Attended Home Delivery Services

ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics
Econometrics and Operational Research

July 3, 2017

Derek Zoutendijk
411370

Supervisor: MSc Thomas Visser
Second assessor: Dr. Remy Spliet

Abstract

Attended home delivery services are gaining popularity, thus improving efficiency of deliveries is becoming more important. In this research, we investigate two methods to increase profits through reduction of delivery costs. These methods consist of offering incentives to customers to influence their choice behavior on selecting delivery times. We use simulations to evaluate the effects of offering incentives. We conclude with insights obtained from these simulations.

1 Introduction

Attended home delivery services are commonly used these days and are still gaining popularity. Online stores started to arise around the year 2000 and the online commerce is predicted to show an annual growth rate of 11% in Western Europe (Ehmke (2012)). Despite these immense growth figures, little research has been carried out on efficient attended home delivery services, whilst large improvements in costs can be made for these services. An example of these services is the delivery of groceries by Albert Heijn, a Dutch supermarket. Customers order their groceries online and have them delivered at home. It is important that the customer is present at the time of this delivery as customers might need the groceries the same day of delivery. To ensure customers are home at the time of delivery, Albert Heijn lets the customer select an hour on the day, a time slot or window, in which they can receive delivery.

In this research we will focus on methods to encourage customers to select certain time windows to decrease travel time for the delivery trucks and thus decrease costs. As these incentives have to be calculated very quickly when a customer wants to order, a method with little calculation time is preferred. In this research, two methods are investigated. The two methods are proposed by Campbell and Savelsbergh (2006). In the first method, incentives are given for one-hour time slots and customers can only select one time slot. The second method introduces two-hour time windows and incentives are given to customers to choose these wider time slots.

First we will give a literature review of past researches on attended home deliveries in Section 2. Then, in Section 3, we describe our problem and introduce variables and parameters. In Section 4 we explain all the methods used in this research. Section 5 consists of a description of the problem instances and the results. At last, we will give conclusions in Section 6 and a final discussion in Section 7.

2 Literature review

Improving efficiency in delivery for attended home delivery has little been researched. Maximizing profit was already investigated in Campbell and Savelsbergh (2005). This was done by deciding whether an order should be accepted or rejected and which time slots to present to customers. Campbell and Savelsbergh (2006) researches offering incentives to influence customers to choose certain time windows. In their work, they propose two models to influence customers. They do not reject orders if it is possible to accept them. It is showed that incentives can make large improvements in profits.

Agatz, Campbell, Fleischmann and Savelsbergh (2008) study the effect of offering only certain time slots to customers. They offer a different set of time windows to customers from different zip codes. The two used optimization methods result in a substantial decrease in delivery costs. Ehmke and Mattfeld (2012) focuses more on the routing part of attended home delivery services. They implement time-dependent vehicle routing models to improve efficiency of delivery in cities.

3 Problem description

When a customer arrives with an order, we present the time slots that the customer can select. We only present time windows in which we are able to deliver to the customer, given all accepted orders. Based on the location of the customer, we offer incentives in the shape of a small amount of money to influence customers choice behavior with the purpose of decreasing delivery costs. Then, if a customer decides to select a time slot, we accept the order and are now obligated to deliver to the customer. This is repeated until we cannot accept anymore orders or there are no more customers who want to order.

3.1 Basic Time Slot Incentives

In the first model (Campbell and Savelsbergh (2006)) customers can select from a number of offered time windows of one hour, non-overlapping. We call this model the Basic Time Slot Incentives (BTSI). Let T be the set of all time slots t. We assume that the probability p_i^t that a customer i selects a certain time window t is known. Also, we assume that the effects of incentives on these probabilities are also known. Namely, the probability of choosing a time window increases by the incentive I^t times a rate x and the probabilities of selecting other time slots that are nonzero will all decrease with an equal amount z such that the sum of probabilities remains 1.

If it not possible to deliver the order to a customer in a certain time window, we will not present the time slot to this customer. We take two scenarios in consideration to handle this. In the first the customer would never want to select a time slot that is not presented. If this is the case, the probability of selecting that time window will be equally distributed amongst all positive probabilities. This gives new probabilities q_i^t. In the other scenario the customer abandons if he or she wants to select this slot. Furthermore, we consider only next day deliveries. Lastly, the delivery time t_{ij} is assumed to be linear to the distance between two points i and j. It takes time to handle the delivery after arriving: the service time s. The vehicles have a capacity Q, but this is assumed to not be constraining in our models, as in (Campbell and Savelsbergh (2006)). To increase profits, consisting of revenues r_i obtained from an order i with demand d_i, we offer the incentives I^t to time windows t. Profits are calculated as follows:

$$\sum_{i \in A} (r_i - \tilde{I}_i - t_{i,i+1}). \tag{1}$$

The set A represents the set of customers of which the order was accepted. The set is ordered in the same order as the customers are serviced in the final delivery schedule. Therefore, the summation over $t_{i,i+1}$ in equation (1) represent the delivery costs. Also, the depot is added to this set twice, once at the start and once at the end to incorporate the travel cost from and to the depot. For this depot holds r_i $= 0$ and $\tilde{I}_i = 0$. $\tilde{I}_i$ is the incentive that was offered for the time slot that was selected by customer i. If $\tilde{I}_i = 0$, no incentive was offered.

3.2 Wider Time Slot Incentives

For the second model with the wider time slots (Campbell and Savelsbergh (2006)), the problem changes slightly. We refer to this model as the Wider Time Slot Incentives (WTSI). Besides the previous time windows, customers can now also select two-hour time windows, overlapping by one hour each time. This means we have one less wider time slot than we have basic one-hour time slots. The probability that a customer selects a two-hour time window is initially zero but increases by an incentive $I^{(t,t+1)}$, with t and $t+1$ consecutive time windows, times a rate x if this incentive is given. This results in an equal decrease of z in probabilities of choosing any one-hour time slot.

