Revenue Management - Theory and Practice

Anne Høj Kjeldsen - s991476 Pernille Meyer - s991494

Supervisor at IMM: Professor Jens Clausen

Supervisor at British Airways: Hans-Martin Gutmann

17th February 2005

Technical University of Denmark

Preface

This report is the Master Thesis for a Master of Engineering at The Technical University of Denmark. This Master Thesis is carried out in the period 1/2-2004 to 1/2-2005 in co-operation with British Airways and the Institute of Informatics and Mathematical Modelling at DTU. The supervisor at British Airways is Hans-Martin Gutmann and the supervisor at IMM, DTU, is Professor Jens Clausen.

We are thankful for the opportunity of carrying out our Master Thesis in co-operation with British Airways and for the supervision received from Hans-Martin Gutmann.

Finally, we appreciate the financial support which we received from the Oticon Foundation for the accomplishment of this Master Thesis. The support has given us some opportunities throughout the project period which we would not have had without it.

Anne Høj Kjeldsen - s991476

Pernille Meyer - s991494

Abstract

The main purpose of this work is to develop an efficient dynamic programming (DP) algorithm for the revenue optimization problem in the presence of trade-up behaviour. Trade-up is when a passenger buys a more expensive ticket than originally intended, if the desired ticket is not available. Efficiency of an algorithm is both measured with respect to revenue and running time. To achieve this objective the Seat Inventory Control (SIC) problem *without* trade-up is described first to give a fundamental understanding of the basic problem. The basic SIC problem is concerned with the allocation of discount and full-fare seats on a flight so as to maximize total revenue.

Next, two different cases of the SIC problem with trade-up are investigated, one with general assumptions and one with more specific assumptions made by British Airways. A dynamic programming model is set up for each of these problems and different solution methods, both exact and approximate methods, are introduced for solving the DP model. Finally the methods are tested by simulating arrival processes and results are obtained by a comparison of the methods applied on these arrival processes.

Numerical results suggest that in a market where trade-up occurs, a large gain in revenue can be obtained by using methods incorporating trade-up instead of methods without trade-up.

Keywords: *revenue management, seat inventory control, dynamic programming, trade-up, approximation method.*

Contents

List of Figures

List of Tables

List of Symbols

The symbols which are used often in the report are listed in the table below.

Symbol	Description
t	Decision period, where smaller values of t represent later points in time.
x	Number of remaining seats, i.e., remaining capacity.
k	Number of different fare classes.
C	Capacity of the aircraft.
T	Total number of decision periods.
F_i	Value of accepting a request for a seat in fare class i.
D_i^t	Mean value of expected demand to come from decision period t to departure for fare class i.
S_i^j	Number of seats protected for class i from class j.
π_i	Number of seats protected for class i from classes $1, \ldots, i-1$.
β_i	The probability that a request is for fare class i.
p_i	The probability that the remaining capacity after selling an additional seat in class $i+1$ will not fail to meet subsequent class i demand.
b_i^t	Number of seats sold for class i from decision period T to t.
BL_i	Booking limit for fare class i.
P_i^t	Probability of a request for class i in decision period t.
P_0^t	Probability of no request in decision period t.
$q_{i,j}$	Probability of trade-up from fare class i to fare class j.
$\Phi_i(Z_i)$	Probability that demand for fare class i is less than or equal to the seat allocation Z_i.
$\hat{x}_i(t)$	Critical booking capacity for decision period t in fare class i.
$\hat{t}_i(x)$	Critical decision period for a remaining capacity x in fare class i.
$V_t(x)$	Total expected revenue that can be generated from decision periods $t, t-1, \ldots, 1$ given a remaining capacity x.
$\Delta V_t(x)$	Expected marginal value of capacity, when x seats remain in decision period t, i.e., $\Delta V_t(x) = V_t(x) - V_t(x-1)$.
$U_t^i(x)$	Expected revenue which can be genereated with t decision periods and x seats remaining when a request for class i is rejected.
$\hat{V}_t(x)$	Upper bound for the value function $V_t(x)$.
$\check{V}_t(x)$	Lower bound for the value function $V_t(x)$.
EMSR	Expected marginal seat revenue.

Chapter 1

Introduction

In this chapter the basic ideas in Revenue Management will be explained and the Seat Inventory Control problem, denoted the SIC problem, will be described. Furthermore a short literature review on the SIC problem is given. Finally, the contributions of this work are listed and an outline of the report is presented.

1.1 Problem Description

In the past years many airlines have been forced to develop an efficient and structured way of pricing and selling the seats on their flights so as to maximize total revenue. This is due to the entry of many low-cost airlines in the market, which has resulted in hard competition. Existing airlines needed a new strategy to be able to compete with the low fares from the new airlines. This strategy is based on dividing the aircraft into a number of classes with different fares, where different conditions apply to each class. This way the airlines are able to offer both discounted and full-fare tickets and therefore they can compete with low-cost airlines. With as many as 20 fare classes on a flight, a big task for the airline is to manage how many seats to sell in each fare class. The task of pricing and allocating capacity to different fare classes is known as Revenue Management. Seat Inventory Control is the part of Revenue Management, which is concerned only with the allocation of discount and full-fare seats on a flight so as to maximize total expected revenue.

In this report only single-leg flights are considered, i.e., only flights from one city to another with no stops in-between. For single-leg flights the SIC problem can be described as follows. Consider an aircraft with capacity C. Passengers can request one of k fare classes, where class 1 corresponds to the most expensive fare and class k to the least expensive. Requests for the different fare classes arrive throughout the booking period, which is divided into smaller time periods called decision periods. Based on the number of seats already booked and the decision period in which a request arrives, the task is to decide whether to accept or reject the request. The decision is made such that the total expected revenue is maximized.

The problem just described is the basic SIC problem, which can be extended in several ways, for instance by including one or more of the items below

- *No-shows*: A certain percentage of the passengers will not show up at departure.
- *Cancellation*: A certain percentage of the passengers will cancel their reservation before departure.
- *Overbooking*: Accepting more passengers than the capacity of the aircraft in the expectation of no-shows and cancellations.
- *Go-shows*: A number of passengers shows up at departure without a ticket wanting to buy one.
- *Trade-up*: A rejected passenger will request a more expensive ticket on the same flight.
- *Recapture*: A rejected passenger will buy a ticket on another flight from the same airline.
- *Network*: Modelling transfer traffic instead of single-leg flights.
- *Multiple bookings*: Modelling multiple bookings instead of bookings of a single seat, for instance the booking of an entire family instead of just a single person.

In this report the basic SIC problem is extended to incorporate trade-up. As mentioned above, trade-up is when a person requesting fare class i chooses to buy one of the classes $i-1,\dots,1$ with a certain probability, if his or her request for a class i ticket is rejected. In the basic SIC problem without trade-up the probabilities for trade-up equal zero, i.e., a rejected request is lost revenue for the airline. This is not necessarily the case when trade-up is incorporated.

The problem in Seat Inventory Control is basically to determine how many seats to sell at discounted fares. If too many seats are sold at low fares, the airline may have to reject full-fare passengers. If, on the other hand, too few seats are offered at discounted fares, the airline may not sell all the seats on the aircraft. The SIC problem is complicated by the fact that passengers requesting discounted-fare classes often book before passengers requesting full-fare classes. A simple solution to the problem is to split the capacity of the aircraft into blocks of seats to be sold to individual classes exclusively. This solution method could result in having to reject a request for a high-fare class even if low-fare classes are still open for sale, though. Therefore the concept *nested availability* is introduced. Nested availability means that classes can be ordered by their fare, such that a class with a high fare can take seats at the expense of classes with a lower fare. Thus, unexpected demand for a class can be satisfied as long as lower-fare classes are still open for sale. If nested availability is used it is necessary to calculate a *booking limit* for each class. A booking limit for a certain class indicates the maximum number of seats the airline is willing to sell in this class and all lower-fare classes. Closely related to the booking limits are *protection levels*. The relationship is that the booking limit is equal to the remaining capacity minus the protection levels for all classes with a higher fare. The booking systems which are used by most airlines for accepting or rejecting requests are based on booking limits.

Another factor which complicates the SIC problem is the uncertainty in the demand forecasts. Demand is often affected by factors external to the airline, which implies that forecasts based on booking data from previous flights may be very inaccurate. Hence, forecasts usually need to be revised during the booking period as new information about demand becomes available. Therefore, the booking limits need to be recalculated when the forecasts have been updated.

Thus, the calculation of the booking limits has to be computationally efficient to be usable. An airline like British Airways (BA) operates around 1000 flights a day. As selling starts one year before departure, at any one time there are around 365000 flights in the system. Booking limits do not have to be updated every day, for instance, there may not be much booking activity several months before departure. As a rough estimate 100000 flights go through an optimization every day. This means that one optimization must not take longer than 0.85 seconds.

1.2 Purpose of this Work

The main purpose defined by BA is to develop an efficient dynamic programming algorithm for the SIC problem with trade-up. Efficiency is measured both with respect to expected revenue and running time.

This objective is achieved by carrying out several tasks. The first task is an extensive review of the literature about modelling trade-up behaviour and dynamic programming formulations of the SIC problem. Furthermore, DP models are formulated, both for the problem without and with trade-up. For solving these, different algorithms are examined, especially algorithms using approximations. Finally, far-reaching numerical experiments are accomplished to compare the different algorithms.

1.3 Literature Review

In this section a short introduction to the existing literature about the SIC problem will be given.

The first literature about the problem was published in the early seventies, when the first airlines began offering discounted fare products as well as the regular high-fare tickets. The paper [10] by Botimer goes into more detail about the reasons for using a differentiated fare product structure. As described in Section 1.1, this development had the potential for major airlines to compete with discount airlines and thereby increase revenues. However, it also presented them with the challenge of determining how many seats to offer in each fare class. Hence, after the introduction of differentiated fare classes a large amount of literature has been written about the problem and possible solution methods.

As mentioned above, the history of the revenue management problem for the airline industry started in the early seventies. This history is introduced further by McGill and van Ryzin in [21], where an overview of existing literature about the SIC problem is given. Furthermore, forecasting and different extensions to the basic problem such as overbooking, cancellations, no-shows and go-shows are discussed. The paper [22] by Pak and Piersma also gives an overview of the solution methods for the problem presented throughout the literature.

Usually the input data for the problem are demand forecasts and fare values for each class. In [26], Weatherford and Belobaba investigates the impacts of errors in these input data. This is done for a problem with multiple fare classes.

Littlewood, [20], was the first to propose that discount-fare requests should be accepted as long as their revenue value exceeded the expected revenue from future full-fare bookings. This model is for two fare classes and was later extended by Belobaba, [3] and [4], to multiple fare classes. Belobaba called this heuristic EMSR, which is an abbreviation of *Expected Marginal*

Seat Revenue. It is now known as the EMSRa method. In [6] Belobaba proposes a different heuristic, which is similar to the EMSR decision rule to yield even higher revenues, the EMSRb heuristic.

Throughout the booking period requests are accepted or rejected. Thus the number of accepted bookings and possibly the demand forecasts change throughout the period. Hence the problem to be solved is dynamic. In [4] Belobaba describes how the EMSRa heuristic can be used on a dynamic problem even though the EMSRa decision rule is a static rule. Another approach is to set up a dynamic programming model. This is done in [19] by Lee and Hersh for a single-leg flight without trade-up. This model is used the most in the literature, and thus many approximation algorithms and extensions have been made for this model. One of the approximation algorithms is for the network optimization problem and is suggested in [8] by Bertsimas and Popescu. Here the value function in the dynamic programming model is approximated by a deterministic linear program, thus making the model easier to solve. In [18], Lautenbacher and Stidham consider another approximation algorithm for solving the SIC Problem. A framework is set up, which combines both the dynamic programming model proposed by Lee and Hersh in [19] as well as some static models including the EMSR heuristic proposed by Belobaba.

Subramanian et al., [24], extends the dynamic programming model for a single-leg flight and multiple fare classes to include no-shows, cancellations and overbooking. In the article it is shown that the problem is equivalent to a problem in optimal control of admission to a queuing system.

In [14] by Cooper and Homem de Mello the dynamic programming problem is described, and it is proposed to solve this problem using a hybrid method. The idea is to use a heuristic early on in the booking period, where accuracy is not too important and then later on in the booking period switch to an accurate decision rule. The problem is solved for a two-leg flight.

Another approximation algorithm is suggested by Chen, Gunther and Johnson in [11]. Here an entire flight network is considered and again the dynamic programming model by Lee and Hersh is set up. The algorithm finds upper and lower bounds for the value function in the model. A stochastic linear program (LP) is formulated and used as a lower bound and a deterministic LP is used as an upper bound. The acceptance rule is based on these bounds and instead of having to calculate the values of the bounds for all combinations of remaining capacity and time, the bounds are calculated for specific values of remaining capacity and time. Splines are then used to interpolate between these values of remaining capacity and linear interpolation is used between these values of time. This yields an approximation of the bounds for all combinations of remaining capacity and time.

An important extension to the basic SIC Problem is to incorporate trade-up. One of the first papers in which trade-up is incorporated in a model was written in 1993 by Andersson, Algers and Kohler, [1]. Here, a deterministic linear programming model including trade-up is set up for a flight network. In 1998 Andersson wrote a new article about a model for a network where trade-up is included, see [2]. In this article trade-up is modelled specifically as a passenger utility maximization model. Additionally, it is discussed in which markets it may be profitable to use a model including trade-up and an allocation model for a single-leg flight including trade-up is set up. This model is an expansion of the EMSRa model with trade-up. Finally, a deterministic model for a network with trade-up is set up and described. This model is equivalent to the model set up in [1].

Bodily and Weatherford extends the basic SIC Problem to handle situations with continuous, non-discrete resources and overbooking, see [9]. Furthermore, a decision rule for the problem with trade-up is incorporated for more than two fare classes.

In [6] Belobaba and Weatherford describes both the EMSRb heuristic and the decision rule proposed by Bodily and Weatherford in [9] for the SIC Problem with trade-up. These two approaches are combined to develop a heuristic, which is better than both EMSRb and the decision rule for the problem with trade-up.

In [29] a dynamic programming problem with two classes which incorporates trade-up is considered by Zhao and Zheng. An additional assumption which is applied is that once a discount class has been closed for sales, this class cannot be reopened. The latter assumption is important, since airlines are interested in making the passengers realize, that the earlier they book, the larger is the probability that they are able to get a discount-fare ticket.

A dynamic programming model for multiple fare classes is set up by You in [28]. This model extends the model by Lee and Hersh in [19] to incorporate trade-up. The decision making has two stages. In the first stage, it is decided whether to accept or reject the request. This decision is analogous to the decision in [19]. The second decision is, after rejecting a request, which classes should be offered to the rejected passenger.

In [25] Talluri and van Ryzin also consider the SIC problem for a single-leg flight including trade-up. In this paper buyers' choice behaviour is modelled explicitly and a method for choosing which classes should be open at each point in time is developed.

Authors	Paper	Multiple Classes	Network	DP-Form.	Heuristic	Trade-up
Algers, Andersson and Kohler	[1]	X	X	-	X	X
Andersson	[2]	X	X	-	X	X
Belobaba	[3]	X	-	-	X	-
Belobaba	[4]	X	-	-	X	-
Belobaba and Weatherford	[6]	X	-	-	X	X
Bertsimas and Popescu	[8]	X	X	X	X	-
Bodily and Weatherford	[9]	X	-	-	X	X
Chen, Gunther and Johnson	[11]	X	X	X	X	-
Cooper and Homem de Mello	[14]	X	X	X	X	-
Lautenbacher and Stidham	[18]	X	-	X	X	-
Lee and Hersh	[19]	X	-	X	-	-
Littlewood	[20]	-	-	-	X	-
Subramanian et al.	[24]	X	-	X	-	-
Talluri and van Ryzin	[25]	X	-	X	-	X
You	[28]	X	-	X	-	X
Zhao and Zheng	[29]	-	-	X	X	X
Kjeldsen and Meyer	This work	X	-	X	X	X

Table 1.1: Literature Overview.

1.4 Contributions of this Work

As seen from the literature overview in Table 1.1 and the extensive bibliography in this report, many papers describe the basic SIC problem. Contrary to this only few articles treat the SIC problem with trade-up. Furthermore, it is especially difficult to find papers, which compare more than two different solution methods or articles which deal with approximation methods for solving a dynamic programming model for the SIC problem with trade-up. So this report adds a new angle to the literature about revenue management, since multiple classes, trade-up, DP formulations and heuristics is handled.

Several solution methods for both the SIC problem without and with trade-up, especially approximation methods, are investigated. Furthermore, existing solution methods for the problem without trade-up are adjusted to fit the problem with trade-up.

An extensive number of numerical experiments are made to determine the best methods for each of the problems without and with trade-up. Moreover, a comparison of the methods without and with trade-up is accomplished. These are compared in a trade-up market to see the differences in the revenues obtained by using methods with trade-up instead of methods without trade-up.

1.5 Structure of the Report

The structure of this report is as follows. In Chapter 2 an introduction to dynamic programming will be given and it will be explained how dynamic programming can be applied to the SIC problem. In this report the SIC problem is treated both with and without trade-up. The reason that the problem without trade-up is included is to give the reader a fundamental understanding of the SIC problem before trade-up is incorporated. Hence, in Chapter 3 different solution methods for the SIC problem without trade-up will be described. The chapter also includes a description of the implementation of the solution methods for the problem without trade-up. For the problem with trade-up the solution methods are set up for two cases, where different assumptions apply. In Chapter 4 a discussion of when trade-up should be included in the solution methods is given and furthermore it is explained which conditions and assumtions apply when a trade-up market is under consideration. The assumptions for the two different trade-up cases are explained and solution methods for both cases are set up. Furthermore the implementation of these methods is described. Finally, in Chapter 5 it is described how the parameters of various methods are tuned, and numerical results are presented, both for the SIC problem with and without trade-up. Also, methods without trade-up are compared to methods with trade-up when applied to a trade-up world to show the benefit of modelling trade-up behaviour. The report closes with conclusions in Chapter 6.

Chapter 2

Dynamic Programming

In Section 1.3 a number of different papers describing the SIC problem were introduced. Several authors suggest solving the problem using dynamic programming (DP), since a DP model is the most accurate model for the SIC problem both with and without trade-up. Therefore, in this chapter an introduction to dynamic programming is given. Furthermore, it will be explained how DP can be applied to the SIC Problem.

2.1 Basic DP Theory

Like other branches of mathematical programming, dynamic programming is a general approach to solving certain problems, for instance, problems where the input to the model varies with time. Hence, some general characteristics apply for all DP problems, but particular equations must be made to fit each specific problem. Dynamic programming is a way of decomposing the problem under consideration into smaller subproblems, which are easier to solve. DP can for instance be used to solve a problem where decisions must be made at different points in time, i.e., in different stages, and where the input to the problem also changes as time progresses. The problem to be solved can either be a maximization or minimization problem. In the following the general problem is assumed to be a maximization problem.

Problems which can be solved using dynamic programming may be very different, but there are a number of characteristics which are common for all DP problems. These are the following

- The problem can be divided into *stages* and a decision has to be made in each stage.
- Each stage has a number of *states* associated with it.

- The decision in one stage transforms one state into a state in the next stage.
- Given the current state, the optimal decision for each of the remaining states do not depend on the previous states or decisions.
- A recursive relationship exists, which identifies the optimal decision for stage t, given that stage $t + 1$ has already been solved.
- The final stage must be solvable by itself.

One of the challenges when defining a DP problem is to determine stages and states such that all of the above characteristics are satisfied.

An additional property of the problems, which can be solved using DP, is that a decision made at a given point in time cannot be viewed in isolation, future decisions must be taken into account as well. Hence, if the objective function is to be maximized, it may not be sufficient to maximize this in each stage, since the decision made in the present stage affects which decisions can be made in future stages. The best decision in the present stage might imply an inevitably low objective function value in future stages. In DP, this problem is overcome by making a decision in each stage, which maximizes the sum of the objective function value gained in the present stage and the expected objective function value gained in future stages, assuming optimal decision making in these stages. Due to a random parameter the outcome of making a decision in a stage is only predictable to some extent. Therefore it is the *expected* objective function value gained in future stages which is maximized. In the following, it is assumed that stages are points in time, i.e. a discrete time dynamic system is considered.

A model for determining optimal decisions in a dynamic system over a finite number of stages has two main features

1. There is an underlying discrete time dynamic system.
2. The objective function, R, is additive over time.

The underlying discrete time dynamic system has the form

$$x_{t+1} = f_t(x_t, u_t, w_t), \qquad t = 0, 1, \ldots, N-1,$$

where

t Indexes discrete time, the stages of the system.

x_t State of the system at time t.

u_t Decision or decision variable to be determined at time t.

w_t Random parameter at time t, also called disturbance or noise.

N Time horizon or the number of times control is applied.

The objective function, R, is additive over time, thus it can be expressed as

$$R = g_N(x_N) + \sum_{t=0}^{N-1} g_t(x_t, u_t, w_t),$$

where $g_t(x_t, u_t, w_t)$ is the objective function value obtained at time t and the objective function value $g_N(x_N)$ is incurred at the termination of the process.

As mentioned previously, the presence of the random parameter, w_t, makes it impossible to optimize the total objective function value. Instead, the total *expected* objective function value is optimized

$$R_E = E\left\{ g_N(x_N) + \sum_{t=0}^{N-1} g_t(x_t, u_t, w_t) \right\}.$$

The decision variables in the system are $u_0, u_1, \ldots, u_{N-1}$, i.e., the system is optimized with respect to these.

Let X_t be the set of possible states in stage t, let H_t be the set of possible decisions in stage t, and let W_t be the set of possible outcomes of the random parameter w_t at time t, then $x_t \in X_t$, $u_t \in H_t$ and $w_t \in W_t$. The decisions u_t must take on values in a nonempty subset, which is dependent on x_t, $Q_t(x_t) \subset H_t$. The value of w_t belongs to a probability distribution $P(\cdot|x_t, u_t)$, which may depend on x_t and u_t, but not on previous disturbances, $w_{t-1}, \ldots, w_0$.

In each stage, t, a decision law, μ_t, for mapping the present state into an appropriate decision is to be determined, i.e., $u_t = \mu_t(x_t)$. The sequence of these functions for all time periods is denoted θ:

$$\theta = \{\mu_0, \ldots, \mu_{N-1}\}.$$

The mapping must satisfy that $\mu_t(x_t) \in Q_t(x_t)$ for all $x_t \in X_t$ and it is then called admissible. Θ is the set of all admissible policies, i.e., $\theta \in \Theta$.

Given an initial state x_0 and an admissible policy, θ, the system equation

$$x_{t+1} = f_t(x_t, \mu_t(x_t), w_t), \qquad t = 0, 1, \ldots, N-1,$$

makes the state, x_t, and the disturbance, w_t, random variables with well-defined distributions. Hence, for given functions, g_t, $t = 0, 1, \ldots, N$, the expected objective function value given by

$$R_\theta(x_0) = \underset{\substack{w_t \\ t=0,1,\ldots,N-1}}{E} \left\{ g_N(x_N) + \sum_{t=0}^{N-1} g_t(x_t, \mu_t(x_t), w_t) \right\}$$

is a well-defined quantity. Thus, for a given initial state, an optimal decision policy θ is one, that optimizes the total expected objective function value. The optimal decision problem is then given by

$$R_{\theta^*}(x_0) = \max_{\theta \in \Theta} R_\theta(x_0).$$

2.1.1 The Dynamic Programming Algorithm

The dynamic programming algorithm is an algorithm for finding optimal solutions to DP models. It is based on the *principle of optimality*, which has the following definition

> **Principle of Optimality**
>
> Let $\theta^* = \{\mu_0^*, \mu_1^*, \ldots, \mu_{N-1}^*\}$ be an optimal decision policy for the DP problem and assume, that when using θ^*, a given state x_i occurs at time i with positive probability. Consider the subproblem which is in state x_i at time i. Then the aim is to maximize the objective function value "to-come" from time i to time N:
>
> $$R_{to-come} = E \left\{ g_N(x_N) + \sum_{t=i}^{N-1} g_t(x_t, \mu_t(x_t), w_t) \right\}.$$
>
> Then the truncated control policy $\mu_i^*, \mu_{i+1}^*, \ldots, \mu_{N-1}^*$ is optimal for this subproblem.

The intuitive interpretation of the principle of optimality is that if the present state is x_i, the optimal control policy from time i to time $N-1$ is $\mu_i^*, \mu_{i+1}^*, \ldots, \mu_{N-1}^*$, but this is overall the optimal policy. Thus, given the current state, an optimal policy for the remaining problem is independent of the policy decisions made in the first part of the problem, it only depends on the current state.

The implication of the principle of optimality is that a systematic procedure for solving DP problems can be used. An optimal decision policy for a dynamic problem can be found by first determining the optimal decisions for the last stage for all possibilities of states in that stage. Then the subproblem is extended to be the last two stages and the optimal controls for all possibilities of states in the second-to-last stage are found using the knowledge about the optimal controls for the last stage. The problem is enlarged and subproblems are solved until the problem is solved in its entirety. Each of the subproblems is much simpler than the entire problem.

Hence, using the principle of optimality, the dynamic programming algorithm is as follows

The Dynamic Programming Algorithm

For every initial state x_0, the optimal objective function value $R^*(x_0)$ of the dynamic programming problem is equal to $R_0(x_0)$, where the function R_0 is given by the final step of the following algorithm, which proceeds backward in time from period $N-1$ to period 0. If the last state, x_N, is known the algorithm proceeds backward in time with respect to this state, otherwise the algorithm proceeds backwards in time from all possible final states, x_N. For $t = N-1, N-2, \ldots, 0$, the dynamic programming algorithm is

$$R_N(x_N) = g_N(x_N)$$
$$R_t(x_t) = \max_{\substack{u_t \in Q_t(x_t) \\ w_t}} E\left\{g_t(x_t, u_t, w_t) + R_{t+1}\left(f_t(x_t, u_t, w_t)\right)\right\}, \quad (2.1)$$

where the expectation, E, is with respect to the probability function of w_t and x_t. Furthermore, if $u_t^* = \mu_t^*(x_t)$ maximizes the right hand side of (2.1) for each x_t and t, then the policy $\theta^* = \{\mu_0^*, \mu_1^*, \ldots, \mu_{N-1}^*\}$ is optimal.

For a proof of the above proposition see [7].

In some cases, the dynamic programming problem can be simplified such that the following statements are satisfied

- For each stage t the system has a finite number of states, X_t.
- A reward $g_t(x_t, u_t, w_t)$ is obtained after the decision u_t has been applied in state x_t.
- The probability that the system will be in state j at stage t is denoted $p_j(t)$. Since the current state can be one of n different states, from Baye's Theorem, see [12], it follows that $p_{j+1}(t)$ is linearly dependent on the current state probabilities and

$$p_j(t+1) = p_{1j}(u,t)p_1(t) + p_{2j}(u,t)p_2(t) + \cdots + p_{nj}(u,t)p_n(t),$$

where

$$
\begin{aligned}
p_{1j}(u,t) &= P\{x_{t+1} = j | x_t = 1, u_t = u\} \\
p_{2j}(u,t) &= P\{x_{t+1} = j | x_t = 2, u_t = u\} \\
&\vdots \\
p_{nj}(u,t) &= P\{x_{t+1} = j | x_t = n, u_t = u\}.
\end{aligned}
$$

The probabilities $p_{ij}(u,t)$ are the transition probabilities, i.e., the probabilities that the state of the system is j in the next stage, given that the system is in state i at the present stage.

When a system satisfies the above, it is said to be Markovian, and a model which characterizes a Markovian system is called a Markov chain. It is possible to totally characterize the chain's state for each stage, since Markov chains are finite. This is one of the advantages of dealing with a Markovian system. For further elaboration on Markov processes, see [23].

2.2 A DP Model for the SIC Problem

An important question is whether dynamic programming provides a good model for the SIC problem. Recall that the SIC problem is characterized by the following. Passengers request tickets at different points in time throughout the booking period. Each time a passenger places a request, a decision has to be made of whether to accept or reject the request. This decision, though, cannot be viewed in isolation, since even if accepting the request yields a positive revenue it may not be optimal for the entire problem. Expected future requests must be taken into account as well. Hence, to maximize total revenue, it is not sufficient to maximize revenue from each request, since the decision made for the present passenger affects which decisions that can be made for future requests.

To use DP to solve the SIC problem the conditions described on page 11 need to be satisfied. It has to be possible to divide the problem into stages, where each stage has a number of states associated with it. The stages in the SIC problem are the decision periods and the states of the system are the possible values of the remaining capacity, $x_t = 0, \ldots, C$. Furthermore, for a DP problem, a decision in one stage transforms one state into a state in the next stage. For the SIC problem, if a request is accepted, then the remaining capacity in the next stage is one less than the remaining capacity in the current stage, whereas if the request is rejected the remaining capacity is unchanged. Furthermore it is a condition that the final stage is solvable by itself. This is also satisfied by the SIC problem, since if a request arrives in the decision period immediately before departure and remaining capacity exists, then the request should be accepted and otherwise it should be rejected. Hence, it is obvious to try using dynamic programming to solve the SIC problem.

In the context of dynamic programming the above implies that the system of the SIC problem is as follows

- The stage is the decision period t.
- The state is the remaining capacity x.
- The decision is either to accept or reject the request.
- The random parameter is the demand for different fare classes.

Furthermore the system is seen to satisfy the conditions for Markov systems described on page 15. The first condition is satisfied, since the capacity of the aircraft is finite, hence the system has a finite number of states. In each decision period, if a request arrives and is accepted, then the fare of the requested class is obtained, thus the second condition is satisfied. Otherwise the reward in this decision period is zero. The third condition is satisfied, since the remaining capacity in the next decision period depends only on the remaining number of seats in the current decision period combined with the decision made in the current decision period. Finally, the transition probabilities for the SIC problem are fairly simple.

Value of j	$p_{ij}(u,t)$	
	$u_t =$ acc.	$u_t =$ rej.
$j = i$	$1 - \lambda$	1
$j = i - 1$	λ	0
Else	0	0

Table 2.1: Transition Probabilities.

All transition probabilities, $p_{ij}(u,t)$ for $j \neq i \wedge j \neq i-1$ are zero, since multiple requests are not considered. The transition probability between the states i and j are shown in Table 2.1 for $j = i - 1$ and $j = i$ given a specific decision u_t. If the decision u_t is to accept a booking request, then there are two possible outcomes. If a request is made, with probability λ, then capacity changes to $i - 1$, but if no request is made, with probability $1 - \lambda$, then capacity does not change. If the decision is to reject a booking request, then independent of whether a request is made or not, capacity does not change.

Hence, since all conditions are satisfied, the SIC problem is Markovian, and therefore it is possible to totally characterize the system's possible states in each stage.

Chapter 3

SIC without Trade-Up

In this chapter different solution methods for the SIC problem without trade-up will be described. In Section 3.1 two static methods, the EMSRa and EMSRb methods, are explained. Next, in Section 3.2 a dynamic programming model for the problem without trade-up will be introduced. For solving this model, two different methods, the L&H solution method and the B&P solution method, are described.

3.1 Static Solution Methods

It is well-known that determining an optimal solution to a DP model can be very time consuming, if the number of stages and states is large. Hence, alternatives to the DP model or for solving the DP model to optimality are necessary. In this section, two heuristics for solving the SIC problem are introduced.

These methods use the Expected Marginal Seat Revenue (EMSR) method, which determines nested booking limits. Recall that the nested booking limit for fare class i is the maximum number of seats, which can be sold to fare class i and all less expensive fare classes $i+1, \ldots, k$, where k is the least expensive fare class.

The EMSR method described in Section 3.1.1 is called EMSRa. In the past most airlines solved the SIC problem with EMSRa, but nowadays this method has been replaced by a similar heuristic called EMSRb. Hence, the EMSRb method will be described in Section 3.1.2.

3.1.1 EMSRa

In [4] the EMSRa solution method is described by Belobaba. To understand this method it is helpful to consider the SIC problem with only two fare classes. The starting point is to allocate all seats to class 2. Now let S_1^2 denote the number of seats, which are protected, i.e., reserved, for fare class 1 and therefore cannot be sold in class 2. Then the number of seats made available to class 2 is $C - S_1^2$.

