schedules. We now investigate this issue further by varying the imposed service requirements.

We consider two scenarios. In the first, we increase the required service level by offering more time slots, more specifically by offering one additional time slot in each zip code except in those zip codes where all time slots are already offered. In the second, we reduce the service level by offering fewer time slots, more specifically by offering one fewer time slot in each zip code except in those zip codes where only a single time slot is currently offered. Note that in

Table 8 Service Level Adjustments—Morning

	No. of vehicles	Time	Distance	Cost	Δ Cost (%)
Reduce service					
ALL	5.3	1,399	443	100	0.00
RAND-ASG	4.9	1,376	412	96	2.13
CA	4.4	1,325	387	91	1.11
IP	4.5	1,341	402	92	1.10
NO-SLOTS	3.9	1,210	329	82	0.00
Increase service					
ALL	5.3	1,399	443	100	0.00
RAND-ASG	5.0	1,418	425	99	5.32
CA	5.1	1,384	416	97	7.78
IP	4.9	1,434	432	99	8.79
NO-SLOTS	3.9	1,210	329	82	0.00

the latter scenario, the changes are minor as most zip codes require only a single time slot to be offered in the base case.

Note also that these changes do not affect the ALL and NO-SLOT schedules because they do not use the service requirements. We only recall their performance as a point of reference. The results can be found in Tables 8 and 9.

As expected, the impact on cost of increasing the service level is much higher than the impact of reducing the service level, as far fewer changes were made to the service level requirements. What may be more surprising is that for the morning shift a reduction of the service level actually leads to a cost increase! This seems counterintuitive, because one would expect to gain operational efficiency from a decrease in customer service. However, we have to realize that we only reduced the number of time slots offered in zip codes in which we used to offer more than one time slot. These zip codes tend to be urban areas with a relatively high demand density. A reduction in the number of time slots offered forces demand to be concentrated in fewer time slots, which results in a less

Table 9 Service Level Adjustments—Afternoon

	No. of vehicles	Time	Distance	Cost	Δ Cost (%)
Reduce service					
ALL	5.4	1,352	444	98	0.00
RAND-ASG	5.1	1,318	412	94	0.00
CA	4.5	1,265	397	89	0.00
IP	4.5	1,274	387	89	0.00
NO-SLOTS	3.9	1,171	329	80	0.00
Increase service					
ALL	5.4	1,352	444	98	0.00
RAND-ASG	5.1	1,385	439	98	4.26
CA	5.0	1,385	443	98	10.11
IP	5.1	1,342	412	95	6.74
NO-SLOTS	3.9	1,171	329	80	0.00

balanced workload and thereby in a lower capacity utilization.

This example highlights two opposing effects. Capacity utilization benefits from a well-balanced workload and therefore from spreading demand over multiple time slots. However, the distance between customers decreases when clustering demand in fewer slots. The interplay between both of these effects results in complex trade-offs, which are very hard to make without any analytical support tools.

In both the morning and afternoon shifts, we do see increases in cost associated with improving the service level. This is not surprising but shows a valuable aspect of developing such models. They reveal the cost associated with such changes, so companies can make better decisions with regard to deciding the level of service to offer to their customers.

6.2.4. Time Slot Template. The set of time slots that may be offered, known as the time slot template, has overlapping time slots in both shifts (see Figure 4). A time slot template with overlapping time slots may be attractive for both the delivery company and the customer. Overlapping slots provide more routing flexibility to the company than shorter nonoverlapping slots. At the same time, they provide a higher service to the customer than a single overall slot. Note that from a modeling perspective the overlap is challenging.

We investigate the impact of overlapping time slots by analyzing alternative time slot templates. We consider two such time slot templates. The first uses time slots with smaller widths that do not overlap, i.e., 8:00–9:20, 9:20–10:40, 10:40–12:00, 12:00–14:00 for the morning and 16:00–18:00, 18:00–19:30, 19:30–21:00 for the afternoon. The second uses different overlapping two-hour time slots, i.e., 8:00–10:00, 9:30–11:30, 11:00–13:00, 12:00–14:00 for the morning and 16:00–18:00, 17:30–19:30, 19:00–21:00 for the afternoon. The results for these different time slot templates can be found in Tables 10 and 11. Note that the different templates do not affect the NO-SLOTS scenario, which is just reported as a benchmark.