4 Methodology

To improve the attended home delivery services in the sense of profit, two models are proposed by Campbell and Savelsbergh (2006). We will implement both models. The models consist of two parts, a routing problem and a profit optimization part via offering incentives. First, we will describe the routing problem. This problem consists of two subproblems. In the first problem, we create several delivery schedules each time we accept an order. The second problem arises when a customer arrives. We then calculate C^t based on the delivery schedules obtained in the first problem. How this C^t is calculated, will be explained in Section 4.1.1. Next we will solve the profit optimization part via offering incentives. The customer then selects a time slot and we repeat this cycle again, starting with the creation of delivery schedules. This cycle continues until no more orders can be accepted. Finally, the impact of offering incentives on total profit via the linear program is determined using computational experiments.

To evaluate this impact, we use the following methods. In the *NoIncentives* method we will not give any incentives, this will be used to examine effects of giving incentives at all. The *FlatIncentives* method consists of offering the same incentives to some time windows and is used to compare with the effect of using more advanced methods to determine incentives. Our *MainIncentives* method uses linear programming to decide on what incentives to give to the time windows that have the lowest C^t. The *BestIncentives* method uses the same linear program, but not only for the time slots with the lowest C^t. We decide on incentives for all combinations of feasible time windows up to a number of m slots. This is time-consuming but is useful to determine how much profit is lost if *BestIncentives* is used to decrease computation times. The *BestCase* method assumes a scenario where the customer will always select the time window with lowest C^t without giving any incentive. This represents an upper bound but can be exceeded by the other methods: the current best time slot may not be the best given future orders.

We investigate two scenarios what happens if a new order is infeasible in a certain time slot: In the *NoAbandonment* case the probabilities for selecting the time windows that will not be presented to the customer are distributed equally amongst all positive probabilities and in the *Abandonment* case we do not distribute these probabilities. If the customer would choose a time slot that is not presented, this customer abandons and the order is lost. For both cases we will implement all models and methods given above. Our methodology is extensive and we have many different variants. Therefore we start with the variants of the model where we offer incentives to customers to select a single one-hour time window and only give the required adjustments on this model to implement the other two models.

4.1 Basic Time Slot Incentives

We will now describe the implementation of the first model. We will refer to this model as Basic Time Slot Incentives (BTSI). Here, customers can only choose a single one-hour time slot and time slots in T are non-overlapping.

4.1.1 Routing

Suppose order O_j is a new order, then we calculate costs of inserting order O_j in the delivery schedules, that we create after accepting each order, for all time windows. We will first describe how we create these delivery schedules. Hereafter, the incentives calculations are explained.

We create S, a set of feasible schedules of only the accepted orders. The first schedule in S is the schedule acquired when the previous order O_{j-1} was accepted with time slot t. How this schedule is acquired will be explained later on. To complete S, the rest of the schedules are built iteratively. We start with an empty schedule and create a candidate list of the c orders that have cheapest insertion costs between orders O_{i-1} and O_i. This is measured by the increase in travel time from inserting O_j between O_i and O_{i-1} in time window t where time window t is $[begin^t, end^t]$.

$$c_{increase} = t_{i-1,j} + t_{j,i} - t_{i-1,i}. \tag{2}$$

with t the travel time. If O_i is the first order or O_{i-1} is the last order in the schedule, the new order will be inserted at the start or end at the schedule and the travel times to the depot are used. The insertion has to be feasible, we can only add this insertion to our candidate list, if the delivery can occur in the time window that was accepted for this order. This is possible if the following holds:

$$e_j \leq l_j. \tag{3}$$

with

$$e_j = \max(e_{i-1} + s + t_{i-1,j}, begin^t),$$

$$l_j = \min(l_i - s - t_{j,i}, end^t).$$

We iterate over all insertion points for all not accepted orders. We keep the insertion point and the order in the candidate list if the insertion is cheaper than an insertion that is in the candidate list, the latter insertion is then removed. When the iterating is complete, we randomly select an order and insertion point from the candidate list and perform the insertion. After insertion, we update the values e_i for all $i \geq j$ and all l_i for $i \leq j$. We repeat this until either all orders are inserted or we cannot feasibly insert anymore orders. If all orders are inserted, we keep this schedule and store it in the set S together with its costs. If a schedule does not contain all orders,it is discarded. We try to create a schedule n times and since S will always contain the schedule acquired after accepting an order, we will later explain how to acquire this schedule, our set S will always have between 1 and $n+1$ schedules.

We can now use the set of schedules S to evaluate the cost of accepting a newly arrived order O_j in a time slot. For every schedule in S we check feasibility and calculate cost for inserting the order j in time window t between every position in every route. The increase in costs are then given by the following formula:

$$\tilde{c} = (t_{i-1,j} + t_{j,i} - t_{i-1,i}) + C(s) - C(*). \tag{4}$$

Here, $C(s)$ denote the costs of the schedule $s \in S$ in which we try to insert order j. $C(*)$ is the least cost schedule of S and the remaining term represents the increase in cost, expressed in travel time, of inserting the new order. Together they represent the true added cost of inserting in a time slot as some schedules are already more expensive. We store all values $\tilde{c}$ per time window t in the set $\tilde{C}_t$. The insertion costs C^t is then as follows.

$$C^t = \min(\tilde{C}_t) \quad \forall t \in T. \tag{5}$$

4.1.2 Incentives

For this part, we will describe incentive calculation methods. This will be done for all five methods *NoIncentives*, *FlatIncentives*, *MainIncentives*, *BestIncentives* and *BestCase* for both scenarios *NoAbandonment* and *Abandonment*.

NoAbandonment
In this scenario a customer i will not abandon if its order can be feasibly inserted in any time window $t \in T$. If there are no time slots presented to the customer for which holds $q_i^t > 0$ the customer does abandon.