Given a protection of S_1^2 seats for fare class 1, the probability that all requests for this fare class is accepted is given by

$$\Phi_1(S_1^2) = P[X_1 \leq S_1^2] = \int_0^{S_1^2} \varphi_1(X_1)\, dX_1$$

where φ_1 is the probability density function for the total number of requests for fare class 1, X_1. This implies

$$\begin{aligned} P[X_1 > S_1^2] &= \int_{S_1^2}^{\infty} \varphi_1(X_1)\, dX_1 \\ &= 1 - \Phi_1(S_1^2) = \overline{\Phi}_1(S_1^2), \end{aligned} \tag{3.1}$$

thus $\overline{\Phi}_1(S_1^2)$ is the probability of spill occuring, i.e., the probability of having to reject customers requesting fare class 1.

The EMSR for fare class 1, EMSR_1, is the expected marginal seat revenue, when the number of seats available to class 1 is increased by one. It is given by the product of the fare level for class 1, F_1, and the probability of being able to sell more than S_1^2 seats in fare class 1, $\overline{\Phi}_1(S_1^2)$, i.e.,

$$\text{EMSR}_1(S_1^2) = F_1 \cdot \overline{\Phi}_1(S_1^2)$$

The EMSRa procedure is to increase the number of seats protected for fare class 1 from class 2, S_1^2, by one as long as the expected marginal value of the next seat in fare class 1 is greater than or equal to the marginal value of selling the seat in fare class 2. Hence, increase S_1^2 as long as

$$F_1 \cdot \overline{\Phi}_1(S_1^2) \geq F_2.$$

The SIC problem usually consists of multiple fare classes. Hence, the solution method handles multiple fare classes as well. In this case the procedure is to consider fare classes in pairs and find the optimal seat allocation between the two classes considered. This is done in the same way as described above. Since the fare classes are considered in pairs, there is a risk that when the optimal seat allocation for pairs of classes are combined to give the total seat allocation for each class, this might not be the optimal solution for the entire problem.

For the case with multiple fare classes, let S_i^j denote the number of seats protected for class i from class j. The value of the seat allocation for class i from class j is determined by increasing S_i^j by one as long as

$$F_i \cdot \overline{\Phi}_i(S_i^j) \geq F_j, \qquad i < j, \quad j = 1, \dots, k, \tag{3.2}$$

where k is the number of fare classes. Once the seat allocations for each fare class from all lower-fare classes have been determined, it remains to find the booking limit for each class. The booking limit for fare class j, BL_j, is the maximum number of seats available for fare classes $j, j+1, \dots, k$. The booking limit for class j is given by

$$BL_j = \max\left[0, C - \sum_{i<j} S_i^j\right], \qquad j = 1, \dots, k. \tag{3.3}$$

It is seen that the booking limit is calculated by subtracting the sum of the protected seats from class j to all more expensive fare classes from the capacity, hence the booking limits are nested. Recall that with nested availability, full-fare classes can take seats at the expense of discounted classes.

The EMSRa model is static, but the problem to be solved is dynamic. To solve the dynamic problem the static model is applied a number of times with revised input data. For instance, the second time the model is solved, the bookings so far are known, thus the seat allocations can be updated in the following way

$$\sum_{i<j} S_i^j = \sum_{i<j} S_i^j(t) + \sum_{i<j} b_i^t$$

where $S_i^j(t)$ is the seat allocation calculated at time t from the demand forecast from time t to departure and b_i^t is the number of accepted requests in fare class i at time t. Thus the revised booking limit for fare class j at time t is given by

$$BL_j(t) = C - \sum_{i<j} S_i^j(t) - \sum_{i<j} b_i^t.$$

The booking limit for fare class j is constrained to be greater than or equal to zero and also to be no lower than the number of accepted requests in fare class j and all lower-fare classes, i.e.,

$$BL_j(t) = \max\left[C - \sum_{i<j} S_i^j(t) - \sum_{i<j} b_i^t, \sum_{l \geq j} b_l^t, 0\right] \tag{3.4}$$

The approximation of the nested booking limits gets worse the more classes are considered. The EMSRb heuristic is a better heuristic and will be described in the following section.

3.1.2 EMSRb

In [5] a heuristic similar to the EMSRa is described by Belobaba. This heuristic is called EMSRb. In the EMSRa method the expected marginal seat revenue for one fare class is compared with the fare of a lower-priced fare class. In the EMSRb solution method this procedure is replaced by a calculation of joint seat protection levels for all higher-fare classes relative to a given lower-fare class. These calculations are based on combined demand and a weighted price level for all classes with a higher fare than the one for which a booking limit is being calculated.

Denote the joint protection level for class i and all higher-fare classes $i-1,\ldots,1$ by π_i. Initially in the method, π_1 is determined, then π_2 is determined, etc.

The seat protection level for fare class 1 is determined by finding the largest value of π_1, which satisfies the following

$$F_{1,1} \cdot \overline{\Phi}_{1,1}(\pi_1) \geq F_2,$$

where $F_{1,1} = F_1$ and $\overline{\Phi}_{1,1}(\pi_1) = P[X_1 > \pi_1]$. Hence, this yields the same equation as in the EMSRa framework. Now this π_1 is used to find the booking limit for fare class 2 by

$$BL_2 = C - \pi_1.$$

To find the booking limit for fare class 3, the joint protection level for fare classes 1 and 2 is calculated by combining the demand and price levels for the two classes in the following way

$$\begin{aligned}
\overline{X}_{1,2} &= \overline{X}_1 + \overline{X}_2, \\
F_{1,2} &= \frac{F_1 \cdot \overline{X}_1 + F_2 \cdot \overline{X}_2}{\overline{X}_{1,2}}, \\
\overline{\Phi}_{1,2}(\pi_2) &= P\left[X_1 + X_2 \geq \pi_2\right], \\
\text{EMSRb}_{1,2}(\pi_2) &= F_{1,2} \cdot \overline{\Phi}_{1,2}(\pi_2).
\end{aligned}$$

where X_i is the random demand for fare class i and $\overline{X}_i$ is the expected value of X_i. Now the problem is to find the largest value of π_2 which satisfies

$$\text{EMSRb}_{1,2}(\pi_2) \geq F_3.$$

By using this joint protection level for classes 1 and 2, the booking limit for fare class 3 can be determined by

$$BL_3 = C - \pi_2.$$

The booking limit for fare class $i+1$ is determined by finding the joint protection level for fare class i and all higher-fare classes. This is done by calculating

$$\begin{aligned}
\overline{X}_{1,i} &= \sum_{n=1}^{i} \overline{X}_n, \\
F_{1,i} &= \frac{\sum_{n=1}^{i} F_n \cdot \overline{X}_n}{\overline{X}_{1,i}}, \\
\overline{\Phi}_{1,i}(\pi_i) &= P\left[\sum_{n=1}^{i} X_n \geq \pi_i\right], \\
\text{EMSRb}_{1,i}(\pi_i) &= F_{1,i} \cdot \overline{\Phi}_{1,i}(\pi_i).
\end{aligned} \tag{3.5}$$

The largest value of π_i which satisfies the following must be found

$$\text{EMSRb}_{1,i}(\pi_i) \geq F_{i+1}$$

The booking limit for fare class $i+1$ can then be found by

$$BL_{i+1} = C - \pi_i. \tag{3.6}$$

The EMSRb method can be used on a dynamic problem in the same way as the EMSRa method, where the booking limits are updated in a similar way, i.e.,

$$BL_{i+1}(t) = \max\left[C - \pi_i - \sum_{j<i+1} b_j^t, \sum_{l \geq i+1} b_l^t, 0\right], \tag{3.7}$$

where b_n^t is the number of accepted requests for fare class n at time t.

The EMSRb method yields better results than the EMSRa method and it is currently the method, which most airlines use.

3.2 Dynamic Programming Model

As mentioned previously, the static EMSRa and EMSRb heuristics were introduced, since it can be very time consuming to solve a dynamic programming model. A problem with these heuristics is, that it is difficult to incorporate time-dependent variability in the demand, since it is assumed that demand from the current time to departure can be described by a single random variable. To include the variability, a DP model for the SIC problem without trade-up is derived and two different algorithms for solving this are introduced. In [19] Lee and Hersh set up the DP model and it is solved to optimality by means of critical values. This solution method will be denoted the L&H solution method in all of the following, and it is described in Section 3.2.1. In [8] Bertsimas and Popescu use an approximation algorithm to solve the model. This method is denoted the B&P solution method, and it will be described in Section 3.2.2.

Contrary to the description of dynamic programming in Section 2.1, in all of the following decision period $t = 0$ corresponds to the end of the booking period, i.e., time of departure, since this is most common in the literature. The following simplifying assumptions are used in the model:

- Requests for different fare classes are independent, hence the demand for one class does not affect the demand for another class.
- A rejected request is lost sale, i.e., trade-up and recapture is not modelled.
- Demand is modelled as a stochastic process, i.e., the demand distributions are assumed to be known.
- Request probabilities vary with time.
- No more than one request arrives during a decision period.
- An accept/deny decision has to be made each time a request arrives.

The model is in the following first described in an intuitive way and afterwards in a mathematically accurate way. The demand intensity for a seat in a fare class at a point in time is modelled by a request probability, which varies with time. The booking period is split into a number of decision periods in which at most one request can arrive, where decision period T is at the beginning of the booking period and decision period 1 is closest to departure. A request for the highest-fare class will always be accepted as long as the capacity of the aircraft is not exceeded. Hence, the core problem consists of whether or not to accept or reject requests for seats in fare classes $2, 3, \ldots, k$, where k is the number of fare classes. I.e., the core problem is to determine the decisions in the dynamic programming algorithm as described in Section 2.1.1. To determine these decisions, an acceptance criterion must be derived. As mentioned previously, in the following this criterion is derived intuitively and afterwards it will be derived mathematically.

First an expression for the total expected revenue, $V_t(x)$, generated from decision period t to departure with a remaining capacity of x, is to be set up. If a request for fare class i is accepted in decision period t, then the total expected revenue, $V_t(x)$ is given by the value of accepting the request F_i plus the optimal total expected revenue obtained in decision periods $t-1,\ldots,0$ with a remaining capacity of $x-1$, i.e.,

$$V_t(x) = F_i + V_{t-1}(x-1).$$

Note that this expression is equivalent to the dynamic programming algorithm in (2.1) page 15.

If the request is rejected, then the total expected revenue is given by the optimal expected revenue from the remaining decision periods $t-1,\ldots,0$, but still with a remaining capacity of x, i.e., $V_{t-1}(x)$. Intuitively, an acceptance criterion is to accept a request for fare class i if the optimal expected revenue of accepting the request is higher than the optimal expected revenue when rejecting the request. This yields

$$F_i + V_{t-1}(x-1) \geq V_{t-1}(x). \tag{3.8}$$

This acceptance criterion is derived mathematically in the following by setting up a recursive formula for the total expected revenue, $V_t(x)$. This formula must include the case where the request is accepted, the case where it is rejected, and the case where no request arrives. The probability of no request in decision period t is given by P_0^t, hence, the expected revenue from no requests in period t is $P_0^t V_{t-1}(x)$. If a request arrives, the decision of whether to accept it or not depends on which fare class the request is for. As mentioned previously, if the request is for the most expensive fare class, this will always be accepted as long as the capacity of the aircraft is not exceeded. A request for fare class 1 yields the following expected revenue, $P_1^t(F_1 + V_{t-1}(x-1))$, where P_1^t is the probability of a request for fare class 1 in decision period t. If the request is for any of the fare classes $2, 3, \ldots, k$, the acceptance rule (3.8) must be incorporated. This is done in the following expression, $\sum_{i=2}^{k} P_i^t \max(F_i + V_{t-1}(x-1), V_{t-1}(x))$.

To summarize, the following recursion formula is obtained

$$V_t(x) = \begin{cases} P_0^t V_{t-1}(x) + P_1^t\big(F_1 + V_{t-1}(x-1)\big) & \\ +\sum_{i=2}^{k} P_i^t \max\big(F_i + V_{t-1}(x-1), V_{t-1}(x)\big), & \text{for } x > 0,\ t > 0 \\ & \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

The expected marginal value of a seat in decision period t, given a remaining capacity x, $\Delta V_t(x)$, is the increase in expected revenue by selling a seat in decision period t, i.e., $\Delta V_t(x) = V_t(x) - V_t(x-1)$. Rewriting equation (3.9) to include $\Delta V_t(x)$ yields

$$V_t(x) - V_{t-1}(x) = \sum_{i=1}^{k} P_i^t \max\left(F_i - \Delta V_{t-1}(x), 0\right). \tag{3.10}$$

For the derivation of this expression, see Appendix A.

The term $V_t(x) - V_{t-1}(x)$ is the expected opportunity cost of holding x seats from decision period t to $t-1$. From (3.10) it is seen, that it is more profitable to sell the requested seat in class i, if the first term in the maximization is positive. Thus, a request is accepted if the revenue obtained by accepting the request is larger than or equal to the expected marginal value of that seat. Hence, the acceptance rule can be expressed as

$$F_i \geq \Delta V_{t-1}(x). \tag{3.11}$$

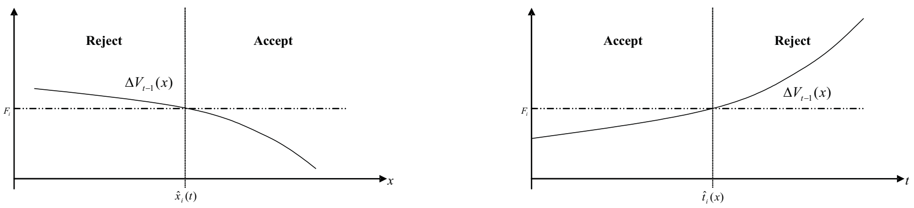
This was the mathematical derivation of the intuitive acceptance rule proposed in (3.8). I.e., a DP model has been set up.

In the following two sections, two algorithms for solving this model are described.

3.2.1 The L&H Solution Method

In [19], Lee and Hersh show that the function $\Delta V_t(x)$ is non-increasing in x for a fixed t. Thus, in a specific decision period the more remaining seats for sale, the lower expected marginal value of a seat. Furthermore, $\Delta V_t(x)$ is shown to be non-decreasing in t for a fixed x, i.e., given a specific remaining capacity, the expected marginal value of a seat is lower closer to departure. This is also reasonable, since when almost no time is left the airline is interested in selling the seat to any class.

The monotonicity of the function $\Delta V_t(x)$ in x and t is shown in Figure 3.1. Using the monotonicity of $\Delta V_t(x)$ leads to the fact, that the decision making throughout the booking process is a matter of determining a set of *critical values*. Figure 3.1 shows that for some value $\hat{x}_i(t)$ when $x < \hat{x}_i(t)$ the value of $\Delta V_{t-1}(x)$ is greater than F_i and when $x \geq \hat{x}_i(t)$ the value of $\Delta V_{t-1}(x)$ is smaller than F_i. According to (3.11) this means that for $x < \hat{x}_i(t)$ the request is rejected and for $x \geq \hat{x}_i(t)$ the request is accepted. Similarly a rule for the decision period t can be derived. This yields the following two sets of critical values



(a) Critical Booking Capacity.　　(b) Critical Decision Period.

Figure 3.1: Monotonicity of $\Delta V_t(x)$.

$\hat{x}_i(t)$ For decision period t, for each fare class i, a set of *critical booking capacities* exists, such that a request for a seat in fare class i is rejected for $x < \hat{x}_i(t)$ and accepted for $x \geq \hat{x}_i(t)$.

$\hat{t}_i(x)$ For a remaining capacity x, for each fare class i, a set of *critical decision periods* exists, such that a request for a seat in booking class i is rejected for $t > \hat{t}_i(x)$ and accepted for $t \leq \hat{t}_i(x)$.

For proof of the monotonicity of $\Delta V_t(x)$ and further explanation, see [19].

For making the accept/deny decision only one of the sets of critical values is necessary. This set of critical values is revised only at the time the demand forecasts and thus the request probabilities are revised. As mentioned in Section 1.1 the airline's booking system is such that decisions are made using booking limits for each fare class. Therefore the critical values must be transformed into booking limits. This is straightforward with the critical capacities, hence only these and not the critical desision periods are used in this report. The critical capacity for fare class i in decision period t, $\hat{x}_i(t)$, represents the lowest remaining capacity for a request for class i to be accepted, see Figure 3.1. Hence, the booking limits for fare class i in decision period t can be determined by subtracting the critical capacity $\hat{x}_i(t)$ from the remaining capacity and adding 1. This is illustrated by an example. Let the capacity be $C = 10$ and the number of accepted requests be $c = 3$, then the remaining capacity is $x = 7$. Furthermore, let the critical booking capacity for fare class i be $\hat{x}_i(t) = 7$. Assuming an incoming request for fare class i, then according to the critical capacity this request should be accepted, since $x \geq \hat{x}_i(t)$. Next time a request arrives, the remaning capacity is $x = 6$, hence if the request is for class i it should be rejected. These decisions should be the same, if booking limits are used instead. Thus, in the same scenario when $x = 7$, the booking limit for class i must be equal to 1, since then the first request will be accepted and all of the following will be rejected. This is the case if the booking limit is calculated as described above, i.e.,

$$BL_i(t) = x - \hat{x}_i(t) + 1 = 7 - 7 + 1 = 1.$$

Next time a request arrives the booking limit for fare class i is $BL_i(t) = 0$, hence if the request is for class i it is rejected.

To use the model proposed in Section 3.2 the request probabilities P_i^t must be determined. A technique for dividing the booking period into decision periods and evaluating the request probabilities is given in [19]. The booking period is partitioned into data intervals. A data interval is a time interval in the booking period in which the airline has collected demand data. Hence, demand forecasts are given for each data interval by the airline. The data intervals need not be of equal length, but it is assumed that a data interval satisfies that requests in that interval follow a Poisson process. The user of the model is required to decide upon the number of data intervals and these need not be of equal length. Once the data intervals have been determined, each data interval j is divided into ν^j decision periods of equal length. Let μ_i^j be the expected number of requests in data interval j for fare class i and let $\mu^j = \mu_1^j + \mu_2^j + \cdots + \mu_k^j$ be the expected number of requests in data interval j for all fare classes. Then the requests for each decision period is a Poisson process with mean μ^j/ν^j. A decision period is required to have no more than one request arrival. This is handled by increasing the number of decision periods in data interval j, ν^j, until $P(s \geq 2) \leq \epsilon$, where ϵ is a small probability and s is a random variable denoting the number of requests arriving during a decision period. Using the probability function for a Poisson process given by

$$P(s) = \frac{(\mu^j/\nu^j)^s \exp(-\mu^j/\nu^j)}{s!}, \quad \text{for } s = 0, 1, 2, \ldots \tag{3.12}$$

the value of ν^j can be determined from the following equation

$$1 - P(0) - P(1) \leq \epsilon, \quad \text{or} \quad 1 - \exp(-\mu^j/\nu^j) - (\mu^j/\nu^j)\exp(-\mu^j/\nu^j) \leq \epsilon.$$

When the number of decision periods in each data interval have been calculated the request probabilities can be computed. The requests for fare class i in decision period t in data interval j follow a Poisson process with mean μ_i^j/ν^j. Since the request probability is the probability of exactly one request arriving in decision period t, inserting in (3.12) with $s = 1$ yields the following

$$P_i^t = (\mu_i^j/\nu^j)\exp(-\mu_i^j/\nu^j). \tag{3.13}$$

Hence, the only data needed is μ_i^j. The method does not require much data storage either, since only the critical values need to be stored to make the accept/deny decision.

Using the L&H method requires calculations of the value function $V_t(x)$ for all combinations of remaining capacity and decision periods. Hence, computation time may get long for large problems, but the method is expected to yield good results.

3.2.2 The B&P Solution Method

The L&H solution method solves the model described in Section 3.2 by calculating the value of $V_t(x)$ for all combinations of decision periods t and remaining capacity x. This can be rather time consuming, even though it only has to be done each time the demand forecasts are revised. In [8] an approximation algorithm to the dynamic model in (3.10) page 26 is suggested by Bertsimas and Popescu. The idea in the algorithm is to approximate the value function $V_t(x)$ in (3.10) with the optimal value of the objective function of a linear programming (LP) problem, which is a deterministic analogue of the stochastic dynamic problem. In the deterministic problem it is assumed that the expected demand to come from time t to departure is also the actual demand to come. Let D_i^t denote the expected demand to come from time t to departure for fare class i. Assuming D_i^t is given, then for all $x \leq C$ and $t \leq T$, an LP model for the problem is easily obtained. The model is

$$LP(x,t) = \max \sum_{i=1}^{k} F_i \cdot y_i \tag{3.14}$$

$$\text{s.t.} \sum_{i=1}^{k} y_i \leq x \tag{3.15}$$

$$0 \leq y_i \leq D_i^t, \qquad \forall i,t \tag{3.16}$$

where y_i is the number of seats allocated to class i and the times t can be decision periods, data intervals or other times specified by the user. The objective in the LP model is to maximize revenue. Hence, the objective function value (3.14) is the revenue obtained by the allocation of seats. The constraint (3.15) expresses that the remaining capacity must not be exceeded and (3.16) ensures the number of allocated seats to be non-negative and less than or equal to the expected demand for the remaining t time periods. Given integer data, the optimal solution to this model will be integral. The reason for this is explained in the following. The constraint (3.15) can be written as $\mathbf{Ay} \leq x$ where the coefficient matrix $\mathbf{A}$ is a $1 \times k$ matrix of ones. Proposition 3.3 in [27] by Wolsey says that the linear program has an integral optimal solution if x is integer and if the coefficient matrix is totally unimodular. According to Definition 3.2 in [27] a matrix is totally unimodular if every square submatrix of the matrix has a determinant of +1, -1 or 0. The square submatrices in the coefficient matrix, $\mathbf{A}$ are 1×1 matrices of ones. The determinants of all these are 1 and hence the coefficient matrix is totally unimodular and the optimal solution to the LP is integral.

The approximation DP algorithm is as follows:

For all combinations of remaining capacity x and user-specified times t

1. For a class i request at time t with a remaining capacity of x, compute the LP-based estimate of the expected marginal value given by

$$\Delta V_{t-1}^{LP}(x) = LP(x, t-1) - LP(x-1, t-1).$$

2. Accept the request for class i if and only if its fare F_i exceeds the expected marginal value estimate, i.e.,

$$F_i \geq \Delta V_{t-1}^{LP}(x).$$

3. Go to step 1 and iterate as long as $x < C$ for each $t \leq T$.

Note that the acceptance rule in step 2 is given as in the L&H method, see equation (3.11). The reason that the value function does not necessarily need to be approximated in each decision period is that the value of $V_t^{LP}(x)$ is independent of $V_{t-1}^{LP}(x)$, as opposed to the value function in the DP model. In the above algorithm, at each stage a decision is applied, which is optimal when the uncertain parameter, the actual demand to come for each class i is fixed at the expected value D_i^t. According to [7] by Bertsekas, this is a certainty equivalent control (CEC) policy, which is a suboptimal control scheme.

A problem with the CEC policy is that it uses a deterministic approach, thus, it does not take the stochasticity of the problem into account. An extension to this algorithm is therefore in some way to incorporate the variability in demand. This can be done by using Monte Carlo demand estimation. Assume the cumulative demand to come for fare class i from time periods $t-1$ to departure follows a certain distribution. Then r samples from this distribution can be generated, $\widehat{D}_{i,1}^{t-1}, \ldots, \widehat{D}_{i,r}^{t-1}$. For each of these the expected marginal value $\Delta V^{LP}_{t-1,j}(x)$ is calculated and the marginal value used in step 1 in the algorithm is then an average of these, i.e.,

$$\Delta V^{LP}_{t-1}(x) = \tfrac{1}{r}\sum_{j=1}^{r} \Delta V^{LP}_{t-1,j}(x)$$

where $\Delta V^{LP}_{t-1,j}(x) = LP_j(x, t-1) - LP_j(x-1, t-1)$ and $LP_j(x, t-1)$ is the optimal value of the LP (3.14) with the jth sample $\widehat{D}^{t-1}_j$ as the demand.

The B&P solution method with Monte Carlo demand estimation will not be used further in this work. This is due to the fact that running times for the methods are critical and calculating the value function for each sample will take approximately the number of samples times longer than just calculating the value function with a deterministic demand. Hence, only the regular B&P solution method will be used in all of the following. There are other approximation algorithms, though, these are described for the SIC problem with trade-up.

The B&P method is an approximation method for solving the model proposed in Section 3.2. Hence, the revenue obtained when using this method is expected to be smaller than the revenue obtained when using the L&H method, but running times are expected to be shorter.

3.3 Implementation

The four different solution methods for the SIC problem without trade-up described in the previous sections are the EMSRa heuristic, the EMSRb heuristic, the L&H method and the B&P method. These methods are implemented in Matlab Version 7.0 and the programs for these are enclosed on the CD in the folder *SIC without TU/Main Functions*. In Table C.1, Appendix C, a list of the main functions described in this section is given. Furthermore, in this table the different inputs and outputs used by the programs are given. The description of these variables can be seen in Table C.2, Appendix C.

3.3.1 Simulation

Simulation is often used by mathematicians and operations researchers to analyze problems, which are so complicated, that a purely theoretical treatment is practically impossible. Simulations are numerical experiments, where models for the system under consideration are programmed and the experiment is to run the program with different sets of input. Usually randomness plays an important role in the models under consideration. A stochastic simulation is when this randomness is included in the simulation.

In the simulation in this report the aim is to simulate an arrival process, i.e., simulate a booking process for a given flight departure. This is an event-driven model, which means that arrivals occur at irregular and random time intervals. It is assumed that a Poisson process can be used to describe the process of arrivals. Randomness plays an important role since the arrival times of the requests are unknown and so is the class, which is requested.

In the following a Poisson process is briefly described. Let $N(t)$ be a random variable denoting the number of events that occur during the continuous time interval $[0, t]$, and let $n = 0, 1, 2, \ldots$ be the number of events. Then $P_n(t) = P\{N(t) = n\}$ is the probability that exactly n events occur in the time interval $[0, t]$. If this probability can be characterized by the following three items, then $P_n(t)$ is called a Poisson process and the random variable $N(t)$ is a Poisson random variable:

1. The number of events that occur during a time interval is independent of the number of events that have occured over any other non-overlapping time interval. Thus, the number of events occuring in the time interval $[t_1, t_2]$ is statistically independent of the number of events that occur in the time interval $[t_2, t_3]$.

2. The probability that a single event occurs over a *short* time interval $[t, t + h]$ is approximately proportional to h. More accurately $P_1(h) = \lambda h + \varepsilon$, where ε is a small number and λ is the number of arrivals in that time interval.

3. The probability that more than a single event occurs during a *short* time interval $[t, t + h]$ is basically zero. That is $P_n(h) = \varepsilon, n > 1$.

The simulation process is implemented in the program `Simulation.m` and this is used for the simulation in all instances, where trade-up is not incorporated. From the expected demand in each data interval, a matrix of probabilities is calculated such that the (i, j)th element is the probability that an arrival in data interval i is for fare class j. Next, the end time of each decision period is calculated, such that these times can be used to determine in which decision periods the arrivals occur.

The arrival times are generated as a Poisson process, and since these have exponentially distributed inter-event times, a Poisson process can be simulated by generating a sequence of exponentially distributed random variables, representing the times at which a passenger places a request. These random variables can be found by using the following formula until the desired number of arrival times has been obtained.

$$t_{new} = t_{old} - \mu \cdot \ln(\mathrm{RND}), \tag{3.17}$$

where μ is the mean inter-event time and RND is a uniformly distributed random number in the interval $[0, 1]$. Note that the arrivals are simulated from time T to 0. To use (3.17), the mean inter-event time μ must be determined. This is done by dividing the length of the data interval in which the previous arrival occured with the expected number of arrivals in that data interval. This way of determining the mean inter-event times may cause problems if some data intervals have a very low expected demand. I.e., when calculating the mean inter-event time, this number goes toward infinity as the total expected demand in that data interval goes toward zero. Since

$$\mathrm{RND} \in [0, 1] \quad \Rightarrow \quad \ln(\mathrm{RND}) \in (-\infty, 0],$$

and $\mu \to \infty$ if the expected number of arrivals in the data interval of the previous arrival is extremely small, then from (3.17) it is seen that the time between the previous arrival and the incoming arrival gets very large. This may cause the arrivals to skip one or more data intervals and this can happen even though the expected demands in the skipped data intervals are high. To overcome this problem some precautions must be made in the program `Simulation.m`. The stopping criterion is made such that the arrival process only terminates if both the booking time has run out *and* if the previous arrival occured in data interval 1. Furthermore, it is examined whether the following two conditions are satisfied and if both conditions are satisfied the incoming arrival is not stored. The conditions are

- The time between the previous arrival and the incoming arrival is larger than the length of the data interval after the previous arrival.

- The previous arrival and the incoming arrival occur in different data intervals.

In Figure 3.2 two different arrival scenarios are illustrated. Arrival scenario 1 is seen to only satisfy the condition in the first item, whereas arrival scenario 2 satisfies both conditions. In arrival scenario 1 both arrivals are to be stored, but in arrival scenario 2 a data interval is skipped and hence the second arrival is not stored. Instead t_{old} in the equation (3.17) is set to be the beginning of the data interval immediately after the previous arrival, i.e., data interval 3 in the example.

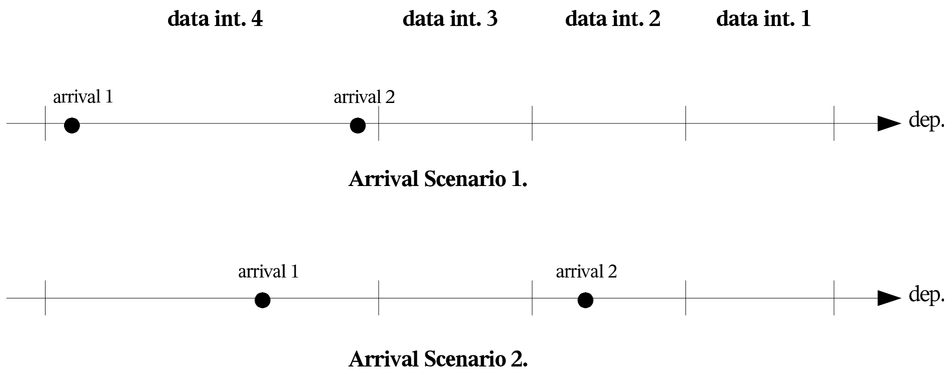


Figure 3.2: Two Different Arrival Scenarios.

This way the problem with too large inter-event times is overcome. The arrival times obtained by the program are stored in a vector and arrivals are generated until the time of departure, which is at time $t = 0$.

The times obtained are in *continuous time*, and the matrix with booking limits obtained when using L&H and B&P is indexed by the fare class and the *decision periods* in which the arrival occurs. Therefore, the continuous times are transformed into decision periods.

Now it remains to determine which fare class each passenger requests. For each data interval, the probability of a request for each class must be found. This is done by, for each data interval, dividing the number of expected requests for a specific class by the total number of expected requests for all classes. The probability is denoted d_i for fare class i. Let $\mathbf{v}$ be a vector given by

$$\begin{aligned}\mathbf{v} &= [d_1, d_1 + d_2, d_1 + d_2 + d_3, \ldots, d_1 + d_2 + \cdots + d_k] \\ &= [d_{1,1}, d_{1,2}, d_{1,3}, \ldots, d_{1,k}]\end{aligned}$$

where k is the number of different fare classes and $d_{1,j} = d_1 + d_2 + \cdots + d_j$ is the cumulated probabilities of a request for class 1 to j and since the probabilities sum to one, $d_{1,k} = 1$. A request can then be simulated to be for a specific class by generating a uniformly distributed random number, RND $\in [0, 1]$ and then finding the index of the smallest element in $\mathbf{v}$, which is larger than RND, since this index corresponds to the requested class. The outputs from the program are four vectors containing the arrival times, the requested classes, the data intervals in which the arrivals occur and the decision periods in which the arrivals occur. See Table C.1 in Appendix C for inputs to and outputs from the program and Table C.2, Appendix C, for explanation of these variables.

3.3.2 The Static Methods

The EMSRa heuristic is implemented in the program `EMSR.m`. Initially in the implementation the probabilities $\overline{\Phi}_i(S_i^j)$, defined in Section 3.1.1, are calculated for all combinations of fare classes, i, and seat allocations, S_i^j. This is done by finding the probability that the total demand for fare class i for the entire booking period is greater than the number of seats protected for fare class i from class j, S_i^j. The built-in Matlab function `poisscdf.m` is used and this function computes the Poisson cumulative distribution function with parameters $1, \ldots, C$ at the values of the forecasted demand for the entire booking period. Thus, the values of $\overline{\Phi}_i(S_i^j)$ are stored in a $C \times k$ matrix, where k is the number of fare classes and C is the capacity of the aircraft. Then initial protection levels for each fare class from all lower-fare classes are calculated. The protection level for class i from fare class j is determined by finding the largest value of S_i^j, which satisfies $F_i \cdot \overline{\Phi}_i(S_i^j) \geq F_j$, as given in (3.2) page 21. From the initial protection level for class i, the initial booking limit for class i is calculated by the maximum of zero and the capacity minus the sum of the protection levels for all higher-fare classes, $1, \ldots, i-1$, see (3.3) page 21.