As expected, we see that the use of nonoverlapping time slots with smaller widths translates in increased costs, especially in the morning shift. The results for the alternative overlap template are less conclusive. In general, the cost effects appear to be smaller than for removing the overlap. Note that the total amount of overlap, the number of hours which fall into more than one slot, is the same as in the original templates, however distributed differently.

The overlap was a difficult issue in terms of modeling for both CA and IP. The continuous approximation model heuristically adjusts the individual time slot lengths based on the order volumes and virtually removes the overlap. Conceptually, the IP model

Table 10 Time Slot Template Changes—Morning

	No. of vehicles	Time	Distance	Cost	(%)
No overlap					
ALL	5.7	1,469	480	106	6.00
RAND-ASG	5.2	1,434	446	101	7.45
CA	4.5	1,430	401	96	6.67
IP	4.6	1,394	416	96	5.49
NO-SLOTS	4.0	1,218	330	82	0.00
Alternative overlap					
ALL	5.1	1,391	435	98	−2.00
RAND-ASG	4.7	1,353	409	94	0.00
CA	4.4	1,325	391	91	1.11
IP	4.7	1,350	408	94	3.30
NO-SLOTS	4.0	1,218	330	82	0.00

Table 11 Time Slot Template Changes—Afternoon

	No. of vehicles	Time	Distance	Cost	(%)
No overlap					
ALL	6.0	1,391	467	103	5.10
RAND-ASG	5.3	1,339	434	97	3.19
CA	4.7	1,302	406	92	3.37
IP	4.8	1,291	403	92	3.37
NO-SLOTS	3.9	1,171	329	80	0.00
Alternative overlap					
ALL	5.4	1,351	440	98	0.00
RAND-ASG	4.8	1,293	407	92	−2.13
CA	4.8	1,270	403	90	1.12
IP	4.6	1,279	399	90	1.12
NO-SLOTS	3.9	1,171	329	80	0.00

deals with the issue in a more accurate way by taking into account the exact timing of a zip code delivery. However, this comes at the cost of an increased problem size and it introduces symmetry in the model. But, we see that the overlap does translate to lower cost solutions, so it is important to consider it.

7. Conclusions

In this paper, we introduce the time slot management problem (TSMP), a novel tactical planning problem which is relevant to many businesses that offer attended home delivery, especially e-grocery. The problem entails the decision of which time slots to offer in each zip code so as to minimize expected delivery costs. The use of time slots gives rise to complex interactions between marketing and operational considerations and requires sophisticated decision support tools.

This paper presents a fully automated approach that is capable of producing high-quality solutions within a short amount of time. The approach, which is based on continuous approximation, estimates the route costs resulting from the time slot selections per zip code based on “local” information. Our numerical

experiments on real-life data from Albert.nl show the following:

- Although narrow delivery time slots are convenient for the customer, they greatly reduce routing efficiency. Our experiments indicate an increase in delivery cost of up to 25% from using two-hour time slots instead of time slots that span an entire morning or afternoon. This illustrates the core trade-off between service and delivery costs.

- The regional differentiation of the delivery time slot offering leads to significant cost savings as compared to simply offering the full range of time slots in the whole delivery region. In particular, the results from our numerical experiments suggest up to 10% delivery cost savings.

- There are clear economies of scale in the delivery operation. The results show potential cost benefits from increasing the number of stops within the current delivery area. This illustrates the importance of growth in order for an e-tailer to become more profitable.

- The time slot constraints and the constraints on physical vehicle capacity interact. Time slot management decisions become more relevant when the vehicle capacity allows a vehicle tour to span several time slots.

- The main trade-off in the time slot schedule design is between demand clustering and demand smoothing. The advantage of clustering demand is that it minimizes the distance between successive stops, and thus travel costs. The disadvantage is the potential underutilization of the vehicle capacity, and a corresponding increase in vehicle costs.

There remain several interesting directions for future research. We see the joint optimization of the service requirements and the time slot schedule as one of the most challenging. This will not only require a more complete understanding of customer behavior, but also more sophisticated solution approaches. We would also be interested in further investigating the IP approach to see if we can find high-quality solutions in a shorter amount of time. In general, we see great potential for additional research on the real-time management of demand and the use of incentives in attended home delivery.