NoIncentives
In this method, no incentives are offered. Given q_i^t, the customer randomly selects one of the time windows according to these probabilities. The order is then inserted at the point in the schedule which had lowest cost for this time slot. The time window is stored and the e_i and l_i values will be updated again. This schedule is the least cost schedule that is added to S as first.

FlatIncentives
Here, an equal incentive will be offered to the m feasible time slots with lowest C^t. These time slots form the set U. As offering incentives will increase the probabilities of choosing these time windows, the probabilities of the other time slots will decrease. This decrease will only occur if the time slot has a positive probability. The time windows for which this holds and that are not in U will be in the set V. The maximum amount spend on incentives will then be:

$$u = min(B, \frac{min_{t \in V}(q_i^t)}{x}|V|). \tag{6}$$

We will offer an incentive I^t of u/l to the time sots in U. The probabilities in V will decrease by z, with $z = u/|V|$. If the costs from time windows for which hold $q_i^t > 0$ are all equal, offering incentives cannot decrease costs and thus this will not be done. If U contains less than m time windows, V will be empty and then no incentives will be offered either. With the new probabilities, we inserted the order as in *NoIncentives.*

MainIncentives
For this method, we use the *FlatIncentives* method and extend it. The choices for V and U remain the same. Instead of offering a flat incentive we use a linear program to select incentives for time slots in U that maximize expected profit. The following objective function should be maximized.

$$\sum_{t \in U}(r_i - C^t - I^t)(q_i^t + xI^t) + \sum_{t \in V}(r_i - C^t)(q_i^t - z). \tag{7}$$

This represents the expected profit. The first summation represents the expected profit for time slots that may receive incentives. Order O_i has revenue r_i and insertion costs C^t in time window t. If an incentive is given, this is also subtracted and the profit of inserting order i in time window t remains. This is multiplied by the probability that this insertion occurs, possible enlarged by giving the incentive. For the second term, the profit is simply the expected revenue minus the insertion costs, but the probability, used for the expected revenue, that a time slot t in V is chosen decreases with z. This z is not allowed to be bigger than the smallest probability in V. Also, the total increase in probability for windows in U should be equal to the total decrease of probabilities in V. At last, incentives are not allowed to be bigger than the budget B that is reserved for the incentives. After rewriting the optimization function and removing constant terms, the following optimization problem can be formulated:

$$\max \sum_{t \in U}(x(r_i - C^t) - q_i^t)I^t - \sum_{t \in U} x(I^t)^2 - \sum_{t \in V}(r_i - C^t)z \tag{8}$$

$$s.t. \quad z \leq q_i^t \qquad \forall t \in V, \tag{9}$$

$$\sum_{t\in U} xI^t = z|V|, \tag{10}$$

$$0 \le I^t \le B \quad \forall t \in U. \tag{11}$$

As can be seen, the function contains quadratic terms. Solving quadratic programs can be time consuming, especially in our problem setting. Since we require fast methods, a linear approximation will be used for the quadratic terms $(I^t)^2$; a piecewise linear function over $f-1$ intervals j in J between $I^t = 0$ and $I^t = u$ with u as given in *FlatIncentives*. Substituting this into the objective function and adding constraints to link I^t to the approximation function, we receive this final linear program:

$$\max \sum_{t\in U}(x(r_i - C^t) - q_i^t)I^t - \sum_{t\in U} x((\frac{u}{f-1})^2 y_2^t + (\frac{2u}{f-1})^2 y_3^t + ... + (u)^2 y_f^t) - \sum_{t\in V}(r_i - C^t)z \tag{12}$$

$$s.t. \quad z \le q_i^t \qquad \forall t \in V, \tag{13}$$

$$\sum_{t\in U} xI^t = z|V|, \tag{14}$$

$$0 \le I^t \le B \quad \forall t \in U, \tag{15}$$

$$I^t = \frac{u}{f-1} y_2^t + \frac{2u}{f-1} y_3^t + ... + u y_f^t \quad \forall t \in U, \tag{16}$$

$$0 \le y_j^t \le u \quad \forall t \in U,\ \forall j \in J. \tag{17}$$

If any incentives to time slots in U are zero after the optimization, we remove these slots from U and add them to V. We then optimize the expected profit again with the new sets until all incentives are positive. The probabilities will then change in the same way as they did in *FlatIncentives*.

BestIncentives
This method uses the same linear program as $MainIncentives$. Instead of using the m cheapest insertions for the set U, all possible combinations of 1 to m feasible time windows with possible probabilities are used for U and the linear program is rerun each time a new U has been created. The incentives that then give the highest expected profit are offered.

BestCase
As a benchmark, we consider a method where we do not offer incentives but instead of randomly selecting a feasible time slot, the customer selects the time window with lowest insertion costs of the time slots with $q_i^t > 0$.

Abandonment
In this scenario of abandonment, customers will walk away if they select a time slot that is not presented to them. We do not distribute the probabilities of selecting infeasible time slots amongst the other positive probabilities. We now present the slight adjustments needed for our five methods to incorporate this abandonment.

NoIncentives
This method is the same as the *NoIncentives* in the *NoAbandonment* case, but without the altered probabilities q_i^t.

FlatIncentives
We now remain with infeasible time windows that have a positive probability of being selected. These time windows form the new set denoted by F. The sets U and V are as before, but now the probabilities of selecting windows in F reduce as well. To capture this, we also change u:

$$u' = min(B, \frac{min_{t\in V,F}(q_i^t)}{x}(|V| + |F|). \tag{18}$$

Now, the probabilities in V and F will decrease with $u'/(|V| + |F|)$. Further, this method is the same as *FlatIncentives* in *NoAbandonment*.