The arrivals from the simulation process are then considered one by one. Before each arrival it is determined whether an update of the booking limits should be made. This is done by determining whether the next arrival from the simulation process arrives after a decision period where an update must be made according to an input vector containing the decision periods in which an update should be made. If an update of the booking limits must be made, this is done first, and the arrival is accepted or rejected according to the new booking limits. The update of the booking limits is calculated similarly to the initial booking limits by first calculating protection levels for each class from all lower-fare classes. Again a matrix with the probabilities $\overline{\Phi}_i(S_i^j)$ is calculated for all combinations of fare classes and seat allocations by using Matlabs `poisscdf.m`. This time the function is used with the total demand for fare class i from the time at which the booking limits are updated and until departure. The updated booking limit for class i is calculated by using (3.4) page 21.

Now a passenger requesting fare class i is accepted if the booking limit for fare class i minus the number of already accepted passengers in fare classes $i, \ldots, k$ is greater than zero and if none of the booking limits for higher fare classes are exceeded by accepting the request. The last condition is due to the fact that the booking limits are nested as described in Section 1.1. If a passenger is accepted in fare class i, then 1 is added to the number of accepted passengers in fare class i and the fare for that class F_i is added to the total revenue.

The procedure is repeated until either all seats in the aircraft have been sold, no more requests for seats are received or the booking period is terminated.

The EMSRb heuristic is implemented in the program `EMSRb.m`. The implementation of the EMSRb heuristic is similar to the implementation of the EMSRa heuristic. Hence, only the differences in the implementation are described. Initially in the implementation of the EMSRb heuristic, joint demand for all fare classes is calculated. This is due to the fact that in EMSRb seat protection levels are calculated jointly for all higher-fare classes relative to a given lower-fare class, as opposed to EMSRa, where seat protection levels are calculated for all higher fare classes from all lower-fare classes pairwise.

The probabilities $\overline{\Phi}_{1,i}$ are found for all fare classes i by determining the probabilities

$$\overline{\Phi}_{1,i}(\pi_i) = P\left[X_1 + X_2 + \cdots + X_i > \pi_i\right], \qquad i = 1, \ldots, k-1,$$

where k is the number of classes, X_j is the demand for fare class j and π_i is the number of seats protected for classes 1 to i from class $i+1$. Again the built-in Matlab function `poisscdf.m` is used and hence $\overline{\Phi}_{1,i}(\pi_i)$ is a $C \times k$ matrix, where k is the number of fare classes and C is the capacity of the aircraft.

Then joint fares are calculated by

$$F_{1,i} = \frac{\sum_{n=1}^{i} F_n \cdot \overline{X}_n}{\overline{X}_{1,i}}.$$

These joint fares are set to zero, whenever the denominator is zero. This happens when the sum of the expected demand for all fare classes in a data interval is zero, which is only the case when the demand for each class in that data interval is zero, since the demand is non-negative. Therefore the numerator is zero as well, and hence the joint fare is zero. The joint protection levels are calculated using these joint fares and the updated booking limit for class i is calculated by (3.7) page 23. The update of the booking limits and the procedure when a passenger is to be accepted or rejected is exactly the same as in the implementation of the EMSRa heuristic. The outputs from both `EMSR.m` and `EMSRb.m` are the revenues obtained from the simulation process.

3.3.3 The Dynamic Methods

The L&H method is implemented in the Matlab program `LHBL.m`. Initially, the request probabilities P_i^t for each decision period t and each fare class i are calculated using (3.13) page 29. By using these request probabilities, the total expected revenue $V_t(x)$ can be calculated using (3.10) page 26 for each decision period t and each remaining capacity x.

In Section 3.2.1 it is described how an incoming request is accepted or rejected by comparing the remaining capacity with a critical booking capacity. The critical capacities are calculated by finding the smallest value of remaining capacity for which $\Delta V_t(x)$ is lower than or equal to the fare for the requested class. This is done for each decision period and each fare class and thus the critical booking capacities are stored in a $k \times T$ matrix, where k is the number of fare classes and T is the number of decision periods. When the critical capacities have been calculated these can easily be transformed into booking limits, and this is done as described in Section 3.2.1. This matrix is given as output from the program.

The program `calcRevenue.m` is used to calculate the revenue from the simulation process by using the booking limits obtained by the program `LHBL.m`. The inputs to this function can be seen in Table C.1, Appendix C. When using the function `calcRevenue.m` to calculate the revenue after using the solution method from L&H, there are two different input scenarios. If there are only eight inputs to the function, the booking limits are updated in each decision period, hence the entire matrix of booking limits from the method is used. Otherwise the vector specifying the decision periods in which the booking limits need to be updated is non-empty and then the booking limits are only updated at prespecified times. Of course this method is less accurate than updating in each decision period since not all the available information is used. This is an option when using the function, since then the method is directly usable in the airline's booking system as it is now.

Whether to accept or reject a booking request is considered as the requests occur. A request for fare class i in decision period t is accepted or rejected by looking at the (i, t)th element in the matrix containing the booking limits minus the total number of passengers who have already been accepted. If this number is larger than or equal to zero, the request should be accepted and rejected otherwise. When a passenger is accepted, 1 is added to the number of accepted passengers and the fare of the requested class F_i is added to the revenue obtained by accepting previous passengers for the flight. The output from the program is the revenue obtained from the simulation.

The program `PopBL.m` contains the implementation of the B&P method. This implementation is made very similarly to the implementation of the L&H method. Hence, only the differences between the two functions are described. Instead of calculating the request probabilities, the expected demand to come for each decision period and each fare class is calculated. This yields a $T \times k$ matrix, where T is the number of decision periods and k is the number of fare classes.

Now, as described in Section 3.2.2, an LP must be solved for each decision period and each remaining capacity. Since the LP is deterministic, the expected demand is assumed to be the actual demand to come, hence, the LP is easily solved by first accomodating the demand most valuable to the airline, then fill up with the second most valuable demand, then the third most, etc. The matrix obtained by this procedure has $V_t^{LP}(x)$ as entries for each decision period and each remaining capacity, and is an approximation to the matrix containing the values of $V_t(x)$ described in Section 3.2.1. Thus, what remains to be done is to calculate the matrix of critical booking capacities. This is done as in the program `LHBL.m`.

The program `calcRevenue.m` is used to calculate the total revenue from a simulation process when using the B&P method. Again, there are two input scenarios. As with the L&H method when there are eight inputs, the booking limits are updated in each decision period. The difference is when there are nine inputs. Then the matrix containing booking limits is an empty matrix and a vector specifying the decision periods in which an update must be made is input to the program. Next a linear program is solved in each of the decision periods specified in the input vector. This can be done since the LP's in the B&P method only depend on one time-period and can hence be solved independently of each other. Thus, instead of calculating the entire matrix with booking limits, a much simpler approach is to only solve the linear programs for the decision periods specified in the input vector. This procedure is much less time-consuming than calculating the entire matrix of booking limits.

Chapter 4

SIC with Trade-Up

In Chapter 3, four different solution methods for the basic SIC problem were described. These methods, however, do not take trade-up into account. In this chapter different solution methods for the SIC problem with trade-up will be described.

It is assumed, that after a passenger has been rejected this person has the following choices

- Deviate, i.e., travel with a competitor or cancel the trip.
- Be recaptured, i.e., travel on a different flight but with the same airline.
- Trade up, i.e., buy a more expensive ticket than initially intended on the same flight.

Deviation inevitably implies lost sales for the airline and since only single flights are considered in this report, recapture is not modelled. Trade-up, however, can be modelled for a single flight, which may imply a higher revenue for the airline. Since by modelling trade-up, the airline has an opportunity to use the additional information about passengers' preferences in their decision making.

It may not always be preferable to include trade-up in the models, though. Trade-up is very dependent on the competition in the market. Hence, the market under consideration must be investigated carefully and data about customer behaviour must be collected before a solution method for the SIC problem with trade-up is applied. For instance, in a market where competitors have many flights it is assumed that few potential customers trade up, since it is easy to book a ticket on a competitor flight. Similarly, in a market where few competitor flights exist, it is assumed that many passengers trade up, since it is difficult to book a ticket with another airline.

In a market where trade-up behaviour occurs, two different kinds of fare classes exist. The first kind are trade-up classes, i.e., passengers are assumed to trade up from and to these classes. The second kind are independent classes as for the problem without trade-up, for instance transfer traffic, which are fare classes from and to which passengers cannot trade up. Hence, these classes are treated as independent of the trade-up classes. To simplify matters, in this report it is chosen to ignore the independent classes when a trade-up market is considered.

Now there are two types of trade-up markets. The simplest one is where the fares in the different trade-up classes have the same fare conditions, so they only differ in price. For example, in BA's Domestic and European fare structure, five classes have exactly the same conditions. In such a situation, everybody tries to book the lowest open class. In a less simple world, the fares in the trade-up classes differ not only by price, but also by fare conditions. Thus, a passenger may request a class that is not the lowest open class because of the more relaxed fare conditions. For example, in addition to a number of non-flexible fares with the same conditions, BA also offers semi-flexible and fully flexible fares, and trade-up from a non-flexible fare to one with more flexibility can be observed.

To make this report usable regardless of the assumptions on the trade-up market, it is chosen to model both cases. I.e., when trade-up is considered, two cases apply and these are

1. Buying conditions for different trade-up classes differ and hence, demand can occur for all classes, not only the least expensive class. The models for the SIC problem with these assumptions will be called the "general models".

2. Trade-up classes have the same buying conditions, thus, when a request arrives this will always be for the least expensive class in the trade-up market. These models will be denoted the "simplified models", since the SIC problem get simpler when these assumptions apply.

In both cases, if a passenger chooses to trade up, this will always be to the least expensive open class, since fewer buying conditions apply to the ticket, the more expensive the ticket is. Hence, the buying conditions in fare class 4 will always satisfy the needs of a passenger who requests fare class 5, which is a less expensive class.

In the different solution methods for the SIC problem with trade-up, two different varieties of the trade-up rates will be used depending on the method. The first trade-up rate is denoted $TU_{i,j}$ and this has the following definition

> *The proportion of passengers that would buy classes i or all higher-fare classes that would trade up further to classes $1, \dots, j$.*

The trade-up rates collected by the airline are $TU_{k,i}, i = 1, \dots, k$, i.e., the trade-up rates from the lowest class in the market, class k, to another class i and all higher classes. The rates $TU_{i,j}$ are then calculated by

$$TU_{i,j} = \frac{TU_{k,j}}{TU_{k,i}}, \qquad i > j. \tag{4.1}$$

The second trade-up rate is denoted $q_{i,j}$ and is defined by

> *The proportion of passengers that would buy class i or all higher-fare classes that would trade up further only to class j.*

This trade-up rate is calculated by

$$q_{i,j} = TU_{i,j} - TU_{i,j-1} = \frac{TU_{k,j} - TU_{k,j-1}}{TU_{k,i}}, \qquad i > j, \tag{4.2}$$

where, by definition, $TU_{k,0} = 0$.

In the following sections the models for the two cases of the SIC problem with trade-up will be described. In Section 4.1 the models for the general SIC problem with trade-up are introduced. Next, in Section 4.2 the simplified SIC problem with trade-up is described.

4.1 General SIC with Trade-Up

In this section the methods for solving the problem with the general assumptions are set up and described. First the EMSRb method with trade-up is explained. This method is derived by combining a decision rule with trade-up with the EMSRb method without trade-up. Next a DP model is set up for the general SIC problem with trade-up. For solving this an exact method, the You solution method, is described. Furthermore three approximation methods for solving the DP model are suggested. These are the B&P method with trade-up, the C,G&J method and the C&H method.

4.1.1 EMSRb with Trade-Up

It is obvious to start with the well-known EMSRb model described in Section 3.1.2 when solving the SIC problem with trade-up, since this solution method is assumed to produce decent results with fairly short running times. The original EMSRb method does not take trade-up into account, therefore the heuristic is combined with a decision rule which incorporates this. In [6] Belobaba and Weatherford describe a combination of the EMSRb method and a decision rule with trade-up.

The idea in the decision rule with trade-up is to determine whether the seat allocation for a specific class should be increased by one seat each time a request arrives. The passenger should be accepted, i.e., the number of seats allocated for the requested class should be increased by one seat, if the expected net revenue from accepting the request is greater than zero. The expected net revenue is the revenue gained when selling the seat minus the expected loss in revenue if the remaining capacity fails to meet future demand for more expensive fare classes.

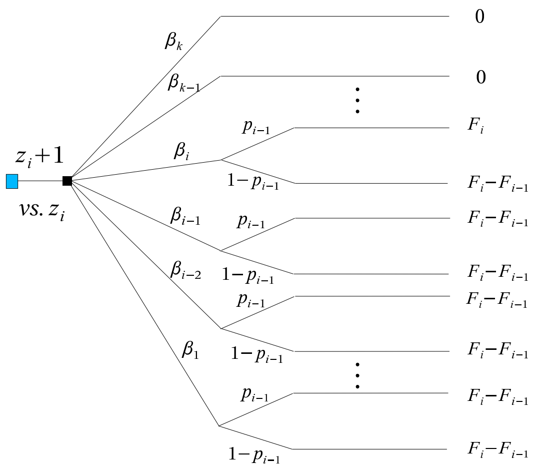


Figure 4.1: Decision Tree for Class i.

The decision of whether to increase the number of seats or not, is easiest to survey with a decision tree. In Figure 4.1 the decision tree for determining if the number of seats for class i should be raised from z_i to $z_i + 1$ is seen. The tree should be contemplated in the following way. An incoming request is for one of the classes $1, \ldots, k$. The parameter β_i is the probability that the incoming request is a strict class i request. The value p_{i-1} is the probability that the remaining capacity after selling an additional seat in class i will not fail to meet subsequent class $i-1$ demand. Trade-up between non-adjacent classes is ignored in [6]. Therefore only subsequent demand for class $i-1$ is considered as opposed to subsequent demand for classes $1, \ldots, i-1$. This is also why the revenue gained, i.e., the values to the right in the figure, can only be $F_i - F_{i-1}$ as opposed to for instance $F_i - F_{i-2}$. Sales in lower-fare classes are expected to be closed first followed by sales in higher-fare classes.

Hence, when considering whether to sell an additional seat in fare class i, classes $i+1, \ldots, k$ have already been closed for sales. Therefore, if requests for these classes occur, the revenue gained is zero, see Figure 4.1.

The expected net revenue is obtained by the sum of all possible outcomes when a passenger arrives, weighted with the probabilities for a specific outcome, i.e., the decision rule is as follows. Accept a request for fare class i if

$$\begin{aligned}&\beta_k \cdot 0 + \beta_{k-1} \cdot 0 + \cdots + \beta_i \cdot p_{i-1} \cdot F_i + \beta_i \cdot (1-p_{i-1}) \cdot (F_i - F_{i-1}) \\ &\quad + \beta_{i-1} \cdot p_{i-1} \cdot (F_i - F_{i-1}) + \beta_{i-1} \cdot (1-p_{i-1}) \cdot (F_i - F_{i-1}) \\ &\quad + \cdots + \beta_1 \cdot p_{i-1} \cdot (F_i - F_{i-1}) + \beta_1 \cdot (1-p_{i-1}) \cdot (F_i - F_{i-1}) > 0.\end{aligned}$$

This expression can be simplified to the following

$$\left(\frac{\beta_i}{\beta_i + \beta_{i-1} + \cdots + \beta_1}\right) \cdot p_{i-1} > \frac{F_{i-1} - F_i}{F_{i-1}}. \tag{4.3}$$

Therefore, when a request for fare class i arrives and if (4.3) holds, the number of seats reserved for class i should be increased by one and otherwise it should not.

Since $\sum_{i=1}^{k} \beta_i = 1$ the decision rule for class k is given by

$$\beta_k \cdot p_{k-1} > \frac{F_{k-1} - F_k}{F_{k-1}}.$$

Recall, that the trade-up rates from class $i+1$ to class i and all higher-fare classes, $TU_{i+1,i}$ is defined by the following

> *The proportion of passengers that would buy classes $i+1$ or all higher-fare classes that would trade up further to classes $1, \ldots, i$.*

Hence, the trade-up rate from class $i+1$ to class i and all higgher-fare classes is given by

$$TU_{i+1,i} = \frac{\beta_i + \ldots + \beta_1}{\beta_{i+1} + \ldots + \beta_1} = 1 - \frac{\beta_{i+1}}{\beta_{i+1} + \ldots + \beta_1}. \tag{4.4}$$

As mentioned previously, p_i is the probability that the remaining capacity after selling an additional seat in class $i+1$ will not fail to meet subsequent class i demand. From Section 3.1.2 recall that $\overline{\Phi}_i(\pi_i)$ is the probability of having to reject requests for fare classes $1, \ldots, i-1$, thus $\overline{\Phi}_i(\pi_i) = 1 - p_i$. Furthermore from this section recall the joint fares given by

$$F_{1,i} = \frac{\sum_{n=1}^{i} F_n \cdot \overline{X}_n}{\overline{X}_{1,i}},$$

where $\overline{X}_{1,i} = \sum_{n=1}^{i} \overline{X}_n$. By inserting the joint fares $F_{1,i}$, the probabilities $\overline{\Phi}_i(\pi_i) = 1 - p_i$ and (4.4) in the decision rule (4.3) and replacing the inequality sign with an equality sign the following is obtained

$$\frac{\beta_{i+1}}{\beta_{i+1} + \beta_i + \cdots + \beta_1} \cdot (1 - \overline{\Phi}_i(\pi_i)) = \frac{F_{1,i} - F_{i+1}}{F_{1,i}}$$

$$\Updownarrow$$

$$(1 - TU_{i+1,i}) \cdot (\overline{\Phi}_i(\pi_i) - 1) = \frac{F_{i+1} - F_{1,i}}{F_{1,i}}$$

$$\Updownarrow$$

$$\overline{\Phi}_i(\pi_i) \cdot (1 - TU_{i+1,i}) \cdot F_{1,i} + F_{1,i} \cdot TU_{i+1,i} = F_{i+1}$$

$$\Updownarrow$$

$$\overline{\Phi}_i(\pi_i) = \frac{F_{i+1} - F_{1,i} \cdot TU_{i+1,i}}{F_{1,i} \cdot (1 - TU_{i+1,i})}. \tag{4.5}$$

This is a combination of the decision rule (4.3) and EMSRb without trade-up, thus $F_{1,i}$ is given as in (3.5) page 23 and the task is to find the protection level π_i such that (4.5) is satisfied.

The booking limit for fare class $i + 1$ can then be found the same way as in Section 3.1.2 by

$$BL_{i+1} = C - \pi_i.$$

where C is the total capacity of the aircraft.

As the EMSRa and EMSRb solution methods described in Sections 3.1.1 and 3.1.2, this EMSRb method with trade-up is a static solution method, which can be applied a number of times to a dynamic problem. The method is a heuristic, thus it is assumed to be reasonably fast, but the results obtained are expected to be worse than those obtained with exact solution methods solving models which incorporate trade-up.

4.1.2 The You Solution Method

As for the problem without trade-up, the SIC problem with trade-up can be formulated as a dynamic programming problem. In [28] You models the problem as a dynamic Markov Decision Problem with trade-up. This model corresponds to the model in [19] by Lee and Hersh but now trade-up is incorporated.

The formulation of the value function, when trade-up is included, is slightly different than the value function for the problem without trade-up given in (3.9) page 25. Recall, that the value function $V_t(x)$ denotes the expected revenue at time t with a remaining capacity of x. A request for fare class i is only accepted, if the expected revenue obtained by accepting the request is greater than or equal to the expected revenue when rejecting the request, i.e.,

$$F_i + V_{t-1}(x-1) \geq U_t^i(x), \tag{4.6}$$

where $U_t^i(x)$ is the total expected revenue generated with t decision periods and x seats remaining when a request for fare class i is rejected. The following value function is obtained

$$V_t(x) = \begin{cases} P_0^t V_{t-1}(x) + P_1^t\big(F_1 + V_{t-1}(x-1)\big) & \\ + \sum_{i=2}^k P_i^t \max\big(V_{t-1}(x-1) + F_i, U_t^i(x)\big) & \text{for } x > 0,\, t > 0 \\ & \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

where $P_0^t = 1 - \sum_{i=1}^k P_i^t$. If a request for fare class $i \geq 2$ is rejected, then with a certain probability the passenger requests a higher-fare class instead. Therefore, a set of higher-fare classes from the set $\{1, 2, \ldots, i-1\}$ is offered, and thus this set must be determined. Let the set, which is offered, be denoted A_0 and let the set A_i be given by $A_i = \{1, 2, \ldots, i-1\}$, i.e., $A_0 \subseteq A_i$. For $x > 0$ and $t > 0$, $U_t^i(x)$ is then given by

$$U_t^i(x) = \max_{A_0 \subseteq A_i} \left\{ \sum_{n \in A_0} q_{i,n}\Big(F_n + V_{t-1}(x-1)\Big) + \Big(1 - \sum_{n \in A_0} q_{i,n}\Big) V_{t-1}(x) \right\} \tag{4.8}$$

where $q_{i,n}$ is the probability that the rejected customer for class i is willing to trade up to fare class n and the set A_0 is the decision variable.

Recall the assumption that when a request is rejected and the rejected passenger chooses to trade up, this will always be to the least expensive open class. Therefore it is only necessary to determine the lowest-fare class which is offered. This class is denoted j and it is a function of t, i and x, i.e., $j_t^i(x)$. Applying the assumption to (4.8) yields the following expression

$$U_t^i(x) = \max_{j \in A_i} \Big\{ TU_{i,j}\Big(F_i + V_{t-1}(x-1)\Big) + \Big(1 - TU_{i,j}\Big) V_{t-1}(x) \Big\}.$$

Notice that different trade-up rates have been used in the two expressions for $U_i^t(x)$. The definitions of the are given in (4.2) and (4.1) page 41. The acceptance/rejection process consists of two steps,

1. Reject or accept a request.

2. If a request is rejected, which fare class should be offered.

The decision rules for these items will be developed in the following. Recall that

$$\Delta V_t(x) = V_t(x) - V_t(x-1), \quad \text{for } x > 0,\ t \geq 0$$

is the expected marginal value of the xth seat in decision period t. Incorporating this expression in the formulas for $V_t(x)$ and $U_t^i(x)$ yields

$$\begin{aligned} V_t(x) =& V_{t-1}(x) + P_1^t\Big(F_1 - \Delta V_{t-1}(x)\Big) \\ &+ \sum_{i=2}^{k} P_i^t \max\left\{F_i - \Delta V_{t-1}(x), \max_{j\in A_i} g_t^i(x,j)\right\} \end{aligned} \tag{4.9}$$

$$U_t^i(x) = V_{t-1}(x) + \max_{j\in A_i} g_t^i(x,j), \quad i \geq 2 \tag{4.10}$$

where

$$g_t^i(x,j) = TU_{i,j}\Big(F_j - \Delta V_{t-1}(x)\Big), \quad i \geq 2. \tag{4.11}$$

Consider the value function given in (4.9) and the value function for the problem without trade-up given in (3.10) page 26. The difference is that the last term in the maximization in (4.9) is positive, whereas the last term in the maximization for the problem without trade-up is zero. This happens because in the problem without trade-up, if a class is closed for sales all demand for that class is lost. However, for the problem with trade-up, if a class is closed, the passenger requesting that class may choose to trade up to another class which is offered, i.e., class j.

As mentioned previously, two decisions need to be made when a request arrives. The first decision is whether to accept or reject the request. As seen in (4.6), this decision depends on the difference between $U_t^i(x)$ and $V_{t-1}(x-1) + F_i$. For $x > 0$, $t > 0$ and $i \geq 2$, let $h_t^i(x)$ describe this difference, i.e.,

$$\begin{aligned} h_t^i(x) &= U_t^i(x) - (V_{t-1}(x-1) + F_i) \\ &= \Delta V_{t-1}(x) + \max_{j\in A_i} g_t^i(x,j) - F_i. \end{aligned} \tag{4.12}$$

This can be interpreted as follows. A request for fare class i in decision period t is accepted if $h_t^i(x) \leq 0$ and otherwise rejected. In [28], You shows that $h_t^i(x)$ is non-increasing in x for a given t and i. Hence, as in Section 3.3.3 the acceptance/rejection decision can be reduced to a set of critical booking capacities $\hat{x}_i(t)$ such that $h_t^i(x) \leq 0$ for $x \geq \hat{x}_i(t)$ and $h_t^i(x) > 0$ for $x < \hat{x}_i(t)$. This is illustrated in Figure 4.2.

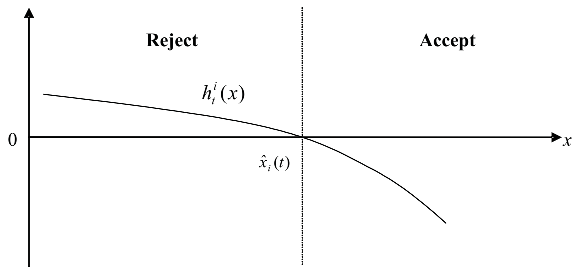


Figure 4.2: Illustration of Critical Booking Capacity.

If a request is rejected, it must be determined which class should be offered, i.e., the class j must be found. Recall that $U_i^t(x)$ is the total expected revenue generated with t decision periods and x seats remaining when a request for fare class i is rejected. Hence, when determining the class j offered to the rejected request, this is done by maximizing the expression for $U_i^t(x)$ given in (4.10). Thus, the class j is determined by

$$\max_{j \in A_i} g_t^i(x, j) = \max_{j \in A_i} \left\{ TU_{i,j} \Big(F_j - \Delta V_{t-1}(x) \Big) \right\}, \quad i \geq 2. \tag{4.13}$$

Now both decision rules for the acceptance/rejection process have been set up.

Calculations of the value function for all combinations of t and x are required when using the You method. Thus computation time may get prohibitive even though results are assumed to be good.

4.1.3 The B&P Solution Methods with Trade-Up

Two versions of the B&P method with trade-up are described in this section. The first method will in all of the following be denoted the adjusted B&P method with trade-up and the second method will be denoted the B&P LP method with trade-up. Either method can be used as an approximation method for the SIC problem with trade-up.

The Adjusted B&P Method with Trade-Up

The B&P solution method for the SIC problem without trade-up is easy to implement and is assumed to yield fairly good results fast. Hence, it is obvious to attempt using this method for the problem with trade-up as well. The expected demand used for the problem without trade-up can be adjusted such that trade-up is taken into account. This is done by for each class adding the expected number of passengers who will trade up from all lower-fare classes *to* this class and subtracting the expected number of passengers who will trade up *from* this class to all higher-fare classes. This yields the following adjusted demand for fare class i

$$D_i^{adjust} = D_i + \sum_{j=i+1}^{k} q_{j,i} \cdot D_j - \sum_{h=1}^{i-1} q_{i,h} \cdot D_i, \qquad i = 1, \ldots, k, \tag{4.14}$$

where D_i is the expected number of requests for fare class i and $q_{i,j}$ is the trade-up rate from class i to class j. There is no trade-up to class k and no trade-up from class 1.

The procedure and LP model when using this method is the same as described in Section 3.2.2 but the demand used in the method is the adjusted demand given by (4.14), where trade-up is taken into account. Thus, the method is used to approximate the value function suggested by Lee and Hersh described in Section 3.2, where the demand has been revised using (4.14). The value function $V_t(x)$ is approximated for all combinations of decision periods t and remaining capacities x with the optimal value of the objective function of the linear programming model, $\mathrm{LP}(x, t)$ given in (3.14) page 29 with adjusted demand. This LP is very easy to solve, an optimal solution is to fill in as much as possible of the demand for the most expensive class, then the demand for the second most expensive class, etc. This continues until the capacity of the aircraft has been reached.

Recall that $\Delta V_t^{LP}(x) = V_t^{LP}(x) - V_t^{LP}(x-1) = \mathrm{LP}(x,t) - \mathrm{LP}(x-1,t)$ and that the decision rule is to accept a request for fare class i if the following holds

$$F_i \geq \Delta V_{t-1}^{LP}(x), \qquad \text{for } t > 0, x \geq 0, \forall i.$$

As for the problem without trade-up this method is an approximation algorithm, which is expected to be fast but the results obtained are not expected to be as good as the results obtained when using the You solution method.

The B&P LP Method with Trade-Up

The adjusted B&P method is simple and it is easy to solve the LP, but it may not be a very good approximation method for the problem with trade-up. Another approach is to solve the following LP problem proposed by Algers, Andersson and Kohler in [1]:

$$LP(x,t) = \max \sum_{i=1}^{k} F_i \cdot y_i \tag{4.15}$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{i=1}^{k} y_i \le x \\ & y_k + r_k = D_k^t \\ & y_i + r_i = D_i^t + r_{i+1} \cdot TU_{i+1,i}, \qquad \text{for } i = 1,\ldots,k-1 \quad (4.16) \\ & y_i \ge 0,\ r_i \ge 0, \qquad \forall i. \end{aligned}$$

In this model y_i is the seat allocation for fare class i, r_i is the number of rejections of passengers requesting fare class i, D_i^t is the expected total demand for class i from time t to departure and $TU_{i+1,i}^t$ is the probability at time t that a rejected passenger requesting fare class $i+1$ is willing to trade up to fare class i and all higher-fare classes. The objective is to maximize total revenue and this is subject to the constraint that the remaining capacity cannot be exceeded.

The constraints in (4.16) have the sum of the number of allocated seats for fare class i and the number of rejected requests for fare class i on the left-hand side. This must be equal to the right-hand side, which is the demand for fare class i plus the proportion of rejected passengers in class $i+1$ that trades up to class i, i.e., the total demand for fare class i. It is only necessary to consider neighboring classes, since if it is optimal to close class i, too, then $y_i = 0$, and the whole right-hand side is rejected. Then a proportion of that demand, namely $TU_{i,i-1}$, may trade up to class $i-1$ and is added to D_{i-1}, etc. The decision variables in the model are y_i and r_i, $i = 1,\ldots,k$.

This method approximates the value function suggested by You described in Section 4.1.2. Therefore the decision rule (4.12) page 46 is used for the acceptance/rejection decisions and the offering decisions are made using (4.13) page 47.

4.1.4 The C,G&J Solution Method

In [11] an approximation algorithm for solving the SIC Problem *without* trade-up is proposed by Chen, Gunther and Johnson. The method will be denoted the C,G&J solution method and in this section the ideas in this method are applied to the model for the SIC problem *with* trade-up. In the algorithm upper and lower bounds for $\Delta V_t(x)$ are calculated. These bounds are then used to make the decision of whether an incoming request is accepted or rejected. A main idea is to use methods which calculate the bounds quickly, since it may be computationally difficult to determine the optimal value of $\Delta V_t(x)$.

In [11] it is suggested to use the optimal value of a deterministic LP to calculate the upper bound for $\Delta V_t(x)$ for the problem without trade-up. The value function $V_t(x)$ for the problem without trade-up is given in (3.9) page 25. The LP suggested in the paper is as given in (3.14) page 29. Let the optimal value of this LP be denoted $\hat{V}_t(x)$. In [11] it is proven that $\hat{V}_t(x)$ forms an upper bound for the value function (3.9). This fact is used to prove that $\Delta\hat{V}_t(x)$ is an upper bound for $\Delta V_t(x)$. Furthermore, a stochastic LP is proven to be a lower bound for the value function. Let this lower bound be denoted $\check{V}_t(x)$, then the paper proves that $\Delta\check{V}_t(x)$ is also a lower bound for $\Delta V_t(x)$. From these results the following algorithm for the acceptance/rejection of a class i request is obtained.

1. If $\Delta\check{V}_{t-1}(x) \geq F_i$ then reject the request. Otherwise go to step 2.
2. If $\Delta\hat{V}_{t-1}(x) \leq F_i$ then accept the request. Otherwise go to step 3.
3. Draw a random number r from the interval $[\Delta\check{V}_{t-1}(x), \Delta\hat{V}_{t-1}(x)]$.
4. If $r < F_i$ then accept the request. Otherwise reject the request.