References

Agatz, N. A. H. 2009. Demand management in e-fulfillment. Erim Ph.D. Series Research in Management, Rotterdam, The Netherlands, <http://hdl.handle.net/1765/15425>.

- Agatz, N. A. H., A. M. Campbell, M. Fleischmann, M. Savelsbergh. 2008a. Challenges and opportunities in attended home delivery. B. Golden, S. Raghavan, E. Wasil, eds. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer Science & Business Media, New York, 379–396.
- Agatz, N. A. H., M. Fleischmann, J. A. E. van Nunen. 2008b. E-fulfillment and multi-channel distribution—A review. *Eur. J. Oper. Res.* **187**(2) 339–356.
- Bent, R., P. Van Hentenryck. 2004. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Oper. Res.* **52**(6) 977–987.
- Berning, C., S. Ernst, N. H. Hooker. 2005. Are e-grocers serving the right markets? *Choices* **20**(1) 77–80.
- Bertsimas, D. J., P. Jaillet, A. R. Odoni. 1990. A priori optimization. *Oper. Res.* **38**(6) 1019–1033.
- Braysy, I., M. Gendreau. 2005a. Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Sci.* **39**(1) 104–118.
- Braysy, I., M. Gendreau. 2005b. Vehicle routing problem with time windows, part II: Metaheuristics. *Transportation Sci.* **39**(1) 119–139.
- Campbell, A. M., M. P. W. Savelsbergh. 2005a. Decision support for consumer direct grocery initiatives. *Transportation Sci.* **39**(3) 313–327.
- Campbell, A. M., M. P. W. Savelsbergh. 2005b. Incentive schemes for attended home delivery services. *Transportation Sci.* **40**(3) 327–341.
- Daganzo, C. F. 1987a. Modelling distribution problems with time windows: Part 1. *Transportation Sci.* **21**(3) 171–179.
- Daganzo, C. F. 1987b. Modelling distribution problems with time windows. part 2: Two customer types. *Transportation Sci.* **21**(3) 180–187.
- Daganzo, C. F. 2005. *Logistics Systems Analysis*, 4th ed. Springer, Berlin.
- Dror, M., G. Laporte, P. Trudeau. 1989. Vehicle-routing with stochastic demands—Properties and solution frameworks. *Transportation Sci.* **23**(3) 166–176.
- Fisher, M. L., R. Jaikumar. 1981. A generalized assignment heuristic for vehicle-routing. *Networks* **11**(2) 109–124.
- Francis, P., K. Smilowitz. 2006. Modeling techniques for periodic vehicle routing problems. *Transportation Res. Part B: Methodological* **40**(10) 872–884.
- Francis, P., K. Smilowitz, M. Tzur. 2006. The period vehicle routing problem with service choice. *Transportation Sci.* **40**(4) 439–454.
- Gendreau, M., J. Potvin. 1998. Dynamic vehicle routing and dispatching. T. Crainic, G. Laporte, eds. *Fleet Management and Logistics*. Kluwer, Norwell, MA, 115–126.
- Hvattum, L. M., A. Lokketangen, G. Laporte. 2006. Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. *Transportation Sci.* **40**(4) 421–438.
- Jaillet, P., A. R. Odoni. 1988. The probabilistic vehicle routing problem. B. Golden, A. Assad, eds. *Vehicle Routing Methods and Studies*. Elsevier, Amsterdam, 293–318.
- Kenyon, A. S., D. P. Morton. 2003. Stochastic vehicle routing with random travel times. *Transportation Sci.* **37**(1) 69–82.
- Langevin, A., P. Mbaraga, J. F. Campbell. 1996. Continuous approximation models in freight distribution: An overview. *Transportation Res. Part B—Methodological* **30**(3) 163–188.
- Lin, I. I., H. S. Mahmassani. 2002. Can online grocers deliver? Some logistics considerations. *Transportation Res. Record* **1817** 17–24.
- Punakivi, M., J. Saranen. 2001. Identifying the success factors in e-grocery home delivery. *Internat. J. Retail Distribution Management* **29**(4) 156–163.