MainIncentives

The linear program from *MainIncentives* with *NoAbandonment* is used, but with a small difference. Instead of u, we will use the u' of equation (18) based on the same sets V and F. For the probabilites in the formulation the original values p_i^t are used. There are also some changes required in the constraints of the linear program. Constraint (13) becomes:

$$z \leq p_i^t \qquad \forall t \in V, F. \tag{19}$$

and (14) is now:

$$\sum_{t \in U} xI^t = z(|V| + |F|). \tag{20}$$

BestIncentives
For this method, the same steps as in *BestIncentives* in *NoAbandonment* are followed, but based on the *MainIncentives* method in *Abandoment.* Again, the linear program is run for the different sets of U and the incentives with highest expected profits are chosen.

BestCase
Although it is possible for a customer to select an infeasible time slots, we assume here that customers choose the times lots with lowest insertion costs. This will never be an infeasible time window and thus no changes have to be made.

4.2 Wider Time Slot Incentives

We now explain how the previous model can be adapted to handle wider time slots. In this model (WTSI), customers can choose a single one-hour or two-hour time window.

4.2.1 Routing

For the routing part, no changes have to be made. The larger time slots are captured by a different value for $begin^t$ and end^t. With the routing part, we want to create a list with cheapest insertions for each time window needed in the Incentives part. In addition to the C^t from $BTSI$ we now have $C^{t,t+1}$ too, which is simply the minimum of C^t and C^{t+1}.

4.2.2 Incentives

The probabilities of a customer i choosing a two-hour time slot is initially zero, but can be increased by offering an incentive.

NoAbandonment
In this scenario, a customer i will again not walk away if its order can be feasibly inserted in any time window t such that $q_i^t > 0$. A customer does abandon if all time slots are infeasible.

NoIncentives
As no incentives are given, the probability of choosing a two-hour time slot will remain zero and the method is thus the same as in the case of only small time slots.

FlatIncentives
We now only offer incentives to select two-hour time windows. The set U now consists of two-hour time windows. A two-hour time slot can only be added to U if the two hours from this slot have a positive probability of being chosen as a time window, that is; $q_i^t > 0$ and $q_i^{t+1} > 0$. All one-hour time slots with positive probability are added to V. With these sets U and V, the same steps are followed in calculating the flat incentives and in- and decrease in probabilities.

MainIncentives
As in the *MainIncentives* from TSI, we use the same U and V from *FlatIncentives.* The linear program from TSI has to be adjusted, the expected profit becomes:

$$\sum_{(t,t+1) \in U} (r_i - C^{(t,t+1)}) - I^{(t,t+1)})yI^{(t,t+1)} + \sum_{t \in V} (r_i - C^t)(p_i^t - z). \tag{21}$$

The second summation is the same as before, but the first summation now uses the insertion and incentive costs of the two-hour time slots. We reformulate the function again after removing constant terms and approximating the quadratic terms in the same matter as in TSI. The linear program is then as follows:

$$\max \sum_{(t,t+1)\in U} x(r_i - C^{(t,t+1)})I^{(t,t+1)} - \sum_{(t,t+1)\in U} x((\frac{u}{f-1})^2 y_2^{(t,t+1)} + (\frac{2u}{f-1})^2 y_3^{(t,t+1)} + ... + (u)^2 y_f^{(t,t+1)}) - \sum_{t\in V}(r_i - C^t)z, \quad (22)$$

$$s.t. \quad z \le q_i^t \qquad \forall t \in V, \quad (23)$$

$$\sum_{(t,t+1)\in U} xI^{(t,t+1)} = z|V|, \quad (24)$$

$$0 \le I^{(t,t+1)} \le B \quad \forall (t,t+1) \in U, \quad (25)$$

$$I^{(t,t+1)} = \frac{u}{f-1}y_2^{(t,t+1)} + \frac{2u}{f-1}y_3^{(t,t+1)} + ... + uy_f^{(t,t+1)} \quad \forall (t,t+1) \in U, \quad (26)$$

$$0 \le y_j^{(t,t+1)} \le u \quad \forall (t,t+1) \in U, \ \forall j \in J. \quad (27)$$

The probabilities will then change in the same way as they did in *FlatIncentives* with the new incentives.

BestIncentives
This method is not used in the WTSI, as the *MainIncentives* method already ensures incentives that give highest expected profit. This is because the probabilities of selecting a wider time slot are initially all zero. Offering an incentive to a wider time window will therefore not result in a decrease in expected profit of other wider time windows, which would make the decision based on both insertion costs and initial probabilities. This is not the case, thus enumerating over many different sets U would give the same outcome as *MainIncentives*.

BestCase
Although the probabilities of selecting a two-hour time window is zero without giving incentives, we assume here that it is possible. In the best case, the customer will select a two-hour time window with lowest insertion cost. The customer can only choose this time slot if the two one-hour time slots embedded in the two-hour slot both have a probability larger than zero. If there is no such two-hour window, the customer will select the one-hour time window with cheapest insertion cost.

Abandonment
In the scenario with abandonment, customers will walk away if they select a time slot that is not presented to them. We do not distribute the probabilities of selecting infeasible time slots amongst the other positive probabilities. We now present the slight adjustments needed for our five methods to incorporate this abandonment.

NoIncentives
This method is the same as the *NoIncentives* in the *NoAbandonment* case, but without the altered probabilities q_i^t.

FlatIncentives
This method is the same as the *FlatIncentive* with *Abandonment* in the TSI, but with U as the m two-hour slots with cheapest insertion and V all feasible time slots with positive probability.

MainIncentives
Here, we apply the same changes as done in this method and scenario in TSI, but to *MainIncentives* with *NoAbandonment* in WTSI.

BestIncentives
For the same reasons as in the *BestIncentives* method with *NoAbandonment*, this method will give the same outcomes as *MainIncentives* and is therefore again not used.