The simplest way to apply these ideas to the SIC problem *with* trade-up, is to consider the LP given in (3.14) page 29. This LP is the same as the deterministic LP used in [11], and hence the results from the paper can be used directly. In the adjusted B&P method with trade-up, this LP is used with revised demand, such that trade-up is incorporated. If the same revised demand is used in the value function (3.9) page 25 for the L&H method, then the adjusted B&P method yields an upper bound for this value function. Therefore $\Delta\hat{V}_t(x)$ calculated with the adjusted B&P method with trade-up is also an upper bound for $\Delta V_t(x)$ calculated with the L&H method with revised demand. A lower bound can be obtained by revising the expected demand to incorporate trade-up in the stochastic model proposed in [11]. When the upper and lower bounds have been found the above algorithm can be used to determine whether a request should be accepted or rejected. It is not expected that the above procedure yields good results for the problem with trade-up, since trade-up is only incorporated in the demand and not explicitly in the model.

Bounds for $\Delta V_t(x)$ in the You method

The procedure just described is used for determining bounds for $\Delta V_t(x)$ calculated using the L&H method with revised demand. Another approach to the C,G&J method is to determine bounds for $\Delta V_t(x)$ calculated using the You method. This method is expected to give better results, since trade-up is explicitly incorporated in the model. To apply the above ideas to the You method both a deterministic LP and a stochastic LP for the problem with trade-up have to be set up.

Consider the deterministic LP given in (4.15) page 49, suggested by Algers, Andersson and Kohler in [1]. This deterministic LP explicitly incorporates trade-up in the model and is thus assumed to be an upper bound for the value function by You given in (4.7) page 45. Let the optimal value of this LP be denoted $\hat{V}_t(x)$. In this report the value of $\hat{V}_t(x)$ is not proven to be an upper bound for the value function $V_t(x)$ by You. Furthermore it will not be proven that $\Delta\hat{V}_t(x)$ is an upper bound for $\Delta V_t(x)$, but in Chapter 5 this will be tested with numerical experiments, before the C,G&J method is used for any final results.

In [11] it is suggested to use a stochastic LP model for the SIC problem without trade-up as a lower bound, hence this idea is applied here as well. A stochastic model for the SIC problem with trade-up is given by

$$\max \sum_{i=1}^{k} F_i \cdot \sum_{t=1}^{T} \mathbb{E}\left[D_i^t + N_{i+1,i}^t \mid y_i^t\right]$$

s.t.

$$\sum_{t=1}^{T}\sum_{i=1}^{k} y_i^t \le C \tag{4.17}$$

$$y_i^t + R_i^t - Z_i^t = D_i^t + N_{i+1,i}^t, \quad \forall i,t \tag{4.18}$$

$$y_i^t \ge 0, \qquad \forall i,t$$

where D_i^t is a random variable which describes the demand for class i at time t and R_i^t is a random variable describing the number of rejections in class i at time t. The variable Z_i^t is also random and is introduced to account for the variability in the demand D_i^t. The variable denotes the spill, i.e., the number of seats, which are allocated to class i, but which are not requested. The random variable $N_{i+1,i}^t$ is the number of passengers that trade up from class $i+1$ to class i at time t and it is a function of the random variables R_{i+1}^t and the trade-up rates $TU_{i+1,i}^t$. These trade-up rates are the probabilities at time t that a rejected customer for fare class $i+1$ is willing to trade up to fare class i and all higher-fare classes. $TU_{k+1,k}$ is defined to be zero. The term $\mathbb{E}\Big[D_i^t + N_{i+1,i}^t \mid y_i^t\Big]$ is the total expected number of requests in class i given seat allocation y_i^t. The decision variables in the model are the seat allocations y_i^t.

The right-hand side of (4.18) is a sum of the number of passengers who requests class i at time t and the number of passengers who trades up to class i and all higher-fare classes from class $i+1$. If the number of allocated seats for class i at time t is smaller than the total demand for class i at time t, then the left-hand side is a sum of the number of allocated seats and the number of rejected passengers, i.e., the variable Z_i^t is 0. If, on the other hand, the number of allocated seats for class i at time t is greater than the total demand, then the left-hand side is a sum of the number of allocated seats and the surplus amount of seats, Z_i^t. Then the number of rejected passengers for class i at time t is 0.

The constraint (4.17) ensures that the sum of the allocated capacity at all times and for all classes does not exceed the total capacity of the aircraft. Furthermore nonnegativity constraints apply to the decision variables.

Let the optimal value of the objective function for the stochastic LP be denoted $\check{V}_t(x)$. An intuitive explanation as to why the stochastic LP can be assumed to yield a lower bound for the value function by You will be given in the following. The optimal solution to the stochastic LP allocates each capacity unit to a fare class at a given time. If the capacity unit is not used at that time, it cannot be used later or for a different class, hence it remains empty. An optimal solution to the DP with trade-up can change the allocation as time progresses, hence, it is reasonable to assume, that this method yields better results than the stochastic LP with trade-up. In this report it is not proven that the optimal solution to the stochastic LP forms a lower bound for the value function, neither is it shown that $\Delta\check{V}_t(x)$ is a lower bound for $\Delta V_t(x)$.

The stochastic LP is not easy to solve and therefore a different approach for determining the lower bound for $\Delta V_t(x)$ is employed. A possibility is to use the upper bound calculated with the deterministic LP to form a lower bound as well. This can be done in a number of different ways. For instance by subtracting a positive number from the upper bound or by subtracting a certain percentage of the upper bound from the upper bound. Of course some research has to be made as to which method yields the best lower bound and obviously this does not necessarily need to be calculated as a function of the upper bound. Intuitively this seems to be the simplest solution, though.

When an upper bound and a lower bound have been determined, the algorithm given in the four items on page 50 can be used. Notice, though, that when the bounds are determined for $\Delta V_t(x)$ calculated using the You method, the decision rules in the algorithm must be replaced by the decision rules used in the You method given in (4.12) page 46.

Calculation of the Bounds

In order to determine whether a request should be accepted or rejected, it is necessary to calculate $\Delta\check{V}_t(x)$ and $\Delta\hat{V}_t(x)$. Hence, $\check{V}_t(x)$, $\check{V}_t(x-1)$, $\hat{V}_t(x)$ and $\hat{V}_t(x-1)$ must be calculated each time a request arrives. Since decisions have to be made quickly when requests arrive, it is not feasible to calculate these four values each time a decision has to be made. Furthermore there is a large amount of remaining capacities and decision periods, hence calculating bounds for all combinations of decision periods and remaining capacity will be very time comsuming, even when this can be done before the beginning of the booking period. Instead the bounds are only calculated at some predetermined values of the remaining capacity and then the bounds for all values of remaining capacities are estimated using splines between the calculated values of the bounds. Thus the values of remaining capacity for which the bounds are calculated are used as knots in cubic splines. There is one spline for the upper bound and one for the lower bound and this way estimations of the bounds for all remaining capacities are obtained. To estimate the values of the bounds in the entire time space, these bounds are calculated in the beginning of each data interval and then the bounds for all decision periods between these values are estimated using linear interpolation. Hence, this way the bounds are estimated for all combinations of decision periods and remaining capacity.

Recall that data interval 1 is nearest departure. At time j the splines used for estimating the bounds between times j and $j-1$ are activated, where time j is the beginning of a data interval and time $j-1$ is the beginning of the next data interval, see Figure 4.3. Consider a request for fare class i, which arrives at time τ with a remaining capacity of x. Recall that the decision rule for a request arriving at time t requires using the value $\Delta V_{t-1}(x)$. Therefore the data interval containing the time $\tau-1$ is considered and it is assumed that $\tau-1 \in [j, j-1]$, see Figure 4.3. If the capacity will not be exceeded by accepting the request, then the upper and lower bounds for the value function, $\hat{V}_{\tau-1}(x)$, $\hat{V}_{\tau-1}(x-1)$, $\check{V}_{\tau-1}(x)$ and $\check{V}_{\tau-1}(x-1)$, must be determined. These are calculated using linear interpolation between times j and $j-1$, i.e., linear interpolation is used to find the values in $t=\tau-1$ for the remaining capacities x and $x-1$. When using linear interpolation to find the values of the bounds in $t=\tau-1$ it is assumed that demand is equally distributed in a given data interval.

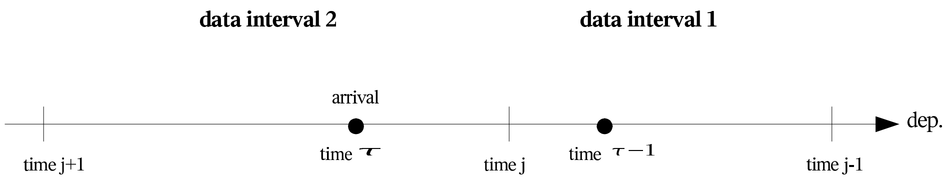


Figure 4.3: Illustration of Activation of Bounds.

Once the values of the bounds have been found, the procedure of determining whether to accept or reject the request is as described in the four items on page 50.

4.1.5 The C&H Solution Method

In this section the last method for solving the general SIC problem with trade-up is described. This method is an approximation algorithm and it is developed by using the requirements suggested by Cooper and Homem-de-Mello in [14] for the SIC problem *without* trade-up. The approach is to use a simple heuristic in the beginning of the booking period and then switch to an exact method closer to departure. A method like this, which is a combination of several solution methods is called a hybrid method. The article refers to [15] by Gallego and van Ryzin and [13] by Cooper, which show that approximation methods can yield good results, when the remaining capacity and expected demand are large, i.e., far from departure, and there is a reference to [24] by Subramanian et al., where it has been suggested that close to departure it is important to use an exact model.

For this problem it is attempted to use the adjusted B&P method with trade-up described in Section 4.1.3 as an approximation method in the first part of the booking period and then switch to the You solution method in the last part of the booking period closest to departure. Theoretically, the B&P LP method with trade-up, which is also described in Section 4.1.3, can be used as the approximation method as well. This method is extremely slow, though, since in this report the LP is solved using Matlab's predefined optimizer `linprog.m`. Using this method as it is now is not feasible for the hybrid methods, since a basic idea for these methods is that the approximation method must be fast. Instead, another approach is to use the You method with a large ϵ in the beginning of the booking period and then switch to the You method with a small ϵ closer to departure. This can be done, since the You method with a large ϵ is faster and less accurate.

It is simple to just use the two methods separately before and after the switch. What comlicates the method is that after the switch, when solving the problem using the exact method, the already accepted passengers from the first part of the booking period need to be taken into account. In this report three different ways of making the switch are suggested. Common for all three methods is that the exact method is used first even though this is in the last part of the booking period, and then the results obtained are used when solving the problem for the first part of the booking period with an approximation method. It makes sense to solve the part of the problem closest to departure first, since this is a common procedure when using DP which is used for the last part of the booking period and in some cases for the first part of the booking period as well.

The three different hybrid methods are described in the following subsections. In these sections the time of the switch will be denoted w.

"Max"

The first solution method is denoted "Max". The exact method is applied in the last part of the booking period in decision periods $1, \ldots, w$, with the demand adjusted to be only for these decision periods. This yields the values $V_t^e(x)$ for $t = 1, \ldots, w$ and $x = 1, \ldots, C$. Then the approximate method is used to solve the problem in the first part of the booking period, in decision periods $w+1, \ldots, T$ with the demand adjusted to be only for these decision periods. This yields the values $V_t^a(x)$ for $t = w+1, \ldots, T$ and all values of x.

The last decision period in which the exact method is applied is w. Then $V_w^e(x)$ for $x = 1, \ldots, C$ is the maximum expected revenue, which can be obtained from decision period w to departure, i.e., in decision periods $1, \ldots, w$, for each remaining capacity. Decision period T is the decision period farthest from departure, and then $V_T^a(x)$ for $x = 1, \ldots, C$ is the maximum expected revenue, which can be obtained in decision periods $w+1, \ldots, T$ for each remaining capacity. If y seats are sold in the first part of the booking period, then only $C - y$ seats can be sold in the last part of the booking period. To determine the number of seats to sell in each part of the booking period such that total expected revenue is maximized, the following value of y is determined

$$\max_y \left\{ V_w^e(z) + V_T^a(y) \right\}, \qquad z = 1, \ldots, C, \quad y = C - z.$$

The optimal value of y is used in the first part of the booking period, i.e., in the part of the booking period farthest from departure. It is used such that after y seats have been sold in decision periods $T, \ldots, w+1$ only the classes that are open at time w, when $C-y$ seats have been sold, should be accepted until decision period w. If y seats are sold in the first part of the booking period, then there are $C-y$ seats available for sales in the last part of the booking period, and this way, the expected revenue for the entire booking period is maximized.

"Add Column"

The second solution method is denoted "Add Column". Again the exact method is used in decision periods $1, \ldots, w$. This yields the values $V_t^e(x)$, for $t = 1, \ldots, w$ and $x = 1 \ldots, C$. Then the approximate method is used to solve the problem in decision periods $w+1, \ldots, T$. This yields the values $V_t^a(x)$ for $t = w+1, \ldots, T$ and all x. The value $V_w^e(x)$ is the expected revenue, which can be obtained in decision periods $1, \ldots, w$ with a remaining capacity of x and the value $V_{w+1}^a(x)$ is the expected revenue, which can be obtained in decision period $w+1$ with a remaining capacity of x. In $V_{w+1}^a(x)$ the revenue which can be obtained in decision periods $1, \ldots, w$ has not been accounted for. Hence, to do this, $V_w^e(x)$ is added to $V_{w+1}^a(x)$ to yield the expected revenue with a remaining capacity of x for decision periods $1, \ldots, w+1$. This is done for all x. Similarly in the value $V_{w+2}^a(x)$ the decision periods $1, \ldots, w$ have not been taken into account, thus, $V_w^e(x)$ must be added to $V_{w+2}^a(x)$ as well for all x. Hence, $V_w^e(x)$ must be added to all $V_t^a(x)$ for $t = w+1, \ldots, T$ and for all x. Then the value function is available for all values of t and x and can thus be used to determine which classes should be open in which decision periods.

"Combi"

The third hybrid solution method is "Combi" and, as in the two first methods, in this method the exact solution method is used first for decision periods $1, \ldots, w$. This yields the values $V_t^e(x)$ for $t = 1, \ldots, w$ and all x. Then the approximation method is used for the *entire* booking period and with the *entire* demand, yielding the values $V_t^a(x)$ for $t = 1, \ldots, T$ and all x. Now the only values of $V_t^a(x)$ which are needed are the values for decision periods $t = w+1, \ldots, T$ and all x. Hence, values of $V_t(x)$ for the entire booking period are $V_t^e(x)$ for $t = 1, \ldots, w$ and $V_t^a(x)$ for $t = w+1, \ldots, T$. These values are then used to determine which classes should be open in the decision periods in the entire booking period. It is feasible to calculate $V_t^a(x)$ for $t = 1, \ldots, T$ and all x because the approximation method is assumed to be so fast that the running time for the hybrid method is still much shorter than the running time for the exact method, for instance the You method with a small ϵ for the entire booking period.

4.2 Simplified SIC with Trade-Up

In this section models and solution methods for the SIC problem with trade-up and the simplifying assumption will be set up and described. Recall, the simplifying assumption is that there are no differences in the buying conditions for different trade-up classes. Thus, all demand is for the least expensive class in the trade-up market and demand for all other classes is only realized due to trade-up from this class.

The solution methods in this section will primarily be simplifications of the methods described in Section 4.1. However, deriving the EMSRb method with the simplifying assumption makes no sense, since the variable definitions would have to be changed completely and the decision tree in Figure 4.1 page 42 would only have a single branch. Hence, the EMSRb method for the simplified problem will not be derived. Instead another DP model will be set up, which takes multiple requests in each decision period into account.

In Section 4.2.1 the simplified You solution method where there is only one arrival in each decision period is described and in Section 4.2.2 a DP model is introduced in which several arrivals in each decision period are allowed. This solution method is called the HM method. The B&P methods, the C,G&J methods and the C&H methods are available for the simplified problem as well, and these are described in the sections which follow.

4.2.1 The Simplified You Solution Method

In this section the dynamic programming model described in Section 4.1.2 is simplified to fit the assumption of a trade-up market as exists at BA. Recall, the value function for the general DP model in the You solution method is as follows

$$V_t(x) = \begin{cases} \left(1 - \sum_{i=1}^{k} P_i^t\right) V_{t-1}(x) + P_1^t\left(F_1 + V_{t-1}(x-1)\right) & \\ + \sum_{i=2}^{k} P_i^t \, \max\left(U_t^i(x), F_i + V_{t-1}(x-1)\right) & \text{for } x > 0,\, t > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.19}$$

where

$$U_t^i(x) = \max_{j \in A_i} \left\{ TU_{i,j}\left(F_i + V_{t-1}(x-1)\right) + \left(1 - TU_{i,j}\right) V_{t-1}(x) \right\},$$

and j is the class which is offered if a request for class i is rejected.

Since it is assumed that passengers always request the lowest-fare class in the market, class k, the expression for $U_t^i(x)$ can be simplified to the following

$$U_t^k(x) = \max_{j=1,2,\dots,k-1} \Big\{ TU_{k,j}\big(F_j + V_{t-1}(x-1)\big) + \big(1 - TU_{k,j}\big)V_{t-1}(x) \Big\},$$

where $TU_{k,j}$ is the trade-up rate from class k to classes $1, \dots, j$. Furthermore, for $x > 0$ and $t > 0$, $V_t(x)$ can be simplified to

$$V_t(x) = \big(1 - P_k^t\big)V_{t-1}(x) + P_k^t \max\Big(U_t^k(x), F_k + V_{t-1}(x-1)\Big). \qquad (4.20)$$

The decision rule is thus to accept a request if the following holds

$$\max_{j=1,2,\dots,k-1} \Big\{ TU_{k,j}\big(F_j + V_{t-1}(x-1)\big) + \big(1 - TU_{k,j}\big)V_{t-1}(x) \Big\} \leq F_k + V_{t-1}(x-1)$$

which, in terms of $\Delta V_t(x)$, can be rewritten as

$$\max_{j=1,2,\dots,k-1} \Big\{ \big(F_j - \Delta V_{t-1}(x)\big)TU_{k,j} \Big\} + \Delta V_{t-1}(x) \leq F_k \qquad (4.21)$$

Each time a request arrives, two decisions have to be made. The first decision is whether or not class k should be open for sales and if not, which class is the lowest open class j.

The decision of whether or not to accept a class k request depends on (4.21), i.e., a request is accepted if the inequality is satisfied and otherwise it is rejected. When determining the value on the left-hand side, the value of j, which is the optimal class to offer if class k is not open, is obtained as well.

When using the simplified You solution method calculations of the value function for all combinations of remaining capacity and decision periods are required. To calculate the value of $V_t(x)$, the request probabilities P_k^t for fare class k for all decision periods t have to be determined. Furthermore the trade-up probabilities from fare class k to class i and all higher-fare classes, $TU_{k,i}$, need to be known.

The request probabilities P_k^t are determined in a similar way as in Section 3.2.1 page 26. The difference is that in this section it is assumed that all demand is for class k. Hence, the expected demand in data interval d, μ^d, is only for class k and therefore the request probability has the following simple form

$$P_k^t = (\mu^d/\nu^d)\exp(-\mu^d/\nu^d),$$

where ν^d is the number of decision periods in data interval d. The only data needed are μ^d, and these are provided by British Airways. Now $V_t(x)$ and thereby $\Delta V_t(x)$ can be calculated for all combinations of t and x, thus all the variables needed to make the acceptance/rejection and the offering decision are known.

The model and decision rule derived in this section can be derived from both of the papers [25] by Talluri and van Ryzin and [28] by You.

4.2.2 The HM Solution Method

In the model described in the previous section, it is assumed that only one request in each decision period occurs. To make sure this is satisfied, the booking period needs to be divided into a large number of decision periods. Hence, when calculating the value function for all combinations of decision periods and remaining capacity, the number of calculations are comprehensive. In this section a model is set up where multiple requests may arrive in each decision period. Therefore the number of decision periods, which the booking period needs to be divided into can be reduced considerably compared to the case where there is only one request per decision period. This model is suggested by [16]. It is assumed that all demand is for class k and demand for other classes is only realized due to trade-up, i.e.,

$$D_i^t = TU_{k,i} \cdot D_k^t, \tag{4.22}$$

where D_i^t is the expected demand for fare class i in decision period t and $TU_{k,i}$ is the trade-up rate from class k to classes $1, \ldots, i$. Thus, the demand is nested such that D_i^t is the demand for class i and all higher-fare classes, i.e.,

$$D_1^t \leq D_2^t \leq \cdots \leq D_{k-1}^t \leq D_k^t.$$

In the beginning of each decision period it has to be determined which classes are open. Since it is assumed that when rejected passengers choose to trade up this will always be to the lowest open class, it is only necessary to determine this class j.

The value function for this model is as follows

$$\begin{aligned} V_t(x) = \max_{j=1,\ldots,k} \Big[& P(D_j^t = 0) \cdot \big(0 \cdot F_j + V_{t-1}(x)\big) \\ & + P(D_j^t = 1) \cdot \big(1 \cdot F_j + V_{t-1}(x-1)\big) \\ & + P(D_j^t = 2) \cdot \big(2 \cdot F_j + V_{t-1}(x-2)\big) \\ & + \cdots \\ & + P(D_j^t \geq x) \cdot \big(x \cdot F_j + V_{t-1}(0)\big) \Big], \qquad x > 0, t > 0, \end{aligned} \tag{4.23}$$

where $P(D_j^t = s)$ is the probability that exactly s requests for class j occur in decision period t, and j is the lowest open class, i.e., the decision variable.

As can be seen from (4.23), when several arrivals can occur in each decision period, there are multiple different arrival scenarios in each decision period. If no arrivals occur the gain in revenue is zero and the remaining capacity is unchanged in the following decision period. If g requests for class j occur, where $g = 1, \dots, x$, the gain in revenue is g times the fare for class j and the remaining capacity in the following decision period is $x - g$. Since overbooking is not considered, the airline can sell at most C seats on the aircraft. Therefore, even if the number of arrivals in a decision period is larger than the remaining capacity, the gain in revenue is only x times the fare for class j. Thus, under this scenario the remaining capacity in the following decision period is zero. By calculating $V_t(x)$ in (4.23) the lowest open class j is determined for all values of remaining capacities and decision periods.

Now that several bookings in a decision period are allowed, the number of decision periods in each data interval is much smaller than for the You model. Thus, even if the value function $V_t(x)$ has to be calculated for all combinations of remaining capacity and decision periods, the number of function evaluations is smaller than for the You method, i.e., the computation time is assumed to be shorter than that of the You method.

4.2.3 The Simplified B&P Methods with Trade-Up

In this section two approximation methods for the simplified SIC problem with trade-up are described. These are the simplified adjusted B&P method with trade-up and the simplified B&P LP method with trade-up.

The Simplified Adjusted B&P Method with Trade-Up

The LP used in the B&P method without trade-up and the adjusted B&P method for the general problem with trade-up can be used with the assumptions made by BA as well. This is done by determining the expected demand to come for each fare class from the trade-up rates, i.e.,

$$D_i^t = q_{k,i} \cdot D_k^t, \qquad i = 1, \dots, k-1. \tag{4.24}$$

where D_k^t is the expected demand for class k in decision period t and $q_{k,i}$ is the trade-up rate from class k to class i.

As for the adjusted B&P method with trade-up for the general SIC problem, the procedure and LP when using the simplified adjusted B&P method is as described in Section 3.2.2. This method is used to approximate the value function by Lee and Hersh from Section 3.2 where the simplifying assumption is applied and the demand has been calculated using (4.24). Once all values of the value function have been approximated, $\Delta V_t^{LP}(x)$ can be determined as well. This is used to decide whether a request should be accepted or rejected by using the decision rule (3.11) page 26.

The Simplified B&P LP Method with Trade-Up

As for the general problem with trade-up, an alternative to solving an LP where the demand for each class is determined using the trade-up rates, is to solve an LP problem similar to (4.15) page 49. The following LP is used for the simplified problem

$$\begin{aligned} LP(x,t) = \max & \sum_{i=1}^{k} F_i \cdot y_i & & (4.25) \\ \text{s.t.} \quad & \sum_{i=1}^{k} y_i \leq x & & \\ & y_k + r_k = D_k & & (4.26) \\ & y_i + r_i - TU_{k,i} \cdot r_k = 0, \qquad i = 1, 2, \ldots, k-1 & & (4.27) \\ & y_i \geq 0,\ r_i \geq 0, \qquad \forall i. & & \end{aligned}$$

The constraint (4.26) expresses that the sum of the allocated seats for class k and the number of rejected requests for class k must be equal to the total demand. Constraint (4.27) ensures that the number of seats allocated for class i plus the number of rejections in that class is equal to the number of passengers who are rejected in class k and willing to trade up to class i or higher-fare classes. The latter corresponds to the demand for class i.

The optimal value of the LP is used to approximate the value function $V_t(x)$ in the simplified You model. Once the value function has been approximated, $\Delta V_t(x)$ can be found. The decision rule for determining whether an incoming request should be accepted or rejected is given as for the simplified You method, see (4.21) page 58.

4.2.4 The Simplified C,G&J Solution Method

In this section the simplification of the C,G&J solution method from Section 4.1.4 will be described. Again upper and lower bounds for $\Delta V_t(x)$ need to be set up. In this case the value function for the simplified SIC problem with trade-up is given by (4.20) page 58.

Denote the optimal value of the deterministic LP given in (4.25) by $\hat{V}_t(x)$. As for the general problem, since trade-up is explicitly incorporated in this LP, $\hat{V}_t(x)$ is assumed to be an upper bound for the value function $V_t(x)$ given in (4.20). Furthermore $\Delta\hat{V}_t(x)$ is assumed to be an upper bound for $\Delta V_t(x)$. Neither of these statements are proven in this report.

Now the stochastic LP, which is assumed to form a lower bound for $\Delta V_t(x)$ in the simplified problem must be determined as well. A stochastic model for the simplified problem is as follows

$$\max \sum_{t=1}^{T} \left[F_k \cdot \mathbb{E}\left[D_k^t \mid y_k^t \right] + \sum_{i=1}^{k-1} F_i \cdot \mathbb{E}\left[N_{k,i}^t \mid y_i^t \right] \right]$$

s.t.

$$\sum_{t=1}^{T} \sum_{i=1}^{k} y_i^t \leq C \tag{4.28}$$

$$y_k^t + R_k^t - Z_k^t = D_k^t, \quad \forall t \tag{4.29}$$

$$y_i^t + R_i^t - Z_i^t = N_{k,i}^t, \quad i = 1, \ldots, k-1 \ \forall t \tag{4.30}$$

$$y_i^t \geq 0, \qquad \forall i, t$$

The variables are as explained in Section 4.1.4. The kth term is separated from terms $1, \ldots, k-1$ in the objective function and equations (4.30) and (4.29) substitute (4.18) page 51, since there is only demand for class k and demand for all other classes is only due to trade-up from class k. The simplified stochastic LP is a special case of the general stochastic LP in Section 4.1.4. Since the stochastic LP is simplified by the same assumptions as the DP by You, then if the general stochastic LP is a lower bound for the general value function, then the simplified stochastic LP is also assumed to be a lower bound for the simplified value function. Hence, it is assumed to form a lower bound for $\Delta V_t(x)$ as well. As for the general problem this stochastic LP is difficult to solve. Hence, in this report the lower bound is determined as a function of the upper bound for the simplified problem as well.

Again, the LP (3.14) page 29 where the expected demand for each fare class has been determined using (4.24) page 60 is an upper bound for $\Delta V_t(x)$ calculated with the L&H method where the expected demand is calculated in the same way. This can also be used as a C,G&J method.

The subsequent procedure when finding a lower bound for the value function and determining when to accept and reject passenger requests is the exact same as for the general C,G&J method described in Section 4.1.4.

4.2.5 The Simplified C&H Solution Method

In this section the ideas from the approximation algorithm suggested by Cooper and Homem-de-Mello in [14] will be applied to the simplified SIC problem with trade-up. As for the general problem, a simple heuristic is used in the beginning of the booking period and then a switch to a more exact method in the end of the booking period is made.

For the approximation method the B&P method as well as the You method with a large ϵ is used. Again, if the LP (4.25) page 61 can be solved quickly, it would also be feasible to use the optimal value of this as an approximation of the value function in the first part of the booking period, but since Matlab's `linprog.m` is used, this method is too slow. For the simplified problem two different solution methods are available as the exact method. These are the You solution method with a small ϵ and the HM solution method. Hence, for the simplified problem different combinations of methods are possible.

Apart from which methods are used as approximate and exact methods, as for the general problem three different hybrid methods are suggested in this report. Again these are the "Max" hybrid method, the "AddColumn" hybrid method and the "Combi" hybrid method. These methods are exactly as explained for the general problem in Section 4.1.5, hence these will not be described further in this section.

4.3 Implementation

In this section the implementations of the methods explained in Sections 4.1 and 4.2 are described. In the programs for these methods it is chosen not to transform the lowest open class or the outputs of the functions into booking limits as was done for the SIC problem without trade-up. The methods have different outputs and therefore for the general SIC problem with trade-up a different program calculates the revenue of the simulation process for each method. For the SIC problem with trade-up there are many methods. Hence, the B&P and the You solution methods with update are not implemented, as was done for the problem without trade-up.

4.3.1 General SIC with Trade-Up

The programs described in this section are enclosed on the CD in the folder *General SIC with TU/Main Functions*

In Table C.3, Appendix C, a list of the main functions described in this section can be seen. Furthermore, in this table the different inputs and outputs used by the programs are given. The description of the variables which are different from those given in Table C.2, Appendix C, can be seen in Table C.5, Appendix C.

Simulation

The simulation of the booking process for the general SIC problem with trade-up is implemented in the program `SimulationTradeup.m`. This program is very similar to the program `Simulation.m` where trade-up is not incorporated, see Section 3.3.1 for description. The only difference between these two programs is that `SimulationTradeup.m` has an additional output variable representing the simulated arrival's willingness to trade up. For each arrival this probability is generated as a random number between zero and one and these numbers are stored in the vector $TUarr$.

EMSRb with Trade-Up

The decision rule regarding the booking limits for the fare classes is the only difference between the EMSRb method with trade-up and the EMSRb method without trade-up. Thus, the programs for the two methods do not differ much either. Therefore only the difference in the implementation of the decision rule and the extensions regarding trade-up will be described in this section. For a description of the implementation of the basic calculations in the EMSRb method, see Section 3.1.2.

The EMSRb method with trade-up is implemented in the Matlab program `EMSRbTUdecPerTU.m`. To calculate the protection levels with the decision rule given in (4.5) page 44, it is necessary to have the trade-up rates $TU_{i+1,i}$ for $i = 1, \ldots, k-1$. The trade-up rates given by the airline are the probabilities, that a passenger is willing to trade up from the lowest class in the market, class k, to class i and all more expensive classes $1, 2, \ldots, i-1$, i.e., $TU_{k,i}$ for $i = 1, \ldots, k-1$. The trade-up rates $TU_{i+1,i}$ are calculated by using (4.1) page 41.

These trade-up rates $TU_{i+1,i}$ are calculated with the auxiliary function `calcTU.m`. An $r \times k$ matrix containing these trade-up rates between all neighboring classes for all data intervals is given as input to `EMSRbTUdecPerTU.m`, where k is the number of classes and r is the number of data intervals. The seat protection levels can then be calculated using the decision rule given in (4.5) page 44.

The procedure if a passenger is accepted is exactly the same as in the implementation of the EMSRb heuristic without trade-up, but the procedure following the rejection of a request is different. In the EMSRb method with trade-up a rejected passenger is not necessarily lost revenue, as it was for the EMSRb method without trade-up, since the passenger might trade up to another class. This is incorporated by first determining which classes are open, since the requested class was closed. The open classes are determined by for each class comparing the sum of accepted requests for that specific class and all lower-fare classes with the booking limit for that class. For instance the booking limit for fare class 4 is compared with the sum of the accepted requests for classes $4, \ldots, k$. If the booking limit is greater than the sum of the accepted requests, then a request for class 4 can be accepted otherwise it is rejected. If no class satifies that the booking limit is greater than the cumulated number of accepted requests and if the remaining capacity is greater than zero, then class 1 is the only open class.

After the open classes have been found the next step is to determine if the customer will trade up to the lowest open class. Therefore, the total probability for trade-up from the requested class is calculated. This is given by the sum of the trade-up probabilities from the requested class to all open classes. To calculate this sum, the strict trade-up rates between any two classes i and h, $q_{i,h}$, must be calculated using (4.2) page 41. The calculations of these trade-up rates $q_{i,h}$ are implemented in the auxiliary function `calcDtot.m` and a $k \times k \times r$ matrix containing these trade-up rates between all classes for all data intervals is obtained. An example of how it is determined whether or not a rejected passenger will trade up is given in the following. If a rejected request is for fare class 4 and the open classes are all higher-fare classes, then the total trade-up probability from class 4 is

$$TotalQ = q_{4,1} + q_{4,2} + q_{4,3}.$$

This trade-up probability is compared with the element in the input vector $TUarr$, which corresponds to the rejected request. Recall, that $TUarr$ contains random numbers between 0 and 1 generated in the simulation as mentioned in the previous subsection. The passenger is willing to trade up to the lowest open class if the random number in $TUarr$ is smaller than the total trade-up probability, otherwise the customer gets a final rejection and is hence lost revenue. This process is repeated for all the simulated requests, such that a total revenue from the simulation is obtained and given as output from the program.

The You Solution Method

The You solution method for the general SIC problem with trade-up is implemented in the program `You.m`. In the model described in Section 4.1.2, it is assumed that the requested class is known before the offering decision is made. In real life this is not the case, though, since the classes must be offered quickly after a request has been rejected. Therefore all calculations for this method must be made for all decision periods, all remaining capacities and all possible requested classes, since the offering decision depends on which class is requested.

In the implementation of the You method, initially the class which is offered if a request is rejected is determined for all classes, such that no matter which class is requested it is known what class to offer. The class offered, $j_t^i(x)$, is determined using the equation given in (4.13) page 47 and stored in a $T \times C \times k$ matrix, where T is the number of decision periods, C is the total capacity of the aircraft and k is the number of classes. After the set of classes to offer is determined, this can be used to calculate $g_t^i(x, j)$ given in (4.11) page 46. The trade-up rates $TU_{i,j}$ are calculated using the program `calcTU.m`. Then everything needed for calculating the function $h_t^i(x)$ has been determined, hence this is calculated using (4.12) page 46 and stored in a $T \times C \times k$ matrix, where T, C and k are given as above. Recall that the function $h_t^i(x)$ is used to determine whether to accept or reject a request. The outputs from the function are both of these three-dimensional matrices which are used in the acceptance/rejection and offering decision. Furthermore the value function $V_t(x)$ is given as output.

The program `RevenueYT.m` is used to calculate the revenue from the simulation process when using the output from `You.m`. Whether to accept or reject a request is considered as the bookings occur. A request for fare class 1 is always accepted if the remaining capacity is greater than zero. A request for fare class i in decision period t with a remaining capacity x is accepted or rejected by looking at the (t, x, i)th element in the matrix containing the values of $h_t^i(x)$. If the value of this element is less than zero, then the request is accepted, due to the acceptance rule described on page 46. If the element is greater than zero, then the request is rejected and the offering decision is made in a similar way as with the EMSRb method with trade-up. Thus, first the lowest open class is determined by looking at the (t, x, i)th element in the matrix containing the values of $j_t^i(x)$, where t, x and i are given as above. After the class offered has been found it must be determined whether the rejected passenger is willing to trade up to the lowest open class or not. This is done in the exact same way as in the program `EMSRbTUdecPerTU.m`. The output from `RevenueYT.m` is the total revenue obtained from the simulation when using the You solution method.

The Adjusted B&P Method with Trade-Up

The adjusted B&P method with trade-up is implemented in `PopBLTU.m`. The only differences between this program and `PopBL.m` are the demand which is used and the output from the program. The adjusted demand incorporating trade-up is given as input to `PopBLTU.m` and this demand is calculated with the auxiliary function `calcDtot.m`. As described for the EMSRb method with trade-up, the matrix containing the trade-up probabilities between all classes i and j, $q_{j,i}$, is also calculated by this program. Hence, all the variables in equation (4.14) page 48 are known and the adjusted demand can be calculated. This is stored in a $r \times k$ matrix, where r is the number of data intervals and k is the number of classes. The calculation of the approximation of the value function is done in the same way as for the B&P method without trade-up. This value function is used to calculate the lowest open classes for all remaining capacities and decision periods. Therefore the output from the program `PopBLTU.m` is a $C \times T$ matrix containing the lowest open classes instead of booking limits which was the output from `PopBL.m`. Furthermore a $C \times T$ matrix containing the approximate values of the value function $V_t(x)$ is given as output.

The program `matrixRevenue.m` is used to calculate the revenue from the simulation process. This function calculates the revenue by using a matrix containing the lowest open class for each remaining capacity and each decision period. Hence, when it is decided whether to accept or reject a request for fare class i in decision period t with a remaining capacity x, then the (x, t)th element in the matrix is compared with the requested class. If the class is a higher-fare class than the lowest open class, then the request is accepted otherwise it is rejected and the procedure regarding whether or not the passenger will trade up is carried out. This procedure is the same as for the previous two methods.

The B&P LP Method with Trade-Up

The B&P LP method with trade-up is implemented in the program `BPLPTU.m`. Initially, the expected demand to-come for each data interval and each fare class is calculated and stored in a $r \times k$ matrix, where r is the number of data intervals and k is the number of fare classes. Then all the variables in the LP model are determined and the LP can be solved using the predefined Matlab function `linprog.m`. The function `linprog.m` has seven inputs

$f, A, b, Aeq, beq, LB, UB$ and it attempts to solve the LP problem

$$\begin{aligned} &\min \mathbf{f}^T\mathbf{z} \\ \text{s.t} \quad &\mathbf{Az} \le b \\ &\mathbf{Aeq} \cdot \mathbf{z} = \mathbf{beq} \\ &\mathbf{LB} \le \mathbf{z} \le \mathbf{UB}. \end{aligned}$$

Therefore to use this function to solve the LP, the input matrices must be set up. For instance, for a problem with three classes for each time t the matrices for the LP model described on page 49 are given by

$$\mathbf{f} = \begin{bmatrix} -F_1 \\ -F_2 \\ -F_3 \end{bmatrix}, \qquad \mathbf{z} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix}, \qquad \mathbf{A}^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{beq} = \begin{bmatrix} D_1^t \\ D_2^t \\ D_3^t \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{Aeq} = \begin{bmatrix} 1 & 0 & 0 & 1 & -TU_{2,1}^t & 0 \\ 0 & 1 & 0 & 0 & 1 & -TU_{3,2}^t \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and b is the remaining capacity. Hence, b runs through all values $1, 2, \ldots, C$. Notice, there is a minus in front of the fares in $\mathbf{f}$, since `linprog.m` minimizes and the LP to be solved is a maximization problem.

The LP is solved for each data interval given in an *update* vector, which is input to the function. It is chosen only to calculate the approximations with this method in the beginning of each data interval. This is due to the slowness of the optimizer `linprog.m` in Matlab. For each optimization of the LP for a specific time and all remaining capacities, Matlab takes about 2.30 seconds, which is too slow to be feasible for the airline. This has to be done for each time in the *update* vector, hence by only updating the booking limits calculated with this method in the beginning of each data interval instead of in each decision period, running time is reduced significantly. To determine how fast the LP can be solved, if another optimizer is used, the LP is implemented in GAMS as well. Here the optimization for each time and all remaining capacities takes approximately 0.27 seconds. Thus, by using a more efficient optimizer than `linprog.m` the time can be reduced significantly, for instance by using GAMS the optimization time is reduced by a factor 10.

After the LP's have been solved for all remaining capacities and times in the *update* vector, the approximations of $\Delta V_t(x)$ can be calculated. These are then used to determine the lowest open class for all values of the remaining capacity and for each time in the *update* vector. This is done as described in Section 4.1.2. This class is then stored in a $Lup \times C \times k$ matrix, where Lup is the length of the *update* vector, C is the capacity and k is the number of fare classes. This matrix is transformed such that a row is generated for each decision period instead of only for each element in *update*. Then the revised $T \times C \times k$ matrix is given as output from the program, where T is the number of decision periods. Once the lowest open class has been determined for all remaining capacities and all times in the *update* vector, this is used to calculate $g_t^i(x, j)$ given in (4.11) page 46. Then finally $h_t^i(x)$ is calculated using (4.12) page 46 and stored in a $Lup \times C \times k$ matrix. This matrix is revised in the same way as the matrix of lowest open classes and then also given as output from the program.

The program `RevenueYT.m` is used to calculate the revenue from the simulation process. For a description of this function see the description of the implementation for the You solution method.

The C,G&J Solution Method

The C,G&J method is implemented in two different programs depending on which of the two B&P methods described in the two previous sections that is used to calculate the upper bound for $\Delta V_t(x)$. As described in Section 4.1.4 the upper bound can be calculated by the adjusted B&P method or by the B&P LP method, where the LPs are solved using Matlab's `linprog.m`. When using the adjusted B&P method the C,G&J method is implemented in the program `BPTUSpline.m` and when using the B&P LP method, it is implemented in the program `BPLPSplineNew.m`. The programs are modified versions of `PopBLTU.m` and `BPLPTU.m` described in the previous section. The modification is regarding the number of capacities and times, where the approximation values are calculated. Hence, the programs `BPTUSpline.m` and `BPLPSplineNew.m` need two vectors as input containing the times and capacities in which the approximation values are to be calculated. Both programs calculate the approximation to the value function for the times and capacities given in the input vectors and give a $LC \times LT$ matrix containing these values as output, where LC is the number of capacities and LT is the number of times in which a value is calculated.

After the bounds for the value function have been calculated for the capacities given in the input vector, splines are used to obtain the upper bounds for all capacities. This is done in the auxiliary program `findBounds.m`, which calculates both the upper and the lower bounds for all capacities but only for some specified times. The predefined Matlab function `spline.m` is used to calculate the splines. This function has three input arguments X, Y, XX and one output argument YY and it uses cubic spline interpolation to find YY. This is done using the values of the underlying function Y as knots and then calculating the values of the spline at the points in the vector XX. The vector X specifies the points at which the data Y is given. Thus, when using `spline.m` with the data for the SIC problem with trade-up, X is a vector containing the capacitites for which the upper bound is calculated using one of the two methods described above, Y is the value of the upper bound at these capacities and XX is a vector with the values $1, 2, \ldots, C$, where C is the capacity of the aircraft. A spline is calculated for each specified time and the upper bounds for the value function are saved in a $C \times LT$ matrix, where LT is the number of times in which a value is calculated. When the upper bounds for the value function have been calculated for all capacities, the upper bounds for $\Delta V_t(x)$ for all capacities are determined from these. Then for all capacities, the lower bounds for $\Delta V_t(x)$ are calculated from the upper bounds. This can be done in different ways as described in Section 4.1.4.

The program `RevenueSpline.m` is used to calculate the revenue from the simulation process for the adjusted B&P C,G&J method. This function calculates the revenue by using the matrices containing upper and lower bounds for $\Delta V_t(x)$ for all remaining capacities and some prespecified times. The procedure for accepting/rejecting a request explained in the four items on page 50 is implemented in this program. Similarly, `RevenueSplineBPLP.m` is used to calculate the revenue from the simulation process for the B&P LP C,G&J method. The only difference from `RevenueSpline.m` is that the decision rule from the You method (4.12) page 46 is incorporated in the items on page 50. For both programs all other procedures are the same.

Initially, $\Delta \check{V}_{\tau-1}(x)$ and $\Delta \hat{V}_{\tau-1}(x)$ must be found, where $\tau - 1$ is the decision period immediately after that of the arrival. To do this it is first determined in which data interval the decision period immediately after the arrival time τ is. Then since the bounds are only known in some or all data intervals, the values of the bounds at time $\tau - 1$ are determined by using linear interpolation between the start time and the end time of the data interval containing $\tau - 1$.

The predefined Matlab function `interp1.m` is used to make the linear interpolation. This function has three inputs Z, U, ZI and one output argument UI and it interpolates between the values of U to find UI at the points in the vector ZI. The vector Z specifies the points at which the data U is given. Thus, when using `interp1.m` on the SIC problem with trade-up, Z is a vector containing two elements, i.e., the decision period in which the data interval containing $\tau - 1$ begins and the decision period at which this data interval ends. The vector U contains the corresponding values of the lower or upper bound, depending on which bound is interpolated and ZI is the decision period n, which corresponds to the time τ at which the request occurs. This interpolation is done for both the upper and the lower bounds, such that $\Delta\check{V}_{\tau-1}(x)$ and $\Delta\hat{V}_{\tau-1}(x)$ are determined. When this is done the four items described on page 50, ot the revised items for the B&P LP C,G&J method, are run through and if the request is rejected the procedure regarding trade-up to the lowest open class described for the previous methods is used. The total revenue obtained from the simulation process is then calculated and given as output from the program.

The "Max" C&H Solution Method

The method "Max", which is a variant of the C&H method, is implemented in `hybridMax.m`. This program is made such that the exact method and the approximation method can use different values of ϵ. The data interval in which the switch between the exact and approximate methods must be made is chosen by the user and is therefore given as input to `hybridMax.m`. This input variable is denoted $switchTime$. The demand matrix can then be divided into two matrices, one for the exact method and one for the approximation method, such that the first $switchTime$ number of data intervals are for the exact method and the last $r - switchTime$ data intervals are for the approximation method, where r is the total number of data intervals. Similarly, the matrix containing the trade-up probabilities between all classes is also split into two. Next, the You method which is chosen as the exact method for the general SIC problem with trade-up is run with these reduced matrices and an optimal solution for the first $switchTime$ data intervals is obtained.

Regarding the approximation method, another input to the program `hybridMax.m` is an indicator, such that if the indicator has the value 1, then the approximation method is one of the B&P methods described in Section 4.1.3 and the open classes are determined by using the decision rule given in that section. If the indicator is set to another value than 1, then the You solution method is used as the approximation method. The You solution method is then used with a rather big ϵ, such that it is less accurate and faster. No matter which of the methods is used as the approximation method, this is run with the reduced matrices for the approximation period, and a solution for the last data intervals is obtained.

The switch is made by, for all combinations of remaining capacity, adding the total expected revenue for the data intervals in the exact period and the total expected revenue for the data intervals in the approximation period. Note that the sum is made such that if 7 seats remain after the approximation period, then the total expected revenue from this period is added to the total expected revenue from the exact period, when only 7 seats are available. This way the number of seats which are available is taken into account and this is done for all combinations of remaining capacities. By taking the maximum of all these combinations it is determined how many seats it is optimal to sell in both the exact and approximation periods, i.e., which combination of remaining seats that yields the maximum total expected revenue for the entire booking period. Then the open classes in the approximation period are determined for the capacities $C = 1, 2, \ldots, index_{max}$, where $index_{max}$ is the optimal remaining capacity after the approximation period determined in the maximization.

If the approximation method is the B&P method with trade-up, there are three outputs from `hybridMax.m`. These are the matrices H and j from the You solution method which are used in the acceptance/rejection decision in the exact part of the booking period and the matrix M with lowest open classes from the B&P method which is used in the approximation period. If the approximation method is the You solution method then the outputs are only H and j, which is a combination of the matrices obtained in the exact and approximation method.

The program `RevenueHybrid.m` is used to calculate the revenue from the simulation process. This function calculates the revenue by using the matrices given as outputs from the program `hybridMax.m`. Depending on the input given to `RevenueHybrid.m` the program takes the use of different values of ϵ into account. If different values of ϵ are used, initially the vector containing the decision periods in which an arrival occurs is made as a combination of the decision periods for the approximation period and the exact period.

If the You solution method is used as both the approximation method and the exact method with different values of ϵ, then the acceptance/rejection procedure is as described in the section about the You solution method on page 65. If the adjusted B&P method with trade-up is used as the approximation method, the procedure described in the section about the adjusted B&P method on page 67 is used for the requests occuring in the approximation period and the procedure described for the You method is used in the exact period.

The "AddCol" C&H Method

Another variant of the C&H method is the method "Add Column", which is implemented in `hybridAddCol.m`.

The difference between the programs `hybridAddCol.m` and `hybridMax.m` is how the switch between the exact and the approximation methods is made. In the program `hybridAddCol.m` the switch is handled by first calculating the expected revenue in the exact period with the You solution method. Then the expected revenue is calculated for all data intervals in the approximation period and for all remaining capacities using either the B&P solution method or the You solution method depending on the value of the indicator. When the expected revenue has been calculated for the approximation period, this is only from the beginning of the booking period and until the time of the switch, but it does not take the revenue which can be obtained in the exact period into account. Hence, to make the switch the total expected revenue for the exact period, i.e., the vector for the time of the switch for all values of remaining capacity, is added to all the vectors in the matrix containing the expected revenue from the approximation period for each decision period and for all remaining capacities. This matrix then contains the approximation of the value function and is hence used to determine the open classes in the approximation period. The outputs from the program are the same as for `hybridMax.m`, see Table C.3, Appendix C.

The program `RevenueHybrid.m` is used for the calculation of the revenue obtained from the simulation process. This program is described for the "Max" method.

The "Combi" C&H Method

The last variant of the C&H method is the method "Combi", which is implemented in `hybridCombi.m`.

For this hybrid method the switch is made by using the exact method in the exact period but then the approximation method is used in the entire booking period. These matrices are then combined such that the first part of the matrix is the exact method and the last part of the matrix is the part of the approximation matrix which is only for the approximation period. The matrices H, j and M are given as output and used in the acceptance/rejection decision.

Again, the program `RevenueHybrid.m` is used for the calculation of the revenue obtained from the simulation process.

4.3.2 Simplified SIC with Trade-Up

The programs described in this section are enclosed on the CD in the folder *Simple SIC with TU/Main Functions*.

It is chosen not to describe the implementation of all the programs in this section, since for some of the methods the implementations are very similar to those for the general methods. This is for instance the case for the simplified adjusted B&P method with trade-up and the simplified C,G&J method using the B&P method for calculating the bounds. The only difference in these functions is that the demand is only for class k and is hence given in a vector, and the demand matrix is then obtained by multiplying the demand vector with the trade-up rates as described in (4.24) page 60, since demand for other classes than the lowest class only occurs through trade-up.

The input variables for the programs in this section have already been described in Table C.2 and Table C.5, Appendix C. The only difference is that the demand matrix D in Table C.2 is now a demand vector for class k.

Simulation

The simulation of the booking process for the simplified SIC problem with trade-up is implemented in the program `SimulationTradeupSimple.m`. This program is very similar to the program `Simulation.m` where trade-up is not incorporated, see Section 3.3.1 for a description. There is an additional input argument, though, which is a matrix TU containing the trade-up probabilities $TU_{k,i}$. The (i, j)th element in the matrix is the probability, that a passenger is willing to trade up from the lowest-fare class in the market, class k, to class i and all higher-fare classes in data interval $r - j + 1$, where r is the total number of data intervals. Hence, the probabilities in the matrix TU are cumulated, such that for instance the probability for trade-up from class k to class i includes the probabilities of trade-up from class k to classes $i - 1, \dots, 1$.

Recall, the assumption for the simplified SIC problem with trade-up is that all passengers request the lowest class in the market, class k, since there is no difference in the buying conditions for different fare classes. Therefore demand for another class than class k only occurs through trade-up to this class. Hence, instead of having a demand matrix as input, a demand vector containing the total demand for class k in each data interval is input.

A final difference between the simulation programs `Simulation.m` and `SimulationTradeupSimple.m` is the simulation of the requested class. Since requests are always for the lowest class in the market, instead of simulating the requested class, the highest class a passenger is willing to trade up to is simulated. For instance, if a passenger is willing to trade up to class i that passenger is also willing to buy all lower-fare classes, classes $i+1,\ldots,k$. The highest class a passenger is willing to trade up to is simulated by using the matrix TU. As mentioned before, the trade-up probabilities in this matrix are cumulated in the classes, such that

$$TU_{k,1} \leq TU_{k,2} \leq \cdots \leq TU_{k,k} = 1.$$

Thus, for a specific data interval the highest-fare class which a passenger will trade up to can be simulated by generating a uniformly distributed random number, RND $\in [0,1]$ and then comparing this with the vector in TU corresponding to that data interval. The index of the smallest element in that vector which is larger than RND is the highest-fare class which the passenger is willing to trade up to.

The outputs from the program are four vectors where three of these are the same as for `Simulation.m`, these are t, *dataint* and n. The fourth output vector is *maxclass*, which contains the highest class the arrival is willing to trade up to. For description of the first three outputs, see Table C.2, Appendix C.

The Simplified You Solution Method

The simplified You solution method is implemented in `YouTalluriSimple.m`. This program is much simpler than `YouTalluri.m`, since the method is simpler. In the simplified version it is only necessary to determine the lowest open class, since the request is always for class k. The lowest open class is determined by the maximization given in (4.21) page 58. The output from the program is a $C \times T$ matrix containing the lowest open classes and a matrix containing the total expected revenue for all decision periods and all remaining capacities.

The program `matrixRevenueSimple.m` is used to calculate the revenue from the simulation process. This function calculates the revenue by using a matrix containing the lowest open class. Hence, when it is determined whether to accept or reject a request in decision period t with a remaining capacity x, the (x,t)th element in the matrix is compared with the most expensive class the customer is willing to trade up to, which is *maxClass* from the simulation process. If the class is a higher-fare class than the lowest open class, then the request is accepted for the lowest open class and otherwise it is rejected. For the simplified problem there is no procedure for trade-up after the rejection, since trade-up is incorporated in the simulation of the requested class as described in the previous section.

The HM Method

The HM method is implemented in the program `HMSimple.m`. The equation given in (4.23) page 59 is used to calculate the value of $V_t(x)$. Hence, first the request probabilities $P(D_j^t = s)$ for fare class j for all decision periods and all possible values of number of arrivals s has to be determined. This is done by using the predefined Matlab function `poisspdf.m`. Next the values of D_j^t are calculated by using (4.22), where the trade-up probabilities from fare class k to all other fare classes are input to the program. Now the values of $V_t(x)$ are calculated and the lowest open classes can be determined as described in Section 4.2.2. These lowest open classes are stored in a $C \times T$ matrix, where C is the capacity and T is the total number of decision periods. This matrix and a $C \times T$ matrix containing the value function for all remaining capacities and for all decision periods are given as output.

The program `matrixRevenueSimple.m` is used to calculate the revenue from the simulation process, see the previous section for description.

The Simplified C&H Solution Method

The simplified C&H solution method is implemented in three different programs depending on which of the three methods "Max", "Add Column" and "Combi" is used to make the switch between the approximation method and the exact method. The simplified method "Max" is implemented in `hybridSimpleMax.m`, the simplified method "Add Column" is implemented in `hybridSimpleAddCol.m` and finally the simplified method "Combi" is implemented in `hybridSimpleCombi.m`. Only the differences between the simplified and the general programs are described.

One of the differences between `hybridMax.m` and `hybridSimpleMax.m` is the meaning of the input variable *indicator* which for the general problem determined which method was used as the approximation method. For the simplified problem only the You method with a large ϵ is used as the approximation method, instead the *indicator* decides how the lowest open classes are determined in the approximation period. If *indicator* $= 1$ then the decision rule $\Delta V_{t-1}(x) \leq F$ is used to determine the lowest open classes in the approximation period and otherwise the decision rule given in (4.21) page 58 is used to determine the lowest open classes. Another difference between the two programs is that in the simplified version the outputs from the exact method and the approximation method are the same, i.e., it is a matrix containing the lowest open classes. Therefore there is only one output from `hybridSimpleMax.m`, which is the combined matrix for both the exact and approximation methods containing the lowest open classes. The switch between the methods is made as described for the general problem.

The difference between `hybridAddCol.m` and `hybridSimpleAddCol.m` is that the simplified version can use the HM and the simplified You methods as the exact method and only the simplified You method with a rather large ϵ as the approximation method. Therefore the way the lowest open class is calculated for the approximation method is as described in Section 4.2.1. The switch and combined matrix containing the lowest open classes is made in a similar way as for the general method. The output from the `hybridSimpleAddCol.m` is only one matrix containing the lowest open classes.

For the hybrid method "Combi" the two programs `hybridCombi.m` and `hybridSimpleCombi.m` differ as described for "Max". The revenue obtained from the simulation process is for all three methods calculated by using the program `matrixRevenueSimple.m`. A description of this function can be seen in the section about the simplified You solution method.

Chapter 5

Numerical Experiments

In this chapter numerical experiments for the different methods with and without trade-up are presented. For both the SIC problem without and with trade-up, the data used in the experiments are described.

In Section 5.1 the SIC problem without trade-up is investigated. In this section it is described how the parameters are tuned and the results from the numerical experiments are presented.

Next, in Section 5.2 the general SIC problem with trade-up is examined. In this section the parameter tuning for the general methods and the results obtained with these are presented.

Finally, in Section 5.3 the tuning of the various parameters in the methods for the simplified SIC problem with trade-up is described. Furthermore the results from the numerical experiments are given.

Common for all parameter tunings and results is that these are based on 1000 runs of the methods on different simulations of arrivals for five test sets with four different capacities. The test sets and capacities are described further in the respective sections. Furthermore all results are shown for tests run on a SUN Fire 3800 with a 1200 Mhz processor and 4 GB RAM.

5.1 SIC without Trade-up

The data needed for solving the SIC problem without trade-up are

- Demand forecasts for each data interval and each class.
- Capacity of the aircraft.
- Fare for each class.

- Value of ϵ, where a smaller ϵ yields a smaller probability of two or more arrivals in a decision period.

Furthermore, for the simulation of arrivals it is necessary to know the times at which each data interval begins.

The first three items are data which are provided by the airline, whereas the value of ϵ is chosen by the user of the program. Hence, the value of ϵ must be tuned before the different solution methods can be used. The demand forecasts can for instance be made by the airline from historic booking data by taking an average of the number of bookings over a number of similar flights.

The data obtained from BA contains 16 classes, 15 data intervals and four capacities of $C = 80$, $C = 100$, $C = 120$ and $C = 140$.

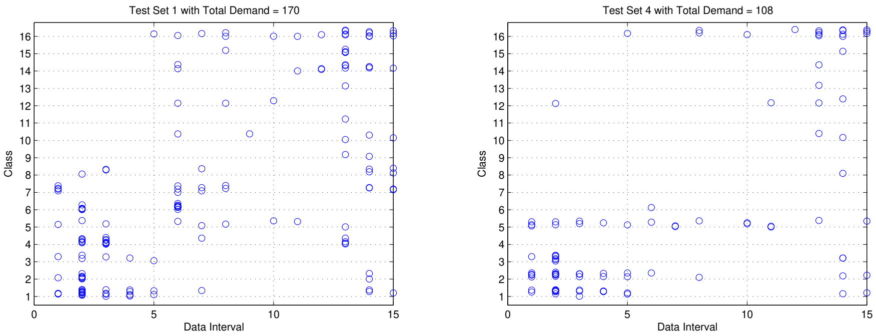
It is important to have different data sets to obtain general results, since if a method yields better results than another for one data set, this may not be the case for a different data set. Therefore five different data sets are provided by the airline. These are assumed to be representative for real life data. The total demand and fares in the data sets can be seen in Table 5.1.

Test Set	Total Demand	Fares
1	170	[290, 210, 160, 145, 130, 115, 100, 90, 85, 78, 65, 55, 45, 35, 30, 20]
2	127	[290, 210, 160, 145, 130, 115, 100, 90, 85, 78, 65, 55, 45, 35, 30, 20]
3	109	[160, 120, 110, 90, 70, 65, 60, 50, 45, 40, 35, 30, 20, 18, 15, 12]
4	108	[270, 220, 190, 140, 120, 110, 100, 90, 85, 75, 70, 63, 60, 45, 35, 30]
5	116	[350, 300, 290, 210, 180, 160, 145, 130, 110, 100, 95, 90, 80, 75, 50, 30]

Table 5.1: Demand and Fares for Five Test Sets without Trade-Up.

The difference in the total demand for test set 1 and test set 4 is seen to be large. The distribution of the demand on classes and in data intervals might be very different as well. In Figure 5.1 the demand patterns for these two test sets are shown. Notice that data interval 1 is nearest departure and data interval 15 is the beginning of the booking period.

The demand for test set 1 is seen to be distributed evenly throughout the booking period. The demand for the discounted classes occur mainly in the beginning of the booking period, between data intervals 10 and 15. In the middle of the booking period, in data intervals 5 to 10, the demand is both for discounted and high-fare classes, whereas close to departure the demand is for the high-fare classes. For test set 4 the demand pattern is different. In the beginning of the booking period the demand is both for discounted and more expensive fare classes. In the middle of the booking period the demand is fairly low, and close to departure with one exception the demand is only



(a) Demand for Test Set 1.

(b) Demand for Test Set 4.

Figure 5.1: Demand Pattern for Test Sets 1 and 4 without Trade-Up.

for classes 1 to 5. For test set 4 the demand for classes 6 to 12 is very low compared to test set 1.

The demand patterns for test sets 2, 3 and 5 are given in Appendix B. Here it is seen that the demand patterns for test sets 2 and 5 are similar to the demand pattern of test set 1. The demand pattern for test set 3 seems to be a combination of the demand patterns for test sets 1 and 4. Therefore it can be concluded that the test sets differ in both fares, total demand and distribution of demand.

In the next section the parameter tuning is described first and afterwards the results obtained from the numerical experiments are presented. The methods which are compared are the EMSRa, EMSRb, L&H and B&P methods, where the two last methods use booking limits for each decision period. Furthermore two versions of the L&H and B&P methods are introduced, where the booking limits are only updated at the same times as in the EMSR methods. These are denoted L&Hup and B&Pup, respectively.

When reading the next sections, keep in mind, as described in Section 1.1, that each optimization for the SIC problem must take less than 0.85 seconds to be computationally feasible for BA.

5.1.1 Parameter Tuning

The value of ϵ must be tuned before the methods can be used to solve the SIC problem without trade-up. Recall that the smaller ϵ is, the smaller is the probability of two or more arrivals in one decision period. Therefore the smaller value of ϵ, the more accurate is the model, since for smaller values of ϵ the assumption that there can be at most one request per decision period is more likely to be satisfied. In this report the tuning of ϵ is with respect to the highest revenue and the shortest running time, since in practice these are the important aspects.

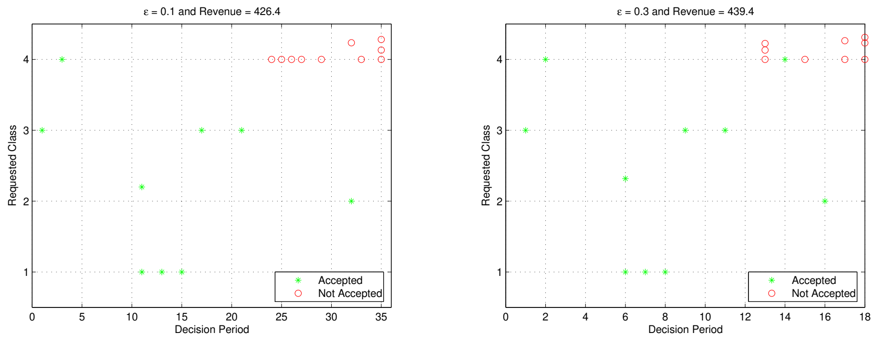
With decreasing values of ϵ, an increase in revenue is expected, but also the running times are then expected to get longer. Thus there is a tradeoff between the revenue gained by using a method and the running time. As mentioned in Section 1.1, the methods have to be computationally fast to be usable, since the airline has an extensive number of flights in the system at all times and the booking limits for each flight are updated frequently. Therefore the final value of ϵ which is chosen may not yield the highest revenue, since the running time might be too long when using that particular ϵ.

A Small Example

A test example which is smaller than the test sets mentioned in the previous section is generated to understand the connection between the revenue obtained by the methods when simulating arrivals, the distribution of arrivals and the value of ϵ. To see the dependence on the distribution of arrivals, the simulations are made such that the arrival times are chosen in advance and not randomly. This way different kind of arrival patterns can be generated. Note that for a specific arrival pattern only the class which is requested by each arrival varies. In the first arrival process requests occur mainly in the beginning of the booking period, in the second process the arrivals are distributed evenly throughout the booking period and in the last process the arrivals occur mainly in the end of the booking period, i.e., close to departure. The small test set is made by scaling the demand, fares and capacities in test set 1 to a case with only four fare classes, five data intervals and capacity $C = 10$. The fares used in the test example are $F = [80.50, 43.50, 28.30, 13.00]$ and the total demand is $D = 17$. Using $\epsilon = 0.1$ the five data intervals are divided into a total of 36 decision periods, whereas using $\epsilon = 0.3$ the data intervals are divided into 18 decision periods. In total 20 arrivals are simulated and for $\epsilon = 0.1$ in the second arrival process, the assumption about one or no arrivals in each decision period is satisfied. This is not the case for $\epsilon = 0.3$.