BestCase
In the TSI there was no difference between *BestCase* with *Abandonment* and *NoAbandonment*. Here we do have a small difference. As we do not distribute probabilities from infeasible time windows amongst the other probabilities, there are more time windows with a positive probability. If there are two consecutive time slots with a positive probability of which one is infeasible, the two-hour slot would not be viable in the *NoAbandonment* scenario as one of the probabilities would become zero. Now, this time window is possible. Again, the customer chooses the two-hour time slot with cheapest insertion if there is any and selects a one-hour window otherwise.

5 Computational Experiments

We will now perform computational experiments for our methods. First, we give a description of our problem instances. As original instances of Campbell and Savelsbergh (2006) are not available, we simulated our own instances in a similar manner. After this, we present and comment on the results of the experiments.

5.1 Problem Instances

All methods use the same instances. We create 25 instances where a single instance consists of 30 orders. We use only a single truck for the deliveries. Every order has a revenue r_i of €100,-. The delivery location of an order is uniformly distributed over an area of 60 by 60 units. The travel time in minutes is calculated as a Euclidean distance between the two points and the costs are €1,- per minute. For the service time, an estimation of 20 minutes is used. In the routing part, we try to create 50 new schedules (n) and for the creation of these schedules we have a candidate list of 3 orders (c). We assume a maximum budget of €5,- per incentive and the rate in which the probability increases by an incentive (x) of 0,2. For the initial probabilities p_i^t, we use three different patterns. In the first pattern (*Pattern* = 1), each customer has 8 consecutive time windows with equal positive probabilities. These time slots can wrap around the day, for example 5 time windows at the start of the day and 3 slots at the end of the day. The second (*Pattern* = 2) and third (*Pattern* = 3) are almost the same but have one time window that has a two or three times larger probability. We also try a different number of time slots we offer an incentive to; $|U|$ = 1,..,4. In the linear approximation we use 5 intervals (f = 6).

We also perform our computational experiments for other values of some parameters so the effects of our incentive methods can be evaluated more accurately. This is only done for the methods with *NoAbandonment*. We will start with the basic results, thus with parameters given before. We then see effects of having only 15 orders arrive instead of 30 orders. After this, we show results when customers only have 4 consecutive time slots with initial positive probabilities . Then we show differences in outcome if a maximum budget per incentive of €2,- is used. Hereafter, we evaluate the effect of a rate x equal to 0,1 instead of 0,2. At last, we present the results of methods with *Abandonment*.

In Section 5.2.3, we present the average calculation times of all methods for both models and scenarios. This calculation time includes both the routing part and the incentives calculations and can be used to evaluate efficiency of the methods. The times are given in seconds.

5.2 Results

We now give the results of the problem instances given above. This will be done for every model separately.

5.2.1 Basic Time Slot Incentives

In Table 1 we present the findings of all methods with *NoAbandonment* from the BTSI model of Section 4.1. For this table, the basic problem instances are used.

Table 1: Profits BTSI methods with *NoAbandonment*

NoAbandonment	$\|U\|$	NoIncentives	FlatIncentives	MainIncentives	BestIncentives	BestCase
Pattern = 1	1	1226,76	1336,21	1390,25	1356,62	1305,14
	2	1226,76	1409,54	1363,69	1383,35	1305,14
	3	1226,76	1380,14	1344,14	1380,86	1305,14
	4	1226,76	1304,20	1343,35	1387,66	1305,14
	Average	1226,76	1357,52	1360,36	1377,12	1305,14
	% improvement	0,00	10,66	10,89	12,66	6,39
Pattern = 2	1	1215,09	1361,26	1375,52	1372,57	1305,14
	2	1215,09	1368,32	1357,44	1353,15	1305,14
	3	1215,09	1316,56	1292,20	1361,66	1305,14
	4	1215,09	1236,24	1290,25	1352,05	1305,14
	Average	1215,09	1320,60	1328,85	1359,86	1305,14
	% improvement	0,00	8,68	9,36	11,91	7,40
Pattern = 3	1	1207,95	1350,40	1363,34	1372,75	1305,14
	2	1207,95	1349,98	1394,69	1362,73	1305,14
	3	1207,95	1314,89	1316,81	1385,31	1305,14
	4	1207,95	1253,96	1376,12	1398,06	1305,14
	Average	1207,95	1317,31	1362,74	1379,72	1305,14
	% improvement	0,00	9,05	12,82	14,22	8,05

In Table 1 we presented the profits for all different *Pattern* and the different values for $|U|$. Also, the average profit over all values of $|U|$ is given for every method and *Pattern*. We also gave the percentage increase of these averages relative to the averages for the *NoIncentives* method. This allows us to make quick comparisons between the methods. All methods where incentives are offered to customers improve the profits with around 10%. The *MainIncentives* method provides higher profits than *FlatIncentives*, but the differences are quite small. The same holds for *BestIncentives* and *MainIncentives*, *BestIncentives* provides slightly higher profits. We also see that the *BestCase* is often surpassed by the methods with incentives. If we take averages over the different values of $|U|$, the highest average profit is found for $|U|$ = 2, followed closely by the profit for $|U| = 1$. If we compare our results to the results of Campbell and Savelsbergh (2006), we see similar outcomes. We notice that there are differences between percentage increases and averages, but this is acceptable as we are using different problem instances and it will be unlikely to obtain the same results.

We will now examine results of using fewer orders, thus 15 instead of 30 orders. The results are displayed in Table 2 in the same manner as the basic results in Table 1. The outcomes show very similar patterns as the basic results of BTSI, but with lower averages and percentage improvements. This is reasonable as we simply used the first 15 orders of the exact same problem instance.