The L&H method with $\epsilon = 0.1$ and $\epsilon = 0.3$ is run 1000 times on the test example for all three arrival processes. First the process with many arrivals in the beginning of the booking period is used. The results from this is that 286 times out of the 1000 runs, the larger value of ϵ obtains a higher revenue than the smaller ϵ does. Furthermore in 483 of the runs the revenues were the same. Thus, in 231 times of the 1000 runs, the revenues with $\epsilon = 0.1$ is better than the revenues with $\epsilon = 0.3$. Hence it seems that when using

$\epsilon = 0.3$ similar results as when using $\epsilon = 0.1$ are obtained. To understand this result the arrival patterns with the two values of ϵ are shown in Figure 5.2. In this figure it is also seen, which requests are rejected and which are accepted. The arrivals mainly occur in the beginning of the booking period, since decision period 0 corresponds to departure.



(a) Test with $\epsilon = 0.1$.

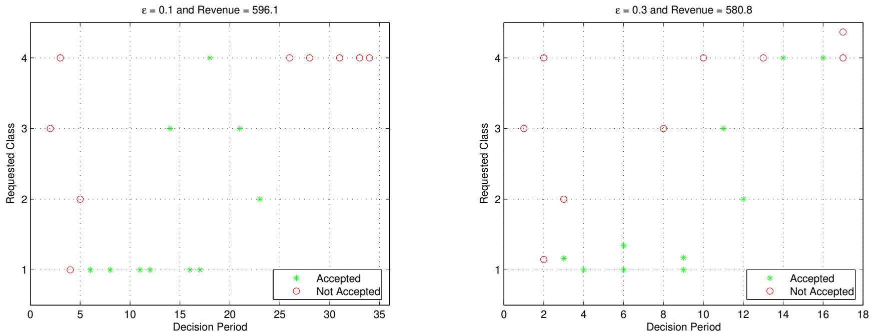
(b) Test with $\epsilon = 0.3$.

Figure 5.2: Arrival Pattern for the First Simulation.

It is seen, that with $\epsilon = 0.1$ the L&H method only accepts one request in the beginning of the booking period in the expectance of requests later on. This causes the lower revenue, since the demand later in the booking period is not as high as expected. With $\epsilon = 0.3$ the L&H method accepts two requests in the beginning of the booking period and in the case where the arrivals occur mainly in the start of the booking period, this yields a higher revenue.

In the second simulation the arrivals are equally distributed throughout the booking period. Again the L&H method is run 1000 times with $\epsilon = 0.1$ and $\epsilon = 0.3$. The results are that 844 times out of the 1000 runs $\epsilon = 0.1$ is better than $\epsilon = 0.3$, and 9 times $\epsilon = 0.3$ is better than $\epsilon = 0.1$. Hence the general observation is that now the smaller value of ϵ creates a higher revenue. The arrival pattern and the accepted and rejected requests are shown in Figure 5.3. Note that as mentioned earlier for $\epsilon = 0.1$ only one or no arrival occur in each decision period, hence the assumption for the model is satisfied. For $\epsilon = 0.3$ there are two arrivals in some decision periods, thus the assumption is not satisfied. See for instance decision periods 17 and 6 in Figure 5.3(b).

As with the first simulation with $\epsilon = 0.1$ the L&H method is seen to reject many arrivals in the beginning of the booking period, where the arrivals



(a) Test with $\epsilon = 0.1$. (b) Test with $\epsilon = 0.3$.

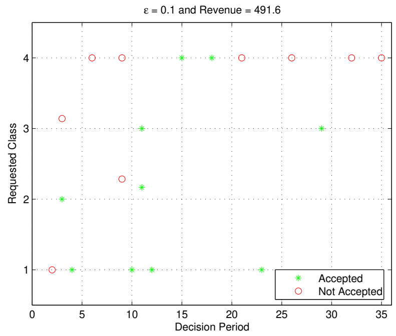
Figure 5.3: Arrival Pattern for the Second Simulation.

are mainly for the discounted classes. With the arrivals equally distributed throughout the booking period, this yields a higher revenue than with $\epsilon = 0.3$, since the expected requests come later in the booking period. In figure 5.3 it is seen that the problem, when using $\epsilon = 0.3$ is that too many requests are accepted in the beginning of the booking period. Hence, later on in the booking process requests for more expensive classes are rejected due to lack of capacity. For instance, a class 3 request is rejected with $\epsilon = 0.3$ and accepted with $\epsilon = 0.1$.

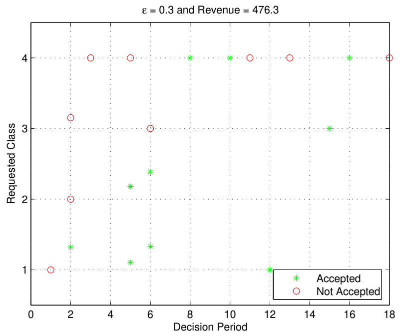
Lastly the L&H method was run 1000 times on the small test example with the third simulation. In this simulation the arrivals occur mainly in the end of the booking period. As with the second simulation $\epsilon = 0.1$ performed better than $\epsilon = 0.3$, since in 740 runs the first was better than the latter and only 17 times the opposite occured.

In Figure 5.4 it is seen that the reason that L&H perform worse for $\epsilon = 0.3$ than for $\epsilon = 0.1$ is the same as with the second simulation. Requests for a discounted fare class are accepted in the beginning of the booking period and hence a request for a more expensive fare has to be rejected later on in the booking period.

From the above tests with the small example it can be concluded that if the arrivals occur mainly in the beginning of the booking period, then a larger value of ϵ should be used in the solution method. The reason for this is that the two values of ϵ give similar results, but a larger ϵ yields a shorter running time. Is the arrival process expected to be like the second or third simulation, then a smaller value of ϵ should be used. In real life, requests are expected to arrive throughout the booking process with more arrivals



(a) Test with $\epsilon = 0.1$.



(b) Test with $\epsilon = 0.3$.

Figure 5.4: Arrival Pattern for the Third Simulation.

closer to departure than in the beginning of the booking period. Hence, when tuning the value of ϵ for the solution methods on the different test sets it is expected that smaller values of ϵ perform better than larger values.

Tuning of ϵ

In the following the parameter tuning for the L&H method with and without updating the booking limits will be presented first. Recall that in the L&H method booking limits are calculated for each decision period in the booking period whereas in the L&H method with update, the booking limits are updated at fewer points in time determined by the airline. After the parameter tuning of the L&H methods the parameter tuning for the B&P method will be described. The parameter tuning is done for four different values of ϵ, which are chosen to be $\epsilon = 0.001$, $\epsilon = 0.1$, $\epsilon = 0.2$ and $\epsilon = 0.3$.

In Table D.2, Appendix D.1.1, a comparison of the revenues obtained with the different values of ϵ is shown. In the table the test set denoted L&H 1 is when the L&H solution method is used on test set 1 and L&H 2 is with test set 2, etc. In L&H 1 booking limits for each decision period are known. The test set denoted L&Hup 1 is, when the booking limits determined with the L&H method are updated at times specified by the airline and used on test set 1.

For capacities larger than the total demands in the test sets, almost all demand can be satisfied and thus not many requests must be rejected. Hence, the numbers in the comparisons for these capacities and test sets are so small that nothing can be concluded about the value of ϵ using these comparisons.

Therefore the comparisons are not shown for all combinations of test sets and capacities. For instance, for test set 2 the total demand is $D = 127$ and the results of the comparisons for $C = 140$ are that in 40 of the 1000 runs using ϵ_1 yields a higher revenue than using ϵ_2 and in 60 of the 1000 runs the opposite is the case. Hence, when using the two values of ϵ, the same revenue is obtained 900 times, and therefore the value of ϵ is less important when the capacity is larger than the total demand. Thus the comparisons for L&H 2 and L&Hup 2 are not shown for $C = 140$. Similarly L&H 3, L&H 4, L&Hup 3 and L&Hup 4 are not shown for $C = 120$ and $C = 140$, since the total demand for test sets 3 and 4 is $D = 109$ and $D = 108$, respectively. Finally the comparisons for L&H 5 and L&Hup 5 are not shown for $C = 120$ and $C = 140$ since the total demand for test set 5 is $D = 116$.

When comparing the revenues obtained by using the different values of ϵ, it is in Table D.2, Appendix D.1.1, shows that $\epsilon = 0.001$ yields the highest revenue for the L&H method without updates. The number of times that $\epsilon = 0.001$ is better than $\epsilon = 0.1$ must be compared with the average running time when using the different values of ϵ, since the running time is a big issue in the usability of the method as described previously. The average running times are given in Table D.1, Appendix D.1.1. In this table the running time with $\epsilon = 0.001$ is seen to be generally 10 times longer than the running time with $\epsilon = 0.1$. When comparing the results for $\epsilon = 0.1$ with those obtained with $\epsilon = 0.2$ it is seen that $\epsilon = 0.1$ in the majority of the runs yields a higher revenue than $\epsilon = 0.2$. Compared with this, the reduction in running time by using the larger value of ϵ is not too large, therefore $\epsilon = 0.1$ is chosen for the L&H method without updates. For the L&H method with updates Table D.2, Appendix D.1.1, shows that $\epsilon = 0.1$ produces the highest revenue in all cases and the running times for this are similar to those for the L&H method with $\epsilon = 0.1$, hence this value is also chosen for the L&H method with updates.

These choices of ϵ are based on the assumptions that if the programs are implemented in another programming language than Matlab, for example C++, then the running times can be reduced significantly, such that the limit of 0.85 seconds is satisfied. Another way of satisfying this time limit is to choose $\epsilon = 0.2$ but in this case the revenues obtained are lower.

As with the L&H method, the method denoted B&P 1 is, when the B&P solution method is used on test set 1, etc. In Table D.4, Appendix D.1.2, a comparison of the revenues obtained with the different values of ϵ is given. The results for the test sets are shown for the same capacities as with the L&H method for the same reasons. The values in the table are much smaller than those for the L&H method in Table D.2. Thus, for the B&P method the results obtained using the four values of ϵ are very similar. Hence, this method is less dependent on the value of ϵ. This might be because the demand used in the approximation of the value function is the expected demand from the present decision period until departure. Thus, whether there are five or ten decision periods in a data interval does not change much in the value of the expected demand until departure. Therefore the value function $V_t(x)$ does not change much and hence the decisions of accepting or rejecting a request are almost the same, yielding a similar revenue.

The results obtained when using the different values of ϵ are similar, but $\epsilon = 0.1$ yields slightly better results than both $\epsilon = 0.2$ and $\epsilon = 0.3$ and the running time is not reduced much by using the last two values of ϵ. Therefore, of these three values $\epsilon = 0.1$ is chosen. When choosing between $\epsilon = 0.001$ or $\epsilon = 0.1$, the last value is chosen from a running time perspective. The running time with $\epsilon = 0.1$ is generally 10 times shorter than the running time with $\epsilon = 0.001$, see Table D.3, Appendix D.1.2. Thus, for the B&P method it is also chosen to use $\epsilon = 0.1$. Note that for this method with the chosen value of ϵ the time limit of 0.85 seconds is satisfied.

5.1.2 Results

In the previous section the value of ϵ was tuned to $\epsilon = 0.1$ for both the L&H method with and without update and the B&P method. In this section the different methods are run with both $\epsilon = 0.1$ and $\epsilon = 0.01$ to show how good results can be obtained with the L&H method, if a longer running time is acceptable, i.e., if a smaller value of ϵ is used. In the following $\epsilon_1 = 0.01$ and $\epsilon_2 = 0.1$.

The results are shown for test sets 1 and 4, since these test sets differ the most, as described in Section 5.1. Hence, the results from these two test sets are assumed to show the general tendency, if one exists. The results for test sets 2, 3 and 5 are shown in Appendix E. In Tables 5.2, 5.3 and 5.4 the updating of the booking limits for L&Hup, B&Pup, EMSRa and EMSRb is done at some prespecified times given by the airline. These times are such that initially in the booking period the booking limits are updated at the beginning of each data interval and later in the booking period the updating is intensified. The specific times for the update of the booking limits can be seen in the vector *DecPer* in Appendix E.

The reason for calculating the L&H and B&P methods with updates of the booking limits at prespecified times, L&Hup and B&Pup, is that the system, which the airlines use now, can only handle one set of booking limits at a time and thus cannot handle a set for each decision period. Therefore, even though data is available for the booking limits in each decision period, this cannot be used. In the numerical experiments run in this report, it is chosen to show results with updates of the booking limits in each decision period for the L&H and B&P methods anyway, since these can be used as benchmarking results. Furthermore, if the results show that there is a large gain in revenue by using all the information available, the airline may consider changing their system.

	$C = 80$		$C = 100$		$C = 120$		$C = 140$	
	$\epsilon_1 = 0.01$	$\epsilon_2 = 0.1$	$\epsilon_1 = 0.01$	$\epsilon_2 = 0.1$	$\epsilon_1 = 0.01$	$\epsilon_2 = 0.1$	$\epsilon_1 = 0.01$	$\epsilon_2 = 0.1$
L&H 1	3.91 s	1.14 s	4.78 s	1.34 s	5.61 s	1.61 s	6.41 s	1.86 s
L&Hup 1	3.93 s	1.16 s	4.80 s	1.36 s	5.64 s	1.63 s	6.43 s	1.88 s
B&P 1	1.53 s	0.45 s	2.00 s	0.57 s	2.49 s	0.71 s	2.98 s	0.85 s
B&Pup 1	0.33 s	0.21 s	0.35 s	0.22 s	0.37 s	0.24 s	0.38 s	0.25 s
EMSRa 1	1.09 s	1.08 s	1.12 s	1.11 s	1.13 s	1.15 s	1.14 s	1.13 s
EMSRb 1	1.13 s	1.12 s	1.16 s	1.15 s	1.17 s	1.16 s	1.19 s	1.17 s
L&H 4	2.45 s	0.72 s	3.03 s	0.87 s	3.54 s	1.08 s	4.31 s	1.20 s
L&Hup 4	2.47 s	0.74 s	3.05 s	0.89 s	3.55 s	1.09 s	4.33 s	1.21 s
B&P 4	1.26 s	0.36 s	1.55 s	0.45 s	1.88 s	0.55 s	2.35 s	0.65 s
B&Pup 4	0.27 s	0.18 s	0.29 s	0.19 s	0.30 s	0.21 s	0.32 s	0.22 s
EMSRa 4	0.94 s	0.93 s	0.96 s	0.95 s	0.99 s	0.98 s	1.02 s	1.01 s
EMSRb 4	0.97 s	0.96 s	0.99 s	0.98 s	1.02 s	1.01 s	1.05 s	1.04 s

Table 5.2: Running Times for the Methods without Trade-Up.

In Table 5.2 the average running times for the different methods are shown for test sets 1 and 4. These are the total times it takes to calculate the revenue obtained when using the methods on the simulated arrival process. The running times for the methods with ϵ_1 are seen to be approximately three to four times longer than with ϵ_2, except for EMSRa and EMSRb since these do not depend on the value of ϵ. The small variations in the running times for these methods are due to the randomness of the number of arrivals in the simulation of the arrival process. Furthermore, the L&H method where the booking limits are updated is seen to be the slowest method, whereas the B&P method with updates is the fastest. L&Hup is the slowest method, since when using this method the entire matrix of booking limits as for L&H needs to be calculated and then specific columns of this are used, yielding the longer running time compared to L&H. For B&Pup the booking limits are only calculated at the same prespecified times as the EMSRa and EMSRb methods. Note that the B&P method with ϵ_2 is approximately twice as fast as the EMSRb method. Furthermore, in Table 5.2 the B&P method with $\epsilon_2 = 0.1$ is seen to be the only method which satisfies the time constraint of 0.85 seconds. Not even the EMSRb method which is currently used by most airlines satisfies this constraint. This substantiates the previously mentioned assumption that implementing the methods in another programming language reduces running times significantly.

To determine if one of the methods yields a significantly higher revenue than the others, it is necessary to use a statistical tool. In this report a paired t-test is used. A paired t-test is normally used to determine whether two paired sets differ from each other in a significant way. The data which are to be compared in this report are naturally paired, since for each of the 1000 runs the revenues are calculated for the same simulated arrivals and test set but with different methods. For each of the 1000 runs the difference between the revenues obtained with the methods are calculated for all combinations of two methods. Let the pair of random variables (Ψ_i, Ω_i) denote the revenue obtained from two different methods in the ith run, for $i = 1, 2, \ldots, 1000$, then the statistical analysis proceeds by considering the differences

$$\Gamma_i = \Psi_i - \Omega_i \quad \text{for} \quad i = 1, 2, \ldots, 1000.$$

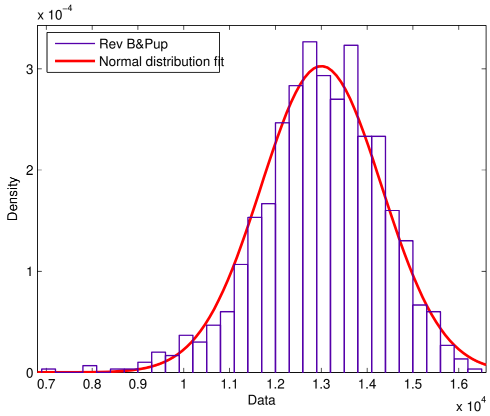


Figure 5.5: B&Pup with ϵ_2 fitted with Normal Distribution.

This collection of differences is then treated as a random sample of size 1000 from a normal distributed population having mean $\overline{\Gamma}$. The difference follows a normal distribution if each of the components in the difference follows a normal distribution. To determine if this can be assumed, the revenues from the B&P method with $\epsilon_2 = 0.1$ is plotted together with a fitting normal distribution. In Figure 5.5 the data are seen to fit the distribution, hence the data is assumed to satisfy the assumption regarding the distribution.

The expression $\overline{\Gamma} = 0$ is interpreted as indicating that the means of the two sets are the same, i.e., the two methods yield similar revenues and $\overline{\Gamma} > 0$ as indicating that the mean revenue obtained by the first method is higher than that obtained by the second method. The null hypothesis, which is tested in the paired t-test is then

$$\textit{Null hypothesis:} \quad \overline{\Gamma} = 0 \qquad \text{against} \qquad \textit{Alternative hypothesis:} \quad \overline{\Gamma} > 0.$$

For more information on t-tests see [17] by Johnson.

A paired t-test with a significance level of $\alpha = 5\%$ is made for all 15 combinations of every two methods and this is done for both ϵ_1 and ϵ_2. In this way it is statistically determined, whether the revenue obtained by using one method is significantly different from the revenue obtained with another method.

	L&Hup 1		**B&P 1**		**B&Pup 1**		**EMSRa 1**		**EMSRb 1**	
	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2
C=80										
L&H 1	2.18	1.84	0.18	**0.07**	1.98	1.70	4.15	3.91	0.76	0.53
L&Hup 1	-	-	**1.99**	**1.68**	**0.19**	**0.16**	1.99	2.11	**1.43**	**1.33**
B&P 1	-	-	-	-	1.81	1.74	3.98	4.02	0.59	0.58
B&Pup 1	-	-	-	-	-	-	2.19	2.30	**1.23**	**1.17**
EMSRa 1	-	-	-	-	-	-	-	-	**3.39**	**3.44**
C=100										
L&H 1	1.86	1.67	0.15	**0.15**	1.74	1.41	2.18	1.86	0.52	0.22
L&Hup 1	-	-	**1.74**	**1.40**	**0.12**	**0.27**	0.35	0.21	**1.35**	**1.47**
B&P 1	-	-	-	-	1.59	1.55	2.03	2.02	0.37	0.35
B&Pup 1	-	-	-	-	-	-	0.46	0.50	**1.23**	**1.20**
EMSRa 1	-	-	-	-	-	-	-	-	**1.67**	**1.68**
C=120										
L&H 1	1.53	1.35	0.13	**0.35**	1.44	0.95	1.17	0.68	0.39	**0.1**
L&Hup 1	-	-	**1.44**	**0.95**	**0.09**	**0.40**	**0.41**	**0.69**	**1.14**	**1.44**
B&P 1	-	-	-	-	1.31	1.29	1.04	1.04	0.27	0.26
B&Pup 1	-	-	-	-	-	-	**0.32**	**0.30**	**1.05**	**1.03**
EMSRa 1	-	-	-	-	-	-	-	-	**0.78**	**0.79**
C=140										
L&H 1	1.17	1.17	0.24	**0.24**	1.33	0.84	0.53	0	0.28	**0.21**
L&Hup 1	-	-	**1.33**	**0.84**	0.16	**0.33**	**0.66**	**1.15**	**0.89**	**1.38**
B&P 1	-	-	-	-	1.10	1.08	0.30	0.29	0	0
B&Pup 1	-	-	-	-	-	-	**0.82**	**0.81**	**1.05**	**1.04**
EMSRa 1	-	-	-	-	-	-	-	-	**0.25**	**0.24**

Table 5.3: Rel. Diff. in % for Test Set 1 without Trade-Up.

In Tables 5.3 and 5.4 the relative differences in percentage between every combination of two methods are given. The differences between the methods are always divided by the revenue from the method which yields the smallest revenue. I.e., the relative difference is with respect to the method which gives the smallest revenue.

The values in the tables are to be read as follows

- 0:
  The null hypothesis cannot be rejected, i.e., in practice the methods yield similar revenues.

- Positive value:
  The null hypothesis is rejected, i.e., it cannot be rejected that $\overline{\Gamma} > 0$, which in practice means that the first method yields a higher revenue than the second on a 5% significance level. The number represents the relative difference in revenue between the two methods considered.

- NoD:
  In all of the 1000 runs no difference in revenue between the two methods was recorded, see Table 5.4.

A bold value implies that the method listed in the top row of the table is significantly better than the method listed in the first column. If the value is not bold, the method listed in the first column yields a significantly higher revenue than the method in the top row.

The methods are compared for the same capacities and values of ϵ, for instance when L&H 1 with $C = 80$ and $\epsilon_1 = 0.01$ is compared with B&P, the latter is also with $C = 80$ and $\epsilon_1 = 0.01$.

	L&Hup 4		**B&P 4**		**B&Pup 4**		**EMSRa 4**		**EMSRb 4**	
	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2
C=80										
L&H 4	1.09	0.54	0.50	**0.51**	1.41	0	5.17	3.88	0.39	**0.72**
L&Hup 4	-	-	**1.40**	**0.21**	0.33	**0.34**	4.09	3.40	**0.71**	**1.24**
B&P 4	-	-	-	-	0.92	0.71	4.61	4.43	**0.11**	**0.20**
B&Pup 4	-	-	-	-	-	-	3.72	3.76	**1.03**	**0.90**
EMSRa 4	-	-	-	-	-	-	-	-	**4.76**	**4.68**
C=100										
L&H 4	0.85	0.31	0.30	**0.43**	0.89	0	2.72	1.87	0.25	**0.56**
L&Hup 4	-	-	**0.89**	0	0.05	**0.31**	1.89	1.60	**0.61**	**0.85**
B&P 4	-	-	-	-	0.60	0.43	2.40	2.32	**0.06**	**0.12**
B&Pup 4	-	-	-	-	-	-	1.83	1.93	**0.65**	**0.54**
EMSRa 4	-	-	-	-	-	-	-	-	**2.48**	**2.47**
C=120										
L&H 4	0.18	0.09	0	0	0.14	0.10	0.16	0.13	0.05	0
L&Hup 4	-	-	**0.14**	**0.10**	**0.04**	0	0	0	**0.14**	**0.07**
B&P 4	-	-	-	-	0.13	0.11	0.14	0.14	0.03	0.03
B&Pup 4	-	-	-	-	-	-	0	0	**0.10**	**0.08**
EMSRa 4	-	-	-	-	-	-	-	-	**0.11**	**0.11**
C=140										
L&H 4	0.01	0.01	NoD	NoD	0.01	0.01	0	0	0	0
L&Hup 4	-	-	**0.01**	**0.01**	**0.00**	0	**0.01**	**0.00**	**0.01**	**0.00**
B&P 4	-	-	-	-	0.01	0.01	0	0	0	0
B&Pup 4	-	-	-	-	-	-	**0.00**	**0.00**	**0.00**	**0.00**
EMSRa 4	-	-	-	-	-	-	-	-	0	0

Table 5.4: Rel. Diff. in % for Test Set 4 without Trade-Up.

In Table 5.3 the L&H method with $\epsilon_1 = 0.01$ is seen to yield a significantly higher revenue than the B&P and EMSRb methods, regardless of the size of the capacity. In Table 5.4 similar results are seen for $C = 80$ and $C = 100$, but for the other capacities there is no significant difference in the revenue. This is due to the fact that for test set 4 the expected total demand is $D = 108$, hence for $C = 120$ and $C = 140$ the capacity of the aircraft is larger than the total demand. Therefore it is not necessary to reject many requests and so the difference between the methods gets small. In Table 5.3 it is also seen that when using ϵ_2, the B&P method always yields a higher revenue than the L&H method. As with ϵ_1, for test set 4 when the capacity is larger than the expected total demand, the difference between the methods gets insignificant. The L&H method with updates, L&Hup, is seen to perform bad for all test sets. It is significantly better than the EMSRa method in most cases, though, but all other methods are significantly better than L&Hup. This was expected, since as mentioned previously the L&H method is based on using all the information available, which L&Hup does not do. The B&P method with updates, B&Pup, yields generally higher revenues than the EMSRa method but smaller revenues than the EMSRb method.

In Table 5.3 it is seen that the more requests which are to be rejected due to low capacity, the larger is the relative difference between the methods. Furthermore it seems as if there is a general tendency that the greater the difference between the expected demand and the capacity is, the better are the dynamic methods compared to the static methods. Hence, this was investigated further by running the methods with different capacities such that the ratios between the total expected demand and the capacities for the test sets were the same. If the results for these tests showed the same tendency, then it could be concluded that this ratio played an important role for some of the methods in general. The results did not show this, though, and hence the tendency is not general enough to conclude anything.

To compare the B&P, L&H, EMSRa and EMSRb methods when the booking limits for the static methods are updated in each decision period, the tests were run with these new update vectors. The results from these runs are shown in the Tables E.6, E.7, E.8, E.9 and E.10 in Appendix E for test sets 1, 2, 3, 4 and 5, respectively. These results do not differ much from the results shown in this section, hence it can be concluded that the EMSRa and EMSRb methods do not gain much in revenue by updating the booking limits more frequently.

Summary of Results

Table 5.5 sums up the main results of this section. The results in this table are shown for $\epsilon_2 = 0.1$. The conclusion is that the EMSRb method yields higher revenues than the B&P method, when using $\epsilon_2 = 0.1$ for test sets 3 and 4, whereas for test sets 1, 2 and 5 the B&P method generally gives a higher revenue. Even when taking the fact that the B&P method is faster than the EMSRb method into account, it is not recommended to change the booking system to update in each decision period and use the B&P method, since nothing can be concluded about which method is the best of these two in general. If time is available, though, it is preferable to use the L&H method with ϵ_1, since this method yields the highest revenues. The time can be reduced by implementing the methods in another programming language than Matlab and thus it might be possible to use the L&H method with $\epsilon_1 = 0.01$.

	Capacity	**L&H 1**	**L&Hup 1**	**B&Pup 1**	**EMSRa 1**	**EMSRb 1**
B&P 1	$C = 80$	0.07	1.68	1.74	4.02	0.58
B&P 1	$C = 100$	0.15	1.40	1.55	2.02	0.35
B&P 1	$C = 120$	0.35	0.95	1.29	1.04	0.26
B&P 1	$C = 140$	0.24	0.84	1.08	0.29	0
	Capacity	**L&H 4**	**L&Hup 4**	**B&P 4**	**B&Pup 4**	**EMSRa 4**
EMSRb 4	$C = 80$	0.72	1.24	0.20	0.90	4.68
EMSRb 4	$C = 100$	0.56	0.85	0.12	0.54	2.47
EMSRb 4	$C = 120$	0	0.07	**0.03**	0.08	0.11
EMSRb 4	$C = 140$	0	0.00	0	0.00	0

Table 5.5: Summary of results for the SIC problem without Trade-Up.

5.2 General SIC with Trade-Up

The data needed for solving the SIC problem with trade-up using the general methods described in Section 4.1 are similar to the data needed for the methods used to solve the SIC problem without trade-up. Some additional data are needed, though.

Recall, for the general problem with trade-up the buying conditions for different fare classes are assumed to differ. Hence, requests for all fare classes are expected. The data needed are therefore

- Demand forecasts for each data interval and each class.
- Capacity of the aircraft.
- Fare for each class.
- Trade-up rates from fare class k to another class and all higher-fare classes.

These data are all provided by the airline. The demand forecasts for each data interval and each class used for the general problem with trade-up are similar to those for the problem without trade-up. There are only 8 different trade-up classes, though, and therefore the demand used for the problem with trade-up is only for these 8 trade-up classes, instead of all 16 independent classes. These demand forecasts can be estimated as explained in Section 5.1, whereas the estimation of the trade-up rates is more complicated. An example of how to estimate these is that if 10 bookings are observed in fare class 2 when this class is open and 3 bookings are observed in fare class 1 when fare class 2 is closed, then the trade-up rate from fare class 2 to class 1 is estimated to 30%. Usually the estimations of the trade-up rates are more complicated, and a more thorough explanation about these can be seen in [25] by Talluri and Van Ryzin.

Besides the data obtained from the airline, some data which are specific for the different methods, are needed. These are

- The times and capacities at which bounds for the value function for the C,G&J method must be calculated.
- The way the lower bound is determined from the upper bound when using the C,G&J method.
- The value of ϵ for the You method and the adjusted B&P method with trade-up.
- The switch time, i.e., the time at which the C&H method switches from the approximation method to the exact method.

These values must all be determined such that they fit the respective methods as well as possible, i.e., such that an acceptable tradeoff between revenue and running time is made. Hence, these values must be tuned before any final results can be obtained.

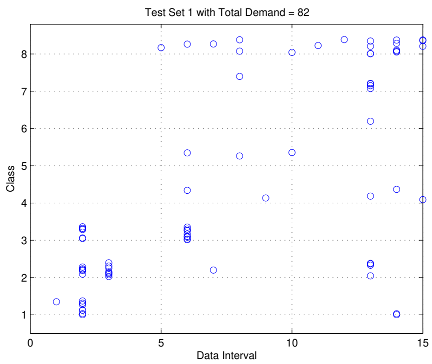
In the data given for the trade-up market there are 8 fare classes, 15 data intervals and the four capacities $C = 25$, $C = 40$, $C = 55$ and $C = 75$ are assumed to be the most realistic. These capacities are lower than the capacities used in the problem without trade-up, since the expected demand is much lower than in the problem without trade-up. Again different data sets of demand and fares are given by the airline and the total demand and the fares for the different classes are given in Table 5.6.

As for the problem without trade-up, the difference in the total demand for test set 1 and test set 4 is large. In Figure 5.6 the demand patterns for these two test sets are shown. In test set 1 there is a large demand, and this

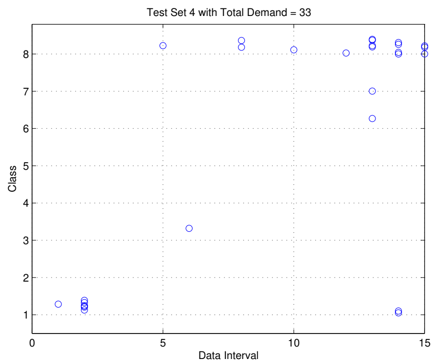
Test Set	Total Demand	Fares
1	82	[290, 210, 145, 130, 100, 65, 45, 30]
2	55	[290, 210, 145, 130, 100, 65, 45, 30]
3	50	[160, 110, 90, 70, 50, 35, 20, 15]
4	33	[270, 190, 140, 110, 90, 70, 60, 45]
5	57	[350, 290, 210, 160, 130, 95, 75, 50]

Table 5.6: Demand and Fares for Five Test Sets with Trade-Up.

is evenly distributed throughout the entire booking period such that in the beginning of the booking period the demand is mainly for low-fare classes and toward the end of the booking period, the demand is mainly for high-fare classes. The demand in test set 4 is fairly low and in the middle of the booking period it is close to zero. The demand patterns for test sets 2, 3 and 5 are given in Appendix B. These figures show that the demand patterns are very similar to each other and to the demand pattern for test set 1, just with a lower demand. All test sets differ, though, and hence these are assumed to be representative for real life data.



(a) Demand for Test Set 1.



(b) Demand for Test Set 4.

Figure 5.6: Demand Pattern for Test Sets 1 and 4 with Trade-Up.