Table 2: Profits BTSI methods with *NoAbandonment* and fewer orders

NoAbandonment	$\|U\|$	NoIncentives	FlatIncentives	MainIncentives	BestIncentives	BestCase
Pattern = 1	1	1052,21	1110,90	1130,09	1122,73	1092,12
	2	1052,21	1127,04	1124,74	1121,82	1092,12
	3	1052,21	1113,37	1114,18	1113,90	1092,12
	4	1052,21	1064,72	1087,04	1110,50	1092,12
	Average	1052,21	1104,01	1114,01	1117,24	1092,12
	% improvement	0,00	4,92	5,87	6,18	3,79
Pattern = 2	1	1030,41	1107,93	1105,43	1119,92	1092,12
	2	1030,41	1099,-	1124,31	1116,-	1092,12
	3	1030,41	1075,13	1095,18	1109,57	1092,12
	4	1030,41	1069,69	1080,08	1105,74	1092,12
	Average	1030,41	1087,94	1101,25	1112,81	1092,12
	% improvement	0,00	5,58	6,87	8,00	5,99
Pattern = 3	1	1026,17	1099,89	1101.59	1108.58	1092,12
	2	1026,17	1088,83	1120,32	1109,77	1092,12
	3	1026,17	1068,95	1092,83	1100,68	1092,12
	4	1026,17	1050,64	1096,76	1102,09	1092,12
	Average	1026,17	1077,08	1102,88	1105,28	1092,12
	% improvement	0,00	4,96	7,47	7,71	6,43

From now on, we will keep using 30 orders for every problem instance again. For the next results, we assumed customers only have 4 time slots with initial positive probabilities instead of 8. In Table 3 we give the results, again in the same structure as the previous two tables. We notice some interesting outcomes. The incentive methods give almost no improvements in profit. The *FlatIncentives* method even has an average decrease for *Pattern*=2. For this value of *Pattern*, we see that *BestCase* provided a decrease in profit too. In comparison with Campbell and Savelsbergh (2006), we have rather low improvements of incentive methods relative to *NoIncentives*. The average profits do however not seem significantly different and the differences in improvement can again be explained by the use of other problem instances.

Table 3: Profits BTSI methods with *NoAbandonment* and fewer acceptable time windows

NoAbandonment	$\|U\|$	NoIncentives	FlatIncentives	MainIncentives	BestIncentives	BestCase
Pattern = 1	1	1182,-	1173.98	1185.65	1207.15	1186,26
	2	1182,-	1224.96	1184.08	1190.80	1186,26
	3	1182,-	1199.14	1205.53	1187.13	1186,26
	4	1182,-	1182,-	1182,-	1187.13	1186,26
	Average	1182,-	1186,40	1189,32	1192,55	1186,26
	% improvement	0,00	0,37	0,62	0,89	0,36
Pattern = 2	1	1219,95	1198,20	1231.83	1231.66	1186,26
	2	1219,95	1176.41	1219.64	1229.42	1186,26
	3	1219,95	1226.66	1209.91	1221.81	1186,26
	4	1219,95	1219.95	1219.95	1221.81	1186,26
	Average	1219,95	1205,31	1220,33	1226,18	1186,26
	% improvement	0,00	-1,20	0,03	0,51	-2,76
Pattern = 3	1	1173,23	1194.36	1232.57	1223.81	1186,26
	2	1173,23	1190,30	1217.09	1228.29	1186,26
	3	1173,23	1185.83	1192.79	1219.82	1186,26
	4	1173,23	1173.23	1173.23	1219.82	1186,26
	Average	1173,23	1185,93	1203,92	1222,94	1186,26
	% improvement	0,00	1,08	2,62	4,24	1,11

In Table 4 we show profits of when a maximum budget of incentives of €2,- is used against the basis results obtained with a budget of €5,-. As we had highest average profits for $|U| = 2$ in the basic results, we only used result of this value of $|U|$ for the ease of comparison. With a lower budget, we obtain lower average profits. The incentives method still give good improvements relative to *NoIncentives*. If we compare results with Campbell and Savelsbergh (2006), we notice an interesting difference. In that research, the results of the *FlatIncentives* method is exactly the same for the different values of the budget. Our results differ a lot for *FlatIncentives* if the budget is changed from €5,- to €2,-. This could mean that we have a different implementation of the *FlatIncentives* method. Also, for $B = 5$, we notice that *FlatIncentives* has a higher average profit than the more sophisticated incentive methods. This does not occur in Campbell and Savelsbergh (2006), which might also be due to different implementations of *FlatIncentives*. Further results do resemble to Campbell and Savelsbergh (2006).

Table 4: Profits BTSI methods with *NoAbandonment* and varying budget

NoAbandonment	*Pattern*	NoIncentives	FlatIncentives	MainIncentives	BestIncentives	BestCase
$B = 2$	1	1226,76	1260,78	1414,90	1394,55	1305,14
	2	1215,09	1294,06	1311,29	1353,16	1305,14
	3	1207,95	1299,05	1327,22	1304,86	1305,14
	Average	1216,60	1284,63	1351,14	1350,86	1305,14
	% improvement	0,00	5,59	11,06	11,04	7,28
$B = 5$	1	1226,76	1409,54	1363,69	1383,35	1305,14
	2	1215,09	1368,32	1357,44	1353,15	1305,14
	3	1207,95	1349,98	1394,69	1362,73	1305,14
	Average	1216,6	1375,95	1371,86	1366,41	1305,14
	% improvement	0,00	13,10	12,76	12,31	7,28

Table 5 presents profits of when a different rate x is used. The table has the same structure as Table 4, we compare results of a rate of 0,1 with the basic rate of 0,2. It can be seen that the higher rate of effect for the incentives gives higher profits on average. This is also showed in Campbell and Savelsbergh (2006), although with higher differences between rates than we found in our results.