Recall, as described in Chapter 4, that in a trade-up market both trade-up classes and independent classes exist. To simplify the problem, in this report it is chosen to ignore the independent classes when a trade-up market is considered. Hence, the optimizations are made for only approximately half the number of classes as the optimizations for the problem without trade-up. Therefore the time used for each optimization in the trade-up market must be approximately half of the time allowed for an optimization in the market without trade-up. Thus, the optmizations for the problem with trade-up must take less than approximately 0.45 seconds to be usable for BA.

In the following sections the parameter tuning and the results of the numerical experiments for the different methods are presented.

5.2.1 Parameter Tuning

As mentioned in the previous section the parameters which need to be tuned for this problem are the times and capacities at which bounds for $\Delta V_t(x)$ must be calculated and the way the lower bound is determined from the upper bound for the C,G&J method. For the You method and the adjusted B&P method with trade-up the values of ϵ need to be tuned, and finally the switch times for the C&H methods must be determined.

Tuning for the C,G&J method

If all the mentioned parameters have to be tuned thoroughly, the tuning for this problem is comprehensive. Hence, for the C,G&J method the times and capacities at which bounds for $\Delta V_t(x)$ must be calculated are chosen reasonably. There is a tradeoff between running time and revenue for the method, since it is assumed that the more times and capacities the bounds are calculated in, the higher is the revenue obtained with the method, but this also requires a longer running time. In this report it is chosen to use the beginning of each data interval as the times at which the bounds are calculated and the capacities at which the bounds are calculated are for each fifth value of the capacity. For instance if the total capacity is 25, then the bounds are calculated in the values $C = \{1, 5, 10, 15, 20, 25\}$.

Recall that when the C,G&J method is used, two different ways of determining the upper bound for $\Delta V_t(x)$ are suggested in Section 4.1.4. The first way is to use the B&P LP method with trade-up, and when this is used, the C,G&J method is denoted the B&P LP C,G&J method. The other way of calculating the upper bound is to use the adjusted B&P method with trade-up. Once the upper bound is determined this way, the C,G&J method is denoted the adjusted B&P C,G&J method.

When the upper bound has been calculated, the lower bound can be determined in a number of different ways. In this report it is chosen to calculate the lower bound as a function of the upper bound. The tuning of the lower bound is fairly perfunctory in this report, hence only three different ways of finding this are tested. These are

1. $\text{LB} = \text{UB} - K - r \cdot \text{UB}$

2. LB = UB − $r \cdot$ UB
3. LB = UB − K

where K is a constant, r determines which ratio of the upper bound, that should be subtracted this, UB is the upper bound for $\Delta V_t(x)$ and LB is the lower bound. By running a program calculating the lower bounds with different K and r it is found that apparently the best lower bound for both C,G&J methods is obtained by

$$\text{LB} = \text{UB} - 0.3 \cdot \text{UB}. \tag{5.1}$$

The number of different solution methods for the general SIC problem is extensive. Hence, to reduce this number only one of the two C,G&J methods is used. Therefore tests are run where the revenue which is obtained when using the adjusted B&P C,G&J method is compared with the revenue obtained with the B&P LP C,G&J method. The lower bound for both methods is calculated using (5.1) page 97. The results from these comparisons are not recorded, but from these it is concluded that higher revenues are obtained with the B&P LP C,G&J method. Due to the use of Matlabs `linprog.m` running times for this method are longer than for the adjusted B&P C,G&J method. As mentioned previously, though, time can be reduced significantly by using a more efficient optimizer and by implementing the program in another programming language. Hence, it is assumed that the running time for this method can be reduced enough to be feasible for BA, and therefore the B&P LP C,G&J method is used in all of the following.

For the problem with trade-up, it is assumed that the You method yields better results than the L&H method with revised demand, since trade-up is directly incorporated in the model by You. Therefore bounds for $\Delta V_t^Y(x)$, calculated with the value function in the You method given in (4.7) page 45, are desirable. To empirically determine if the B&P LP C,G&J method can be used to calculate an upper bound for $\Delta V_t^Y(X)$, the bound calculated with the B&P LP C,G&J method and $\Delta V_t^Y(x)$ are plotted together. In the two uppermost plots in Figure 5.7 the upper and lower bounds, UB and LB respectively, are plotted together with $\Delta V_t^Y(x)$ for capacity $C = 40$ and data intervals 5 and 15. The circles in the figure indicate for which values of remaining capacity there is a knot in the spline. In the figure it is seen that the upper bound fits nicely for data interval 5, but for data interval 15, the upper bound actually lies below $\Delta V_t^Y(x)$ between remaining capacities 20 and 25. This is caused by the interpolation with a spline. The interpolation is made such that the values of the upper bound in the knots are fitted in the best possible way, which may not necessarily yield an upper bound for

$\Delta V_t^Y(x)$. The upper bound calculated with the B&P LP C,G&J method is investigated for all five test sets and all four values of the total capacity. From this it is concluded that for all test sets the B&P LP C,G&J method provides an acceptable upper bound. Only for few values of remaining capacities and in few data intervals the bound is not actually an upper bound.

The lower bound in the topmost plots in Figure 5.7 are calculated using (5.1) with which the best results for the B&P LP C,G&J method are obtained. As can be seen in the figures, this lower bound is not really a very good lower bound. For both data intervals it lies above $\Delta V_t^Y(x)$ for several values of the remaining capacity. In the bottom two plots the lower bound has been calculated by

$$\mathrm{LB} = \mathrm{UB} - 100 - 0.3 \cdot \mathrm{UB}.$$

The plots are shown for data intervals 5 and 15 as well, and they show that when using this lower bound, $\Delta V_t^Y(x)$ is nicely surrounded by the upper and lower bounds. The revenues obtained when using this lower bound is lower than the revenue obtained when using (5.1), though, and since the tuning of the lower bound is made with respect to the highest revenue, (5.1) is used in all of the following, regardless that it is not a lower bound in all data intervals and for all test sets.

Tuning of ϵ

The only parameters, which remain to be tuned are the values of ϵ for the You method and adjusted B&P method with trade-up and the switch time for the C&H method. Since both the You method and the adjusted B&P method with trade-up are used as exact and approximation methods in the C&H method, the values of ϵ for these methods are tuned before the switch time is tuned. Recall that a smaller ϵ gives a smaller probability of two or more arrivals in one decision period, and therefore, the smaller ϵ is, the more accurate the method is, and presumably this yields a higher revenue, but a longer running time.

The You method and the adjusted B&P method with trade-up are run for different values of ϵ, an arrival process is simulated and the revenues obtained by the two methods are calculated. Four different values of ϵ are used for the You method. These are $\epsilon_1^Y = 0.001$, $\epsilon_2^Y = 0.01$, $\epsilon_3^Y = 0.1$ and $\epsilon_4^Y = 0.2$. Five different values of ϵ are used for the adjusted B&P method with trade-up. These are $\epsilon_1^{BP} = 0.1$, $\epsilon_2^{BP} = 0.2$, $\epsilon_3^{BP} = 0.3$, $\epsilon_4^{BP} = 0.4$ and $\epsilon_5^{BP} = 0.5$. These values of ϵ are chosen by looking at the results of the parameter tuning for the problem without trade-up.

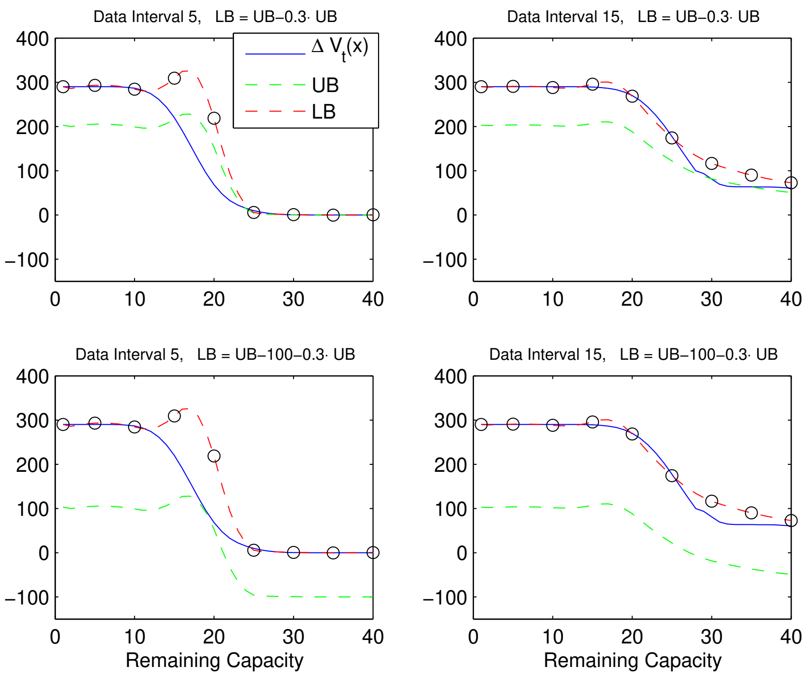


Figure 5.7: $\Delta V_t(x)$ with UB and LB for the General Problem.

First the tuning of ϵ for the You method is described. In Table D.5, Appendix D.2.1, the average running times when calculating the revenue using the You method are given. In the table the test set denoted You 1 is when the You method is used on test set 1, You 2 is with test set 2, etc. In Table D.6, Appendix D.2.1, the revenues obtained for 1000 test runs for the different values of ϵ are compared. The values in the table are the number of times one value of ϵ yields a higher and a smaller revenue than another ϵ. As for the problem without trade-up, when the capacity of the aircraft is larger than the expected total demand, very few requests are rejected by the decision rule. Hence, in this case the difference in revenue when using the method for different values of ϵ is insignificant. For instance, for a capacity of 55 for test set 4, where the expected total demand is 33, the number of times the revenue obtained with $\epsilon_1^Y = 0.001$ is greater than the revenue obtained with $\epsilon_4^Y = 0.2$ is 1 and the number of times the revenue obtained with ϵ_4^Y is greater than the revenue obtained with ϵ_1^Y is 0. This is the general tendency when the capacity is greater than the demand and therefore the results for the following capacities and test sets are omitted: For $C = 40$ the results for test set 4, for $C = 55$ the results for test set 3 and 4 and for $C = 75$ the results for test sets 2, 3, 4 and 5.

In Table D.5, Appendix D.2.1, as expected, the running times are seen to be decreasing with increasing values of ϵ. Hence, the optimal value of ϵ must be determined by investigating the values in Table D.6. Here the results obtained with $\epsilon_2^Y = 0.01$ are seen to be better than those obtained with $\epsilon_1^Y = 0.001$ in most instances. Thus, from these two values $\epsilon_2^Y = 0.01$ is chosen. For most test sets and capacities ϵ_2^Y yields slightly better results than ϵ_3^Y, but in some cases ϵ_3^Y is superior. In either instance the number of times one of the values yields better results than the other is very close to the number of times a worse result is obtained. Hence, from a running time perspective ϵ_3^Y is chosen. The question is then whether to use ϵ_3^Y or ϵ_4^Y. The parameter ϵ_4^Y yields faster running times but lower revenues, and thus $\epsilon_3^Y = 0.1$ is used for the general You method throughout the report. In Table D.5 in Appendix D.2.1, the running times for the You method are seen to be too long to be feasible for BA. It is chosen to use $\epsilon_3^Y = 0.1$ anyway, and this choice reflects the assumption that if the You method is implemented in another programming language, then running times can be reduced, such that a running time closer to the limit of 0.45 seconds can be obtained.

Next the tuning of ϵ for the adjusted B&P method with trade-up is presented. In Table D.7, Appendix D.2.1, the average running times when calculating the revenue using the adjusted B&P method with trade-up are given. For this method it is seen that a larger ϵ does not necessarily yield shorter running times. In Table D.8, Appendix D.2.1, the revenues obtained for the different values of ϵ are compared. Again, when the capacity of the aircraft is much larger than the expected total demand, the difference in revenue when using the method for different values of ϵ is insignificant. Therefore the results for the same capacities and test sets as for the You method are omitted. In Table D.8 it is seen that in general higher revenues are obtained when using a smaller ϵ. Then, to determine which value of ϵ is optimal for this problem, running times are considered. Table D.7 shows that in some cases especially for test set 1, the running times when using $\epsilon_1^{BP} = 0.1$ are much longer than those when using a higher ϵ. For $\epsilon_2^{BP} = 0.2$, though, running times are seen to be as short or only a little longer than for a higher value of ϵ. Therefore it is chosen to use $\epsilon_2^{BP} = 0.2$ for the adjusted B&P method with trade-up for the general problem in this report.

Tuning of the Switch Time

Finally, the switch time for the C&H method is tuned. The exact solution methods are assumed to yield higher revenues than the approximation methods. Therefore it is assumed that the sooner the switch is made from the approximation method to the exact method, the higher is the revenue obtained from a simulated arrival process. At the same time, though, the sooner the switch is made, the longer is the running time for the method. Hence, as in the tuning of ϵ there is a tradeoff between a high revenue and a long running time.

As described in Section 4.1.5 there are three different hybrid methods. These are the "Max" method, the "Add Column" method and the "Combi" method. The You method is the only exact method in this report for solving the general SIC problem with trade-up. For this method in the tuning of ϵ, it was seen that $\epsilon = 0.1$ gave good results with a reasonable running time. Two different methods are used as the approximation method which is applied in the beginning of the booking period. The first method is the adjusted B&P method with trade-up, which is assumed to yield nice results fast and this is used with $\epsilon = 0.2$ as determined in the tuning of ϵ. The second approximation method is the You method but with a larger ϵ, i.e., $\epsilon = 0.5$. With this value of ϵ there is a fairly large probability that more than one request occur in each decision period. Hence, the conditions in the You model are not very well satisfied, but since it is used as the approximation method this is acceptable. With one exact method and two approximation methods there is a total of six different hybrid methods. The possible switch times in the tuning are in the beginning of data intervals $12, 11, \ldots, 3, 2$. These switch times are chosen, since it does not make sense to switch earlier than after three data intervals, because if this is done, not much time is cut off the running time.

The average running times from using the hybrid methods on test set 1 with $C = 55$ and different switch times are given in the upper half of Table 5.7. In the lower half of this table it is shown how well the hybrid methods perform compared with the You solution method with $\epsilon = 0.1$, which has an average running time of 4.92s. It is seen, that as expected the running times for the hybrid methods are shorter than that of the You method. The values in the lower part of the table are the average revenues obtained when using the respective hybrid method a 1000 times divided by the average revenue obtained when using the You method a 1000 times.

The method "Max1" is the "Max" method with the adjusted B&P method with trade-up as the approximation method and "Max2" is the "Max" method with the You method as the approximation method. The same applies for the other methods.

In the table it is seen that, as expected, the later a switch is made from the exact method to the approximation method, the longer are the running times for the hybrid methods. It was also expected, though, that an early switch would give a higher revenue than a late switch, but this is not the

	Switch Time in Data Interval										
	2	3	4	5	6	7	8	9	10	11	12
Method	Average Running Time										
Max1	0.97 s	1.68 s	1.75 s	1.80 s	1.95 s	2.00 s	2.27 s	2.39 s	2.96 s	3.05 s	3.08 s
Max2	2.13 s	2.63 s	2.67 s	2.72 s	2.80 s	2.83 s	3.01 s	3.06 s	3.49 s	3.55 s	3.55 s
AddCol1	0.98 s	1.70 s	1.77 s	1.83 s	1.97 s	2.02 s	2.30 s	2.42 s	2.99 s	3.07 s	3.08 s
AddCol2	3.28 s	3.53 s	3.57 s	3.57 s	3.61 s	3.63 s	3.70 s	3.72 s	3.96 s	4.01 s	3.99 s
Combi1	0.93 s	1.65 s	1.75 s	1.80 s	1.94 s	2.00 s	2.28 s	2.40 s	3.01 s	3.11 s	3.12 s
Combi2	2.36 s	3.11 s	3.20 s	3.26 s	3.41 s	3.46 s	3.74 s	3.86 s	4.46 s	4.58 s	4.57 s
Method	Relative Revenue with respect to the Exact Method in Percentage										
Max1	0.96	0.98	0.97	0.98	0.97	0.98	0.98	1.00	0.99	0.99	0.98
Max2	0.99	1.02	1.02	1.02	1.03	1.02	1.02	1.03	1.01	1.00	1.00
AddCol1	0.85	0.77	0.77	0.78	0.79	0.80	0.87	0.88	0.95	0.96	0.95
AddCol2	0.84	0.89	0.91	0.92	0.95	0.95	0.99	1.01	0.98	0.97	0.98
Combi1	0.97	0.99	0.99	1.00	1.00	1.01	1.02	1.02	1.02	1.02	1.02
Combi2	0.84	0.84	0.84	0.83	0.84	0.83	0.84	0.84	0.88	0.88	0.89

Table 5.7: Tuning of the Switch Time for the Hybrid Methods

case for the methods “Max1”, “Max2” and “AddCol1”. The reason for this may be that when combining two methods, one method may be good in the beginning of the booking period and worse later on in the booking period and the other method may have the opposite pattern. Then nothing can be concluded regarding how the hybrid method will perform when the two methods are combined. The switch times for the different hybrid methods need not be the same, thus, a switch time is determined for each of the methods. The clearest example of how these switch times are obtained is for “AddCol2”. In Table 5.7 for “AddCol2”,the relative revenue is seen to increase from 0.84% to 0.89% when increasing the switch time from data interval 2 to data interval 3. The running time by doing this is increased from 3.28 seconds to 3.53 seconds. Hence, an increase in switch time from data interval 2 to data interval 3 is considered to be reasonable. By increasing the switch time to data interval 6, it is seen that the running time does not increase much but the relative revenue increases. Therefore the switch time for “AddCol2” is chosen to be data interval 6. The switch times for the other methods are determined similarly. This yields the following switch times for the six hybrid methods

$$\begin{aligned}
\text{“Max1”: } & \text{switch time} = 2 \\
\text{“Max2”: } & \text{switch time} = 3 \\
\text{“AddCol1”: } & \text{switch time} = 2 \\
\text{“AddCol2”: } & \text{switch time} = 6 \\
\text{“Combi1”: } & \text{switch time} = 2 \\
\text{“Combi2”: } & \text{switch time} = 2.
\end{aligned} \tag{5.2}$$

These switch times are used for the general SIC problem with trade-up.

Summary of Parameter Tuning

For the C,G&J method the times and capacities are chosen for which the bounds for $\Delta V_t(x)$ are calculated. The times are set to be the beginning of each data interval and the capacities are chosen to be $C = \{1, 5, 10, 15, 20, 25\}$, for instance, when the capacity of the aircraft is 25. The optimal lower bounds for both C,G&J methods are tuned to

$$\mathrm{LB} = \mathrm{UB} - 0.3 \cdot \mathrm{UB}.$$

The two C,G&J methods are compared and the best of these, the B&P LP C,G&J method, is used for all results.

To sum up the tuning of ϵ, for the You method a value of $\epsilon^Y = 0.1$ is chosen. This choice reflects the assumption that the running time is reduced when implementing the method in another programming language, since the running time for the You method with this ϵ is not less than 0.45 s. For the adjusted B&P method the value $\epsilon^{BP} = 0.2$ is chosen.

Finally, the switch times for the C&H methods are tuned. A switch time for each of the six hybrid methods is determined and the results of this tuning are given in the items in (5.2).

5.2.2 Results

There is quite a large number of different solution methods for the general problem with trade-up. These are the EMSRb method with trade-up, the You method, the adjusted B&P method with trade-up, the B&P LP method with trade-up, the B&P LP C,G&J method and six different hybrid methods. To make the number of comparisons of methods manageable the six hybrid methods are compared first and then only the best one of these is used for the results.

In Appendix F.1 the results from comparing the hybrid methods are shown in five tables, one for each test set. The data in the tables are the differences between the revenues obtained by the two methods which are compared, divided by the smallest of these revenues. These data are to be interpreted as described on page 90.

The results show that there is not one method, which is better than all others at all times. Which method is best varies both with capacity and test set. In general, though, the method "Max2" yields good results and for almost all test sets and capacities this method is in the top two of the methods. Hence, this method is used in all of the following.

After this reduction in the number of methods, six different methods for the general SIC problem with trade-up are left. These methods are listed below, where the emphasized name is the notation of the method in the tables of results.

- *EMSRb* method with trade-up.
- *You* method with $\epsilon^Y = 0.1$.
- Adjusted *B&P* method with trade-up and $\epsilon^{BP} = 0.2$.
- *B&P LP* method with trade-up.
- B&PLP *C,G&J* method.
- "Max2" from the *C&H* method.

The You method and the B&P method with trade-up are updated in each decision period, i.e., decision rules are calculated before the beginning of the booking period for all decision periods. The booking limits in the EMSRb method with trade-up are revised at specific times stated by the airline. These times are given as the *DecPer* vector in Appendix E. In the C,G&J method the value function is calculated at predetermined remaining capacities and times. The times used for this method are the beginning of each data interval. These times are given as the vector *DataintStart* in Appendix E.

Comparison of the Methods

The tables of results for the comparisons of the methods are shown only for test sets 1 and 4, since these are the test sets which differ the most. The results for test sets 2, 3 and 5 are given in Appendix F.2.

In Table 5.8 the average running times for the different methods are shown for test sets 1 and 4. In the table the slowest method is by far seen to be the B&P LP, which was expected. All methods except for the EMSRb method with trade-up are faster for smaller capacities and the fastest methods are the B&P and EMSRb methods with trade-up.

As for the problem without trade-up, to determine if one method yields a significantly higher revenue than another, a paired t-test with a significance level of $\alpha = 5\%$ is used in the exact same way as described for the problem without trade-up in Section 5.1.2. The Tables 5.9 and 5.10 show results for test sets 1 and 4, respectively. The values in the tables are the relative differences in percentage and the tables are to be read as explained in the items on page 90.

	$C = 25$	$C = 40$	$C = 55$	$C = 75$
EMSRb 1	0.35 s	0.40 s	0.43 s	0.39 s
You 1	2.21 s	3.47 s	4.65 s	6.50 s
B&P 1	0.11 s	0.14 s	0.18 s	0.23 s
B&P LP 1	15.36 s	24.12 s	32.87 s	45.41 s
C,G&J 1	3.93 s	5.64 s	7.47 s	9.92 s
C&H 1	1.28 s	1.98 s	2.65 s	3.62 s
EMSRb 4	0.16 s	0.13 s	0.15 s	0.16 s
You 4	0.91 s	1.44 s	1.98 s	2.70 s
B&P 4	0.08 s	0.09 s	0.11 s	0.14 s
B&P LP 4	13.16 s	21.06 s	29.35 s	39.46 s
C,G&J 4	3.37 s	4.94 s	6.56 s	8.58 s
C&H 4	0.68 s	1.02 s	1.37 s	1.83 s

Table 5.8: Running Times for the General Methods.

	You	**B&P**	**B&P LP**	**C,G&J**	**C&H**
C=25					
EMSRb	**34.70**	**33.55**	**33.94**	**34.47**	**34.56**
You	-	1.17	0.73	0.21	0.10
B&P	-	-	**0.42**	**0.97**	**1.07**
B&P LP	-	-	-	**0.54**	**0.63**
C,G&J	-	-	-	-	**0.11**
C=40					
EMSRb	**15.40**	**10.45**	**14.10**	**20.43**	**14.05**
You	-	6.69	2.46	**3.90**	1.20
B&P	-	-	**3.99**	**10.07**	**5.54**
B&P LP	-	-	-	**5.85**	0
C,G&J	-	-	-	-	5.18
C=55					
EMSRb	2.68	8.00	5.31	**0.51**	0.63
You	-	5.51	2.79	**3.03**	**2.01**
B&P	-	-	**2.61**	**8.39**	**7.38**
B&P LP	-	-	-	**5.67**	**4.68**
C,G&J	-	-	-	-	1.01
C=75					
EMSRb	**3.59**	3.03	1.94	**4.48**	**2.82**
You	-	6.33	5.19	**1.09**	0
B&P	-	-	**1.13**	**7.37**	**5.83**
B&P LP	-	-	-	**6.19**	**4.71**
C,G&J	-	-	-	-	1.60

Table 5.9: Rel. Diff. in % for Test Set 1 for the General Problem.

To get a general picture of which methods are the best for this problem it is necessary to consider all five tables of results. When considering Tables 5.9 and 5.10 it is seen that the three methods B&PLP, EMSRb with trade-up and B&P with trade-up all have poor performance. This is a general tendency when considering all five tables of results. The B&P LP method is the best of the three methods, but for all test sets this method is only in the top three for five different capacities. In general the C&H method performs

	You	**B&P**	**B&P LP**	**C,G&J**	**C&H**
C=25					
EMSRb	3.00	6.81	3.05	**1.20**	2.95
You	-	0	**0.79**	**4.17**	**0.07**
B&P	-	-	**3.52**	**7.53**	0
B&P LP	-	-	-	**3.86**	0.67
C,G&J	-	-	-	-	4.06
C=40					
EMSRb	**5.32**	4.79	3.88	**5.32**	**4.94**
You	-	9.79	8.84	0	0
B&P	-	-	**0.88**	**9.79**	**9.53**
B&P LP	-	-	-	**8.84**	**8.59**
C,G&J	-	-	-	-	0
C=55					
EMSRb	**6.58**	3.37	3.36	**6.58**	**5.69**
You	-	9.74	9.73	NoD	0.82
B&P	-	-	0	**9.74**	**8.88**
B&P LP	-	-	-	**9.73**	**8.88**
C,G&J	-	-	-	-	0.82
C=75					
EMSRb	**6.54**	2.97	2.97	**6.54**	**5.67**
You	-	9.35	9.35	NoD	0.79
B&P	-	-	NoD	**9.35**	**8.54**
B&P LP	-	-	-	**9.35**	**8.54**
C,G&J	-	-	-	-	0.79

Table 5.10: Rel. Diff. in % for Test Set 4 for the General Problem.

well for some test sets and capacities, but for instance for test set 2 seen in Table (F.7), Appendix F.2, its performance is actually worse than that of all other methods except for the EMSRb method.

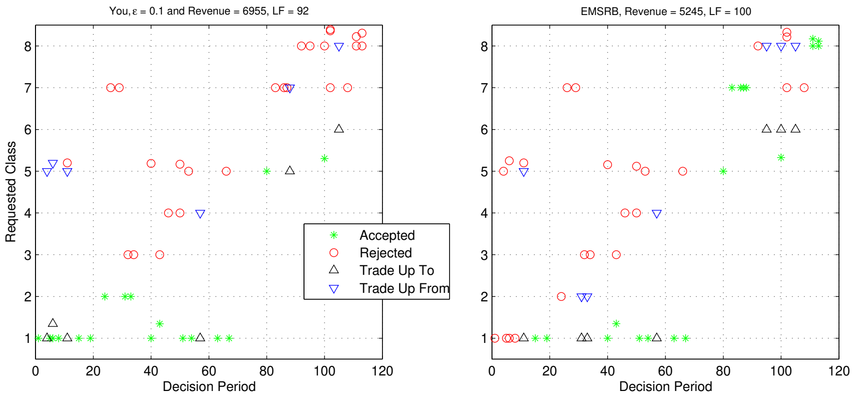


Figure 5.8: Accept/Reject Decision for General Problem.

For almost all test sets and capacities the You and C,G&J methods are superior to all other methods. These are at all times in the top three of the methods. When considering the running times as well as the comparison of revenues, the running times for both methods are seen to be too long for the methods to be feasible. Hence, if the running times for these methods can be reduced to be close to the running times for, for instance, the B&P method, either of these methods is recommended. The question is then which one of these methods should be used. It is difficult to draw any final conclusions as to which one of these methods is the best. The running time for the C,G&J method is longer than that of the You method. As described previously, though, by using for instance GAMS instead of Matlabs `linprog.m` for solving the LPs in the C,G&J method, this method can get approximately 10 times faster. For both methods running times can be reduced by implementing the methods in a higher-level programming language than Matlab.

To get an idea of why one method yields better results than another, Figure 5.8 is included to show the acceptance/rejection decision for an entire booking period for the You method and the EMSRb method with trade-up. The capacity of the aircraft is $C = 25$ and test set 4 is used for the simulation. These methods, test set and total capacity are chosen, since in this case the You method yields a revenue which is averagely 30% higher than that of the EMSRb method. This is the highest difference between any two methods for the general problem. The figure illustrates in which decision period each request occurs, which class is requested and whether each request is accepted or rejected. Furthermore, if a request is rejected, it is seen whether the rejected passenger chooses to trade up, and if so, which class the passenger trades up to. Recall that decision period 1 is closest to departure. A comparison of the decisions made by the You and EMSRb methods show that in the beginning of the booking period, the EMSRb method keeps class 8 open until decision period 110 whereas the You method keeps class 8 closed. Between decision periods 90 and 80 the EMSRb method opens class 7 and hence requests for this class are accepted. The You method closes class 7 and therefore some requests for this class are rejected but one of the rejected passengers chooses to trade-up. In the end of the booking period, where requests are mainly for class 1, the load factor (LF) when using the EMSRb method is 100%, and hence several class 1 requests are rejected. The You method still has remaining capacity and keeps only class 1 open. Therefore all class 1 requests are accepted and passengers requesting class 5 even trades up to class 1. With these acceptance/rejection decisions it makes sense, that the You method performs considerably better than the EMSRb method with trade-up.

Comparison of TU Methods with Non-TU Methods

Now, in a market where trade-up occurs it is interesting to determine whether it is more profitable for the airline to use the methods which incorporate trade-up or the methods which do not incorporate trade-up. Obviously it is expected that the methods which do incorporate trade-up are superior, since these methods take customer behaviour into account. To see if this assumption is correct, it is chosen to compare the best two methods from the problem without trade-up with the best two methods for the problem with trade-up.

In Section 5.1.2 the results for the SIC problem without trade-up were described. In this section it was concluded that if sufficient time is available, it is preferable to use the L&H method with $\epsilon = 0.01$, since this method yields the highest revenues. The second best methods are the EMSRb and B&P methods without trade-up, but it could not be determined which of these methods was the best. It was recommended that the airline should use the EMSRb method.

Thus the four methods used in the comparison are the L&H and the EMSRb methods for the problem without trade-up and the You and C,G&J methods for the general problem with trade-up. The first methods are in the following denoted L&H and EMSRb respectively, and the latter methods are denoted You and C,G&J, respectively.

The demand input to the two methods without trade-up is a matrix, with demand for each of the 8 classes in the trade-up market. This matrix is similar to the matrix used in the problem without trade-up, but it is only for the 8 trade-up classes and not for all 16 independent classes. Then the decision rules are made using these two methods exactly as for the problem without trade-up. Since trade-up is customer behaviour and this occurs in the market, the simulation used when calculating the expected revenue, which can be obtained by the two methods, is the simulation for the trade-up market. The two trade-up methods are used in the same way as in the problem with trade-up.

Tables 5.11 and 5.12 show the relative differences in revenue in percentage for test set 1 and 4, respectively. The remaining tables for test sets 2, 3 and 5 are given in Appendix F.3, and again to get a full picture of the comparisons of the methods, all tables should be considered. As explained for the results when comparing the six methods for the problem with trade-up earlier in this section, the numbers in the tables are the relative differences in percentage and the values are to be read as explained in the items on page 90. For these tests the tables have been reduced a little, since it is not necessary to compare a method with all other methods, i.e., it is not necessary to compare

	L&H	EMSRb
C=25		
You	8.76	10.43
C,G&J	8.60	10.27
C=40		
You	9.49	10.02
C,G&J	23.23	13.80
C=55		
You	8.68	8.97
C,G&J	11.70	12.01
C=75		
You	7.76	7.49
C,G&J	9.03	8.76

Table 5.11: Rel. Diff. in % when Comparing Non-TU with TU Methods, Test Set 1.

	L&H	EMSRb
C=25		
You	6.11	6.36
C,G&J	9.85	9.88
C=40		
You	10.17	10.20
C,G&J	10.18	10.21
C=55		
You	9.85	9.85
C,G&J	9.85	9.85
C=75		
You	9.57	9.57
C,G&J	9.57	9.57

Table 5.12: Rel. Diff. in % when Comparing Non-TU with TU Methods, Test Set 4.

the L&H method with the EMSRb method for the problem without trade-up since these have already been compared in a previous section. Furthermore, the You and C,G&J methods with trade-up for the general problem have already been compared.

In the tables it is seen that for all test sets and all capacities the trade-up methods perform better than the non-trade-up methods. Especially for small capacities the performance of the methods with trade-up is a lot better than that of the two other methods. The amount the trade-up methods are better than the methods for the problem without trade-up decreases as the capacity increases. Hence, it can be concluded that an average increase in revenue of approximately 7% 8% can be obtained when using the methods, which are specifically derived for a trade-up market, instead of the best methods for the problem without trade-up.