Table 5: Profits BTSI methods with *NoAbandonment* and varying rate

NoAbandonment	*Pattern*	NoIncentives	FlatIncentives	MainIncentives	BestIncentives	BestCase
$x = 0.1$	1	1226,76	1343,43	1371,70	1331,33	1305,14
	2	1215,09	1332,36	1358,67	1346,43	1305,14
	3	1207,95	1334,74	1329,36	1305,77	1305,14
	Average	1216,60	1336,84	1353,24	1327,83	1305,14
	% improvement	0,00	9,88	11,23	9,14	7,28
$x = 0.2$	1	1226,76	1409,54	1363,69	1383,35	1305,14
	2	1215,09	1368,32	1357,44	1353,15	1305,14
	3	1207,95	1349,98	1394,69	1362,73	1305,14
	Average	1216,6	1375,95	1371,86	1366,41	1305,14
	% improvement	0,00	13,10	12,76	12,31	7,28

The profits for methods with *Abandonment* from BTSI, with basic values for parameters, are given in Table 6. With the possibility of customers abandoning, we see that the differences in profits between incentives and no incentives grows. Whereas the differences in average profit between *NoIncentives* and the incentives methods were around 10% for *NoAbandonment*, the incentives methods for *Abandonment* increase profit by 14%-32%. Especially the *MainIncentives* method improves profit a lot when customers are able to walk away. This resembles results found in Campbell and Savelsbergh (2006). However, we do find large differences in results with *BestIncentives*. Our profits improve very little compared to Campbell and Savelsbergh (2006). We are not certain what causes this difference, but we suspect it is due to implementation. Lastly, we see that average profits are now highest when $|U| = 1$.

Table 6: Profits BTSI methods with *Abandonment*

Abandonment	$\lvert U \rvert$	NoIncentives	FlatIncentives	MainIncentives	BestIncentives	BestCase
Pattern = 1	1	1070,03	1337,17	1356,02	1240,06	1305,14
	2	1070,03	1354,77	1394,05	1234,86	1305,14
	3	1070,03	1281,5	1414,03	1237,74	1305,14
	4	1070,03	1203,8	1451,53	1249,21	1305,14
	Average	1070,03	1294,31	1403,91	1240,47	1305,14
	% improvement	0,00	20,96	31,20	15,93	21,97
Pattern = 2	1	1091,63	1377,99	1373,23	1249,21	1305,14
	2	1091,63	1289,17	1366,74	1248,85	1305,14
	3	1091,63	1267,23	1394,48	1252,54	1305,14
	4	1091,63	1186,57	1370,88	1256,89	1305,14
	Average	1091,63	1280,24	1376,33	1251,87	1305,14
	% improvement	0,00	17,28	26,08	14,68	19,56
Pattern = 3	1	1113,84	1354,26	1351,07	1261,92	1305,14
	2	1113,84	1268,71	1391,5	1277,83	1305,14
	3	1113,84	1274,15	1342,89	1280,36	1305,14
	4	1113,84	1188,64	1356,28	1271,14	1305,14
	Average	1113,84	1271,44	1360,44	1272,81	1305,14
	% improvement	0,00	14,15	22,14	14,27	17,17

5.2.2 Wider Time Slot Incentives

We now show the results of the WTSI model of Section 4.2. In this model, we only use the basic problem instances. Again, we look first at the *NoAbandonment* scenario. Table 7 includes profits of the four methods used in this model for all different values of *Pattern* and $|U|$. As in BTSI, we see that incentives improve profits a lot. These improvements are also showed by Campbell and Savelsbergh (2006), although with lower improvements of *FlatIncentives* than we obtained. This might also be a reason to suspect differences between the *FlatIncentives* methods. Also, the *FlatIncentives* method gives a higher average profit than *MainIncentives* does. The average profits are again highest for $|U| = 1$.

Table 7: Profits WTSI methods with *NoAbandonment*

NoAbandonment	$\lvert U \rvert$	NoIncentives	FlatIncentives	MainIncentives	BestCase
Pattern = 1	1	1226,76	1447,23	1364,27	1418,44
	2	1226,76	1391,37	1421,54	1418,44
	3	1226,76	1368,63	1433,24	1418,44
	4	1226,76	1366,7	1440,36	1418,44
	Average	1226,76	1393,48	1414,85	1418,44
	% improvement	0,00	13,59	15,33	15,62
Pattern = 2	1	1215,09	1440,08	1337,43	1418,44
	2	1215,09	1373,35	1345,74	1418,44
	3	1215,09	1361,7	1364,32	1418,44
	4	1215,09	1309,59	1364,8	1418,44
	Average	1215,09	1371,18	1353,07	1418,44
	% improvement	0,00	12,85	11,36	16,74
Pattern = 3	1	1207,95	1374,75	1345,51	1418,44
	2	1207,95	1387,98	1346,72	1418,44
	3	1207,95	1378,73	1330,46	1418,44
	4	1207,95	1361,37	1351,67	1418,44
	Average	1207,95	1375,71	1343,59	1418,44
	% improvement	0,00	13,89	11,23	17,43

We include the possibility of abandonment again and present the results in Table 8 in the same manner the *NoAbandonment* results were presented. The incentives methods give higher improvements in profits with *Abandonment* than with *NoAbandonment*. Now, the *MainIncentives* increases average profits more than *FlatIncentives* does. This is also seen in Campbell and Savelsbergh (2006) but with slightly higher improvements. The highest average profits can be found for $|U| = 2$, thus again for a small number of offered time slots.

Table 8: Profits WTSI methods with *Abandonment*

Abandonment	$\|U\|$	NoIncentives	FlatIncentives	MainIncentives	BestCase
Pattern = 1	1	1070,03	1280,79	1332,13	1339,96
	2	1070,03	1348,16	1368	1339,96
	3	1070,03	1322,33	1327,21	1339,96
	4	1070,03	1270,13	1418,65	1339,96
	Average	1070,03	1305,35	1361,60	1339,96
	% improvement	0,00	21,99	27,25	25,23
Pattern = 2	1	1091,63	1333,34	1301,36	1339,96
	2	1091,63	1277,18	1316,89	1339,96
	3	1091,63	1303,95	1347,08	1339,96
	4	1091,63	1211,4	1337,11	1339,96
	Average	1091,63	1281,47	1325,61	1339,96
	% improvement	0,00	17,39	21,43	22,75
Pattern = 3	1	1113,84	1295,33	1234,21	1339,96
	2	1113,84	1278,46	1322,94	1339,96
	3	1113,84	1219,41	1330,83	1339,96
	4	1113,84	1244,87	1321,71	1339,96
	Average	1113,84	1259,52	1302,42	1339,96
	% improvement	0,00	13,08	16,93	20,30

5.2.3 Calculation times

In Table 9 we present the calculation times of our methods. These times are the averages over the calculation times of different values of $|U|$ and *Pattern* in seconds. Also, averages over the two scenarios and percentage improvements relative to *NoIncentives* are calculated. The entries with $-$ are for the *BestIncentives* method that is used in BTSI but not in WTSI. We notice that this method gives an enormous increase in calculation time relative to *NoIncentives*. *FlatIncentives* method increase by around 20% and the *MainIncentives* method doubles in calculation time. There are almost no differences in calculation times between BTSI and WTSI.

Table 9: Calculation times

	scenario	NoIncentives	FlatIncentives	MainIncentives	BestIncentives	BestCase
BTSI	*NoAbandonment*	0,99	1,15	2,01	25,39	0,99
	Abandonment	0,89	1,15	2,24	22,18	1,06
	Average	0,94	1,15	2,12	23,78	1,02
	% improvement	0,00	22,34	125,53	2429,79	8,51
WTSI	*NoAbandonment*	0,99	1,21	2,02	-	1,07
	Abandonment	0,89	1,00	1,86	-	0,94
	Average	0,94	1,11	1,94	-	1,00
	% improvement	0,00	18,09	106,38	-	6,38

6 Conclusion

We see that in the setting of Campbell and Savelsbergh (2005) offering incentives, even simple flat incentives, can increase profits through reduction of delivery costs. In our experiment, calculating incentives via the *MainIncentives* method improved profits the most on average and is much faster than the *BestIncentives* method. The *FlatIncentives* methods makes improvements similar to *MainIncentives* in our base problem instances, this is not seen in Campbell and Savelsbergh (2005) which could suggest that we used a different implementation of this method.

When a single time window is more likely to be chosen by a customer, which makes it harder to influence this customer to select a certain slot, the *MainIncentives* method performs better than *FlatIncentives*. Also if we include abandonment, the *MainIncentives* method highly increases profit. Offering incentives via this method prevents customers from abandoning. The results from offering incentives for wider time slots are not significantly better or worse than for one-hour time windows and the two models have similar calculation times.

Our goal was to find a method to improve profits by offering incentives, the *MainIncentives* is a good method which does not require much calculation time. Offering incentives to only one or two time slots improved profits the most. As giving multiple incentives might be confusing for customers, offering only one incentive seems a good choice. Our conclusions correspond to many of the insights from Campbell and Savelsbergh (2005).

7 Discussion

For our models, we have made assumptions and used assumptions from Campbell and Savelsbergh (2006) that might not be very realistic. First, we assume that customers can always be influenced to select a certain time window. Some customers might only be able to receive their delivery between 19:00 and 20:00 as they are working the rest of the day. We could incorporate this by not only change the pattern, but also the length of the pattern. Customers could have a uniformly distributed length of a pattern from 1 to 8. Also, customers are usually more often home in mornings or evenings due to work. Offering an incentive for a single time window might than be less effective as it is not always possible to leave work during the day.

Currently, the rates in which probabilities increase by incentives are constant. However, if the incentives are extremely low a customer might not be interested as it does almost nothing to the price. When incentives increase and the time slot is viable for a customer, at a certain hight of the incentive the customer might want to select this slot. Even increasing the incentive more would be unnecessary, thus a quadratic rate might be more profitable. However, our incentives can then no longer be calculated with a linear program.

We also propose an extension to the WTSI model of Campbell and Savelsbergh (2006). Here, it is possible to select two consecutive one-hour time slots. It might be reasonable to assume that most customers can receive delivery only in the morning and evening due to work. These customers might be willing to select two non-consecutive time slots. To incorporate this into a model, some adjustments to WTSI have to be made. The most important change is needed in the routing part. Again, the earliest and latest time a delivery can occur has to be calculated. Then, either the earliest time has to fall within the time window that occurs earliest on the day or the latest time must be in the latest time slot. For the incentives calculations, the same methods of WTSI can be used for two non-consecutive time windows.

Finally, results and conclusions are based on simulations with specific values for the parameters. Using different values, more or less customers, more trucks or adding capacity constraints might give other insights. To evaluate these effects, more research is required.

References

[1] Campbell, A.M., Savelsbergh, M. *Decision Support for Consumer Direct Grocery Initiatives.* Transportation Science, 2005.

[2] Campbell, A.M., Savelsbergh, M. *Incentive Schemes for Attended Home Delivery Services.* Institute for Operations Research and the Management Science, Maryland, USA, 2006.

[3] Agatz, N., Campbell, A.M., Fleischmann, M., Savelsbergh, M. *Time Slot Management in Attended Home Delivery.* ERIM Report Series Research In Management, 2008.

[4] Ehmke, J. F., Mattfeld, D. C. *Vehicle Routing for Attended Home Delivery in City Logistics.* Procedia-Social and Behavioral Sciences, Elsevier, 2012.

[5] Ehmke, J. F. *Integration of Information and Optimization Models for Routing in City Logistics.* [*Attended Home Delivery*] Springer, 2012