Summary of Results

Initially in this section the number of solution methods for the general SIC problem is reduced to six methods. Only one out of six C&H methods, the "Max2" hybrid method, is used in the final results and only one of two C,G&J methods, the B&PLP C,G&J method, is used.

	Capacity	**EMSRb 2**	**B&P 2**	**B&P LP 2**	**C,G&J 2**	**C&H 2**
You 2	$C = 25$	13.45	7.17	8.06	2.57	0
You 2	$C = 40$	5.20	4.18	3.31	1.01	1.53
You 2	$C = 55$	10.99	2.28	2.25	0	3.75
You 2	$C = 75$	13.57	1.59	1.58	0.59	5.89
	Capacity	**EMSRb 3**	**You 3**	**B&P 3**	**B&P LP 3**	**C&H 3**
C,G&J 3	$C = 25$	4.20	0	10.89	9.13	0
C,G&J 3	$C = 40$	3.82	0.81	8.88	7.24	0.09
C,G&J 3	$C = 55$	9.40	**0.07**	7.11	6.70	3.69
C,G&J 3	$C = 75$	9.73	NoD	6.57	6.57	4.28

Table 5.13: Summary of results for the General SIC Problem.

Table 5.13 sums up the main results from this section. The results are for the You method with test set 2 and the C,G&J method with test set 3. These are the test sets, which most clearly show the general tendency for these two methods and the complete tables of results for these two test sets can be seen in Appendix F.2. The conclusion is that the You method and the C,G&J method yield the highest revenues compared with all other methods used for solving the general problem with trade-up. Hence, one of these two methods should be used. The running times for both methods are too long to be feasible for BA as they are right now. It is expected, though, that if an efficient optimizer is used to solve the LPs in the C,G&J method and if both methods are implemented in another programming language than Matlab, running times can be reduced sufficiently.

Finally, the best solution methods for the SIC problem without trade-up are applied in a trade-up market. The results obtained with these methods are compared with the results obtained with the trade-up methods in a trade-up market. As expected the trade-up methods perform better than the methods without trade-up for all test sets and all capacities. Depending on the test set and capacity an average increase in revenue of approximately 7% to 8% can be obtained when using the trade-up methods instead of the methods for the problem without trade-up.

5.3 Simplified SIC with Trade-Up

The test sets for the simplified SIC problem with trade-up are the same as for the general SIC problem with trade-up. I.e., the capacities which are used in the experiments are $C = 25$, $C = 40$, $C = 55$ and $C = 75$.

Recall, that for the simplified SIC problem with trade-up, the buying conditions for different fare classes are assumed not to differ. Hence, a passenger will always request the lowest class in the market, class k, and requests for other classes only occur through trade-up from class k. Therefore the data needed when solving the SIC problem with trade-up using the simplified methods described in Section 4.2 are a little different than the data needed for the methods used to solve the general SIC problem with trade-up. The difference occurs in the demand forecasts, since now these only need to be for class k for each data interval.

5.3.1 Parameter Tuning

For the simplified problem there are also data, which are specific for the different methods. These are

- The way the lower bound is determined from the upper bound when using the simplified C,G&J method.
- The value of ϵ for the simplified You method, the simplified B&P method with trade-up and the HM method.
- The switch time, i.e., the time at which the simplified C&H method switches from the approximation method to the exact method.

The times and capacities at which bounds for the value function for the simplified C,G&J method are calculated are the same as for the general problem for the same reasons as described in Section 5.2.1. The values listed in the items must be tuned before any final results can be obtained.

Tuning for the Simplified C,G&J Method

The first parameter which is tuned is the calculation of the lower bound in the simplified C,G&J method. The upper bound can be calculated in two ways. Either by using the simplified B&P LP method with trade-up, denoted Simple B&P LP, or by using the simplified adjusted B&P method with trade-up, denoted Simple B&P. Due to the large number of methods for the simplified problem, it is chosen to have only one simplified C,G&J method and one simplified B&P method with trade-up. Therefore the two simplified

B&P methods are compared to see which one is the best. The results are not included, but the methods alternately give the highest revenue. Simple B&P LP often yields the highest revenue, but in defiance of this, the method seems very unstable. Due to this and the fact that Simple B&P is much faster than Simple B&P LP, it is chosen to proceed with only Simple B&P. Therefore only Simple B&P is used to calculate the upper bounds in the C,G&J method.

Recall that the simplified adjusted B&P method with trade-up is only known to be a good upper bound to $\Delta V_t^{LH}(x)$ from the L&H method with expected demand for all classes calculated by (4.24) page 60. As for the general problem, the simplified You method is expected to yield better results than the L&H method with revised demand. Hence, bounds for $\Delta V_t^Y(x)$ calculated with the value function in the simplified You method given in (4.20) page 58 are desirable. It is empirically determined if the simplified C,G&J method using the simplified adjusted B&P method with trade-up can be used to calculate an upper bound for $\Delta V_t^Y(X)$. This is done by for all five test sets and four capacities plotting the upper bound obtained with the simplified C,G&J method together with $\Delta V_t^Y(x)$. The figures are not included but in most instances it is actually an upper bound, and hence it is reasonable to proceed with using the simplified adjusted B&P method for calculating the bounds in the simplified C,G&J method.

The lower bounds which are considered in the tuning are the same as for the general SIC problem, i.e.,

1. LB $=$ UB $-K - r\cdot$ UB
2. LB $=$ UB $-r\cdot$ UB
3. LB $=$ UB $-K$

where K is a constant, r determines which ratio of the upper bound, that should be subtracted from the upper bound. By testing the different lower bounds with different values of K and r, it is determined that the lower bound given in item 2 yields the highest revenue for the simplified C,G&J method. The value of r which yields the best results is 0.7, hence the lower bound is given by

$$\text{LB} = \text{UB} - 0.7 \cdot \text{UB}. \tag{5.3}$$

Figure 5.9 shows $\Delta V_t^Y(x)$ when using the simplified You solution method and the upper and lower bounds obtained with the simplified C,G&J method on test set 1 with $C = 40$ for data intervals 5 and 15. In the two topmost plots the lower bounds are calculated with (5.3) and in the bottom two plots

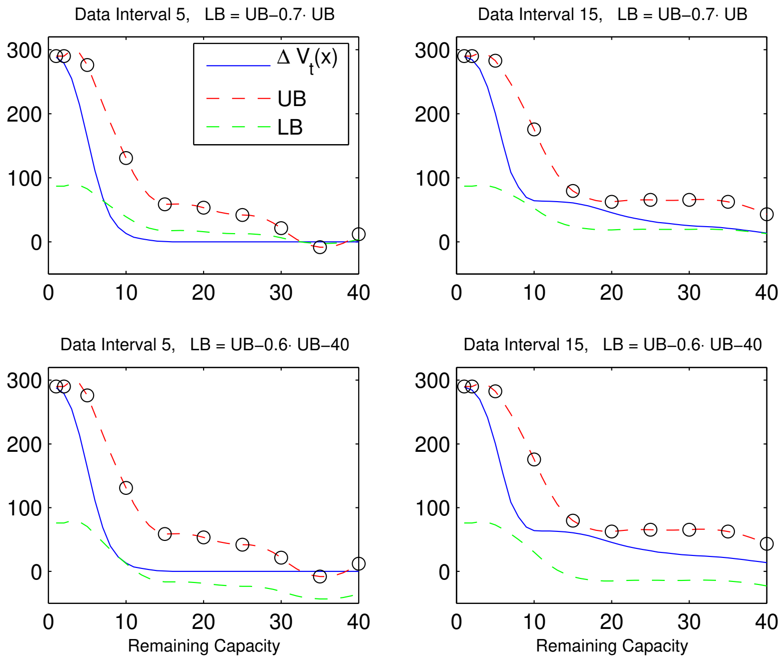


Figure 5.9: $\Delta V_t^Y(x)$ with UB and LB for the Simplified Problem.

the lower bound is calculated by

$$\mathrm{LB} = \mathrm{UB} - 40 - 0.6 \cdot \mathrm{UB}.$$

The first way of calculating the lower bound yields the highest revenue, since this is the optimal tuning of the bound. For data interval 15 the bounds surround $\Delta V_t^Y(x)$ nicely, whereas for data interval 5 the lower bound is not actually a lower bound. In the bottom two figures the second way of calculating the lower bound is seen to give a nice bound. Both for data interval 5 and 15 the bounds are seen to surround $\Delta V_t^Y(x)$. This way of calculating the bound gives lower revenues, though, and hence the lower bound calculated with (5.3) is used in the C,G&J method. This is done regardless that it is not a lower bound for all data intervals and all test sets.

As for the general problem, in Figure 5.9 for data interval 5 it is seen that using a spline to approximate the upper bound gives certain oscillations in the bound, resulting in the upper bound going slightly below the value of $\Delta V_t^Y(x)$.

Tuning of ϵ

The next parameters to be tuned are the values of ϵ for the simplified You method, the simplified adjusted B&P method with trade-up and the HM method.

For the simplified You method the same four values of ϵ are used as for the general You method, i.e., $\epsilon_1^Y = 0.001$, $\epsilon_2^Y = 0.01$, $\epsilon_3^Y = 0.1$ and $\epsilon_4^Y = 0.2$. Similarly, for the simplified adjusted B&P method with trade-up, the same five values of ϵ are used as for the general B&P method with trade-up, i.e., $\epsilon_1^{BP} = 0.1$, $\epsilon_2^{BP} = 0.2$, $\epsilon_3^{BP} = 0.3$, $\epsilon_4^{BP} = 0.4$ and $\epsilon_5^{BP} = 0.5$. Finally for the HM method four different values of ϵ are used and these are $\epsilon_1^{HM} = 0.2$, $\epsilon_2^{HM} = 0.3$, $\epsilon_3^{HM} = 0.4$ and $\epsilon_4^{HM} = 0.5$. These values are relatively large, but this is due to the assumption in the method, that several arrivals can occur in each decision period.

Table D.9, Appendix D.2.2, shows the average running times when using the simplified You method. As for the general problem, the test set denoted You 1 is when the simplified You method is used on test set 1, You 2 is with test set 2, etc. In Table D.10, Appendix D.2.2, the revenues obtained with the different values of ϵ are compared. Note that the results are shown for the same capacities and test sets as for the general problem with trade-up due to the same reasons. In Table D.9 the running times are seen to be decreasing with increasing values of ϵ. Hence, the optimal value of ϵ is found by investigating the comparisons given in Table D.10. In most instances the value $\epsilon_1^Y = 0.001$ yields the best results but sometimes the results obtained with $\epsilon_2^Y = 0.01$ are just as good or better. The simplified You method using ϵ_2^Y is 3 to 4 times faster than when using ϵ_1^Y. Since the running times when $\epsilon_2^Y = 0.01$ is used are reasonable and since when using ϵ_3^Y or ϵ_4^Y the revenues obtained are lower, ϵ_2^Y is chosen for the simplified You method.

Table D.13, Appendix D.2.2, shows the average running times when using the simplified adjusted B&P method. As for the general problem it is seen that a larger ϵ does not necessarily yield shorter running times. In Table D.14, Appendix D.2.2, the revenues obtained with the different values of ϵ are compared. Here the same tendency regarding the revenue and values of ϵ is seen as for the general problem, i.e., higher revenues are obtained using a smaller value of ϵ. In Table D.13 the running times using $\epsilon_1^{sBP} = 0.1$ are seen to be fairly short and therefore it is chosen to use $\epsilon_1^{sBP} = 0.1$ for the simplified adjusted B&P method.

In Table D.11, Appendix D.2.2, the average running times when using the HM method are given. As for the simplified adjusted B&P method it is seen that a larger value of ϵ does not necessarily yield shorter running times. It is seen, though, that using $\epsilon_1^{HM} = 0.2$ yields a longer running time than using $\epsilon_4^{HM} = 0.5$. In Table D.12, Appendix D.2.2, the revenues obtained with different values of ϵ are compared. The values in the table are seen to be much smaller than for the two other methods. Hence, the value of ϵ does not have a big influence on the results. This makes sense, since the method takes the variable number of requests which can arrive in each decision period into account. From the comparisons it is seen, though, that in most cases the value $\epsilon_4^{HM} = 0.5$ yields the best results. This is also the value of ϵ which in general gives the shortest running times and therefore $\epsilon_4^{HM} = 0.5$ is used for the HM method.

Tuning of the Switch Time

Finally, the switch time for the simplified C&H method is tuned. There are four different hybrid methods, i.e, "Max1","Max2", "Add Column" and "Combi", where "Max1" uses the decision rule $\Delta V_{t-1}(x) \leq F$ to determine the lowest open classes in the approximation period and "Max2" uses the decision rule by You given in (4.21) page 58 to determine the lowest open classes.

Both the HM method and the simplified You method are used as the exact method in the simplified C&H method. There are two candidates for the approximation method, namely the simplified adjusted B&P method with trade-up and the simplified You method with $\epsilon = 0.5$. Each of these approximation methods yields eight different hybrid methods and since a comparison of sixteen methods is comprehensive, it is chosen to use the approximation method which yields the best results. Therefore the revenues obtained with the simplified adjusted B&P method with trade-up is compared with those obtained with the simplified You method with $\epsilon = 0.5$ for all test sets and capacities. The results are not included, but it was seen that the simplified You method with $\epsilon = 0.5$ gave the best results and was reasonably fast. Thus, the only approximation method which is used in the simplified C&H method is the simplified You method with $\epsilon = 0.5$. In total this gives eight different hybrid methods. The possible switch times in the tuning are in the beginning of data intervals $12, 11, \ldots, 3, 2$.

The average running times from running the hybrid methods on test set 1 with $C = 55$ and different switch times are shown in the top half of Table 5.14. In the lower half of this table it is seen how well the hybrid methods perform compared with the simplified You solution method with $\epsilon = 0.01$ or with the HM method with $\epsilon = 0.5$, depending on which of these two methods that is used as the exact method. The simplified You method has an average running time of 1.33s and the HM method has an average running time of 1.27s. The values in the lower part of the table are the average revenue obtained when using the respective hybrid method 1000 times divided by the average revenue obtained when using the exact method 1000 times. The first four methods listed in the table use the simplified You solution method as the exact method, whereas the last four methods use the HM method as the exact method.

	Switch Time in Data Interval										
	2	3	4	5	6	7	8	9	10	11	12
Method	Average Running Time										
Max1	0.41 s	0.68 s	0.77 s	0.70 s	0.78 s	0.75 s	0.88 s	0.91 s	1.12 s	1.15 s	1.11 s
Max2	0.41 s	0.64 s	0.78 s	0.70 s	0.75 s	0.75 s	0.84 s	0.88 s	1.06 s	1.15 s	1.09 s
AddCol	0.47 s	0.73 s	0.76 s	0.79 s	0.82 s	0.83 s	0.94 s	0.90 s	1.07 s	1.15 s	1.18 s
Combi	0.40 s	0.74 s	0.80 s	0.74 s	0.83 s	0.84 s	0.88 s	0.98 s	1.15 s	1.21 s	1.25 s
Max1HM	0.47 s	0.71 s	0.65 s	0.75 s	0.82 s	0.84 s	0.88 s	0.97 s	1.16 s	1.19 s	1.20 s
Max2HM	0.43 s	0.65 s	0.75 s	0.69 s	0.75 s	0.79 s	0.92 s	1.01 s	1.05 s	1.22 s	1.18 s
AddColHM	0.49 s	0.71 s	0.78 s	0.80 s	0.84 s	0.90 s	0.96 s	0.99 s	1.24 s	1.10 s	1.24 s
CombiHM	0.44 s	0.66 s	0.73 s	0.77 s	0.84 s	0.90 s	0.98 s	0.99 s	1.22 s	1.32 s	1.24 s
Method	Relative Revenue with respect to the Exact Method in Percentage										
Max1	83.0	86.8	87.9	87.9	93.9	95.0	94.6	95.0	98.8	98.6	99.1
Max2	94.1	89.4	89.8	89.5	91.3	94.0	94.6	95.4	98.3	98.9	99.2
AddCol	96.5	99.1	98.1	98.7	98.9	99.0	99.3	99.5	99.7	99.8	99.7
Combi	93.5	94.2	93.4	95.1	96.6	97.0	98.5	99.3	99.7	99.8	99.7
Max1HM	81.6	85.8	87.0	87.0	93.0	93.8	93.7	94.1	98.4	98.3	98.6
Max2HM	91.8	88.2	88.7	88.3	90.0	92.6	93.7	94.1	97.7	98.3	98.4
AddColHM	96.9	98.9	98.6	98.8	99.4	99.1	99.0	99.5	99.8	99.7	99.6
CombiHM	93.0	93.9	93.3	94.9	96.6	96.6	98.1	99.2	99.7	99.7	99.5

Table 5.14: Tuning of the Switch Time for the Simplified Hybrid Methods.

In the table it is in general seen that the later a switch is performed from the exact method to the approximation method, the longer is the running time for the hybrid method. Generally, when combining the running time and the revenue obtained by the methods, the best results are seen to be when the switch time is in data intervals 6 or 7. Again, a switch time is determined for each of the eight methods. This is done in a similar way as for the general problem, i.e., by looking at the gain in revenue compared with the extra running time when using the exact method in an additional data interval. The switch times for the simplified C&H methods are chosen to be

$$\begin{aligned}
\text{“Max1”}&: \text{switch time} = 7\\
\text{“Max2”}&: \text{switch time} = 7\\
\text{“AddCol”}&: \text{switch time} = 7\\
\text{“Combi”}&: \text{switch time} = 8\\
\text{“Max1HM”}&: \text{switch time} = 7 \qquad (5.4)\\
\text{“Max2HM”}&: \text{switch time} = 7\\
\text{“AddColHM”}&: \text{switch time} = 6\\
\text{“CombiHM”}&: \text{switch time} = 6.
\end{aligned}$$

These switch times are used for the hybrid methods for the simplified SIC problem with trade-up.

Summary of Parameter Tuning

The times for which the bounds for $\Delta V_t(x)$ are calculated are chosen to be the beginning of each data interval, when using the C,G&J method. Furthermore the capacities are chosen to be every fifth capacity, for instance when the capacity of the aircraft is 25 then $C = \{1, 5, 10, 15, 20, 25\}$. The optimal lower bound for the C,G&J method is tuned to

$$\text{LB} = \text{UB} - 0.7 \cdot \text{UB}.$$

To sum up the tuning of ϵ, for the simplified You method a value of $\epsilon^Y = 0.01$ is chosen. For the simplified adjusted B&P method the value $\epsilon^{BP} = 0.1$ is chosen. Finally, for the HM method the value of ϵ does not have big influence on the results obtained using this method, hence, $\epsilon^{HM} = 0.5$ is chosen.

The switch time for the C&H methods are tuned for each of the eight hybrid methods. The results of this tuning are given in the items in (5.4).

5.3.2 Results

As for the general problem with trade-up there is a large number of different solution methods for the simplified problem. These are the simplified You method, the simplified adjusted B&P method with trade-up, the C,G&J method and eight different hybrid methods. Therefore, for the simplified problem it is also chosen to compare the eight hybrid methods first, such that the number of methods to be compared in the final results is reduced.

In Appendix G.1 the results from comparing the hybrid methods are shown in five tables, one for each test set. The data in the tables are the differences between the revenues of the two methods, which are compared, divided by the smallest of these two revenues. Again the data are to be interpreted as described on page 90. The results show that two methods alternately give the highest revenues. In general the method "Max2" yields better results than all other methods for small capacities and "AddColHM" gives the best results for larger capacities. Furthermore, the methods "Max2" and "Max2HM" are seen to give similar results, but in some cases "Max2" is slightly better. It is chosen to proceed with both of the methods "Max2" and "AddColHM" for the simplified C&H method, even though "AddColHM" in most cases gives the best results. This is done, since for the small capacities "Max2" is much better than "AddColHM".

After this reduction in the number of methods there are a total of six different methods for the simplified SIC problem with trade-up. These methods are listed in the following, and again the emphasized word is the notation of the method in the tables of results.

- Simplified *You* method with $\epsilon = 0.01$.
- *HM* method with $\epsilon = 0.5$.
- Simplified adjusted *B&P* method with trade-up using $\epsilon = 0.1$.
- Simplified *C,G&J* method.
- "*Max2*" from the simplified C&H method.
- "*AddColHM*" from the simplified C&H method.

Comparison of the Methods

The results when comparing the methods are shown only for test sets 1 and 4. The results for test sets 2, 3 and 5 are given in Appendix F.2.

Table 5.15 shows the average running times for the methods are seen for test sets 1 and 4. These running times are the total times it takes to calculate the revenue obtained when using the methods on the simulated arrival process. In the table the slowest methods are seen to be the simplified You method and the HM method, which have similar running times. The simplified You method is slightly faster in some cases, though. The HM method was expected to be fast, since the number of decision periods needed for the method is much smaller than the number of decision periods needed for the simplified You method. This is not the case, though, which is probably due

	$C = 25$	$C = 40$	$C = 55$	$C = 75$
You 1	0.67 s	0.99 s	1.33 s	1.77 s
HM 1	0.71 s	0.98 s	1.27 s	1.71 s
B&P 1	0.13 s	0.18 s	0.24 s	0.32 s
C,G&J 1	0.26 s	0.26 s	0.27 s	0.29 s
Max2 1	0.42 s	0.56 s	0.72 s	0.91 s
AddColHM 1	0.38 s	0.51 s	0.66 s	0.87 s
You 4	0.33 s	0.48 s	0.59 s	0.77 s
HM 4	0.49 s	0.64 s	0.76 s	0.96 s
B&P 4	0.08 s	0.11 s	0.14 s	0.17 s
C,G&J 4	0.20 s	0.22 s	0.20 s	0.19 s
Max2 4	0.30 s	0.39 s	0.45 s	0.57 s
AddColHM 4	0.28 s	0.38 s	0.43 s	0.55 s

Table 5.15: Running Times for the Simplified Methods.

to the number of calculations in the model given in the HM method which are more demanding. Furthermore the “Max2” method and the “AddColHM” method have similar running times, which is not surprising, since they are both based on the same method, i.e., the simplified C&H method. These are approximately 1.7 times faster than both the simplified You method and the HM method. All methods except for the simplified C,G&J method are faster for smaller capacities and the fastest method is by far the simplified adjusted B&P method with trade-up.

As for the previous problems, to determine if one of the methods yields a significantly higher revenue than the others, a paired t-test with a significance level of $\alpha = 5\%$ is used, see Section 5.1.2 for a description.

In Tables 5.16 and 5.17 the relative differences between the methods are shown in percentage.

All five tables of results have to be considered when determining which of the methods that is the best for the simplified SIC problem with trade-up. In Table 5.17 some of the relative percentages are seen to be over 100%. This only happens for test set 4, though, since for this test set the demand is very low compared to the capacity of the aircraft. In Table 5.17 the simplified You and HM methods are seen to yield similar results. When considering all the tables of results, it is seen that a few times the HM method is better than the simplified You method and when this happens the revenue is averagely 0.15% higher. In general in all tables, the tendency is as for test set 4, i.e., the simplified You and HM methods yield the highest revenues, but these are also the methods with the longest running times.

Of the methods with shorter running times the “AddColHM” Method generally yields the highest revenues. This method performs poorly, though, when the capacity is much smaller than the demand. For instance for $C = 25$ for test set 1, where the total demand is 82, the “AddColHM” method yields

	HM	B&P	C,G&J	Max2	AddColHM
C=25					
You	0	33.39	14.18	11.78	19.48
HM	-	33.10	14.08	11.91	19.56
B&P	-	-	**17.69**	**22.68**	**13.41**
C,G&J	-	-	-	**4.76**	6.67
Max2	-	-	-	-	9.37
C=40					
You	**0.17**	30.19	7.97	6.94	11.54
HM	-	30.35	8.19	6.93	11.84
B&P	-	-	**20.91**	**22.47**	**18.49**
C,G&J	-	-	-	**1.80**	5.44
Max2	-	-	-	-	5.20
C=55					
You	**0.31**	24.45	3.29	7.24	1.05
HM	-	24.67	3.55	7.41	1.35
B&P	-	-	**20.74**	**17.21**	**23.49**
C,G&J	-	-	-	6.95	**2.46**
Max2	-	-	-	-	**6.38**
C=75					
You	0	28.97	17.18	5.20	0
HM	-	28.97	17.18	5.20	0
B&P	-	-	**10.85**	**23.49**	**28.97**
C,G&J	-	-	-	**12.33**	**17.18**
Max2	-	-	-	-	**5.20**

Table 5.16: Rel. Diff. in % for Test Set 1 for the Simplified Problem.

a revenue which is 19.50% lower than the revenue obtained with the simplified You and HM methods. For all instances with $C = 25$, except for test set 4, "AddColHM" performs much worse than the You and HM methods, but otherwise the three methods yield similar results. The simplified adjusted B&P method with trade-up does not show the same nice results as it did for both the SIC problem without trade-up and the general problem with trade-up. In fact the simplified adjusted B&P method has a very poor performance, since it never yields a higher revenue than any of the other methods. Hence, even though this method has been one of the recommended methods for the other problems, since the simplified SIC problem with trade-up is the one BA is interested in, the simplified B&P method should not be used by the airline to solve the simplified problem. If the demand is expected to be much larger than the capacity of the aircraft, then it is recommended to use either the simplified You solution method or the HM method, otherwise the "AddColHM" method is recommended, since this is about 1.7 times faster than both of these methods.

Notice, the running times for the methods do not satisfy BA's time constraint of 0.45 seconds, but again the running times are assumed to be reduced significantly by using another programming language.

	HM	**B&P**	**C,G&J**	**Max2**	**AddColHM**
C=25					
You	**0.08**	87.95	64.18	5.53	**0.08**
HM	-	87.84	64.22	5.36	0
B&P	-	-	**15.69**	**79.23**	**88.05**
C,G&J	-	-	-	**56.95**	**64.29**
Max2	-	-	-	-	**5.56**
C=40					
You	0.01	100.05	98.18	2.14	NoD
HM	-	100.03	98.16	2.13	**0.01**
B&P	-	-	**1.18**	**96.87**	**100.05**
C,G&J	-	-	-	**95.08**	**98.18**
Max2	-	-	-	-	**2.14**
C=55					
You	NoD	101.79	101.78	2.01	NoD
HM	-	101.79	101.78	2.01	NoD
B&P	-	-	0	**98.68**	**101.79**
C,G&J	-	-	-	**98.66**	**101.78**
Max2	-	-	-	-	**2.01**
C=75					
You	NoD	102.67	102.67	2.23	NoD
HM	-	102.67	102.67	2.23	NoD
B&P	-	-	NoD	**99.23**	**102.67**
C,G&J	-	-	-	**99.23**	**102.67**
Max2	-	-	-	-	**2.23**

Table 5.17: Rel. Diff. in % for Test Set 4 for the Simplified Problem.

Comparison of TU Methods with Non-TU Methods

The assumptions for the simplified SIC problem with trade-up are much different than those for the SIC problem without trade-up. Hence, it is interesting to see how the methods for the SIC problem without trade-up perform in a trade-up market with the assumptions from BA. Therefore it is chosen to compare the two best methods from the SIC problem without trade-up with the two best methods from the simplified SIC problem with trade-up. Since the simplified You method and the HM method yield very similar results, only the HM method is chosen for the comparisons. The "AddColHM" method is chosen as well, since this is one of the faster methods. The methods from the SIC problem without trade-up are the same as used in the comparison in the general problem, i.e., the L&H method with $\epsilon = 0.01$ and the EMSRb method. Thus the four methods used in the comparison are the L&H method and the EMSRb method for the problem without trade-up and the HM and the "AddColHM" methods for the problem with trade-up.

When using the methods without trade-up in the simplified trade-up market, the demand which is used in the methods is the marginal demand. The marginal demand is calculated from the total demand for class k in the trade-up market and the trade-up rates $q_{k,i}$ using (4.24) page 60. The open classes

	L&H	EMSRb
C=25		
HM	60.32	33.94
AddColHM	35.34	13.03
C=40		
HM	40.05	31.47
AddColHM	26.12	18.27
C=55		
HM	29.29	24.25
AddColHM	28.17	23.23
C=75		
HM	29.61	26.38
AddColHM	29.60	26.37

Table 5.18: Rel. Diff. in % when Comp. Non-TU with Simple TU Methods, Test Set 1.

are then determined with the methods without trade-up in the same way as for the problem without trade-up. These open classes are then used in the acceptance/rejection process on the simulated arrivals for the simplified trade-up market.

	L&H	EMSRb
C=25		
HM	138.30	101.09
AddColHM	138.38	101.29
C=40		
HM	97.16	94.40
AddColHM	97.17	94.41
C=55		
HM	100.46	100.47
AddColHM	100.46	100.47
C=75		
HM	99.09	99.09
AddColHM	99.09	99.09

Table 5.19: Rel. Diff. in % when Comp. Non-TU with Simple TU Methods, Test Set 4.

Tables 5.18 and 5.19 show the results for test sets 1 and 4. The remaining tables for test sets 2, 3 and 5 are given in Appendix G.3. Again all tables of results must be considered when comparing the methods. The tables contain the relative differences in percentage and as for the general problem with trade-up, the tables have been reduced, such that the two methods solving the same problem are not compared.

In Table 5.18 the methods for the problem with trade-up are on average seen to be 20% - 30% better than the methods without trade-up. This is the general tendency for test sets 2, 3 and 5 as well. In Table 5.19 the relative differences are seen to be close to 100% or higher. Hence, for test set 4 the methods with trade-up yield a revenue which is approximately two times higher than the methods without trade-up. This is an extremely large difference and therefore this is investigated further. In Figure 5.10 the acceptance/rejection process for all four methods are shown for $C = 25$, since the largest relative differences are seen for this capacity. Notice that there is a variable number of decision periods in the figures, depending on which value of ϵ is used for the method. The title of each figure shows the name of the method, the value of ϵ used, the revenue obtained with the method for that arrival pattern and the load factor (LF) of the aircraft. The load factor is the number of accepted requests divided by the capacity of the aircraft and multiplied by 100%.

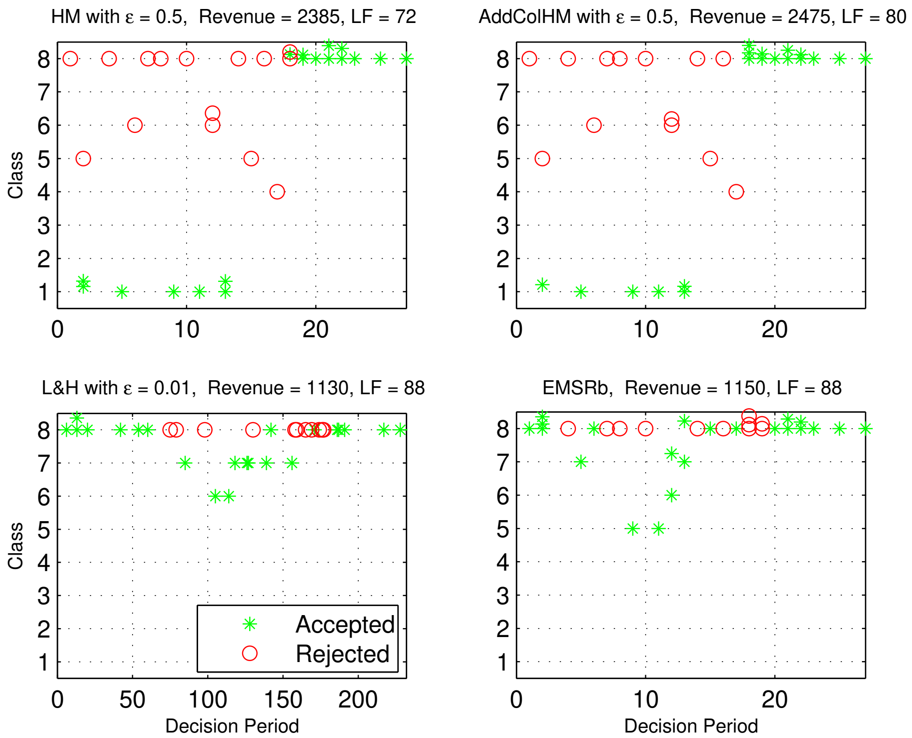


Figure 5.10: Acceptance/Rejection process for the Simplified Problem, Test Set 4.

Recall, that only demand for class 8 occur, since this is the lowest class in the market, thus requests for all other classes is due to trade-up. If a request for class 8 is rejected, the figures show the highest-fare classes to which a passenger is willing to trade up. Hence, many passengers are seen to be willing to trade up. Furthermore, Figure 5.10 shows that both the HM and "AddColHM" methods accept requests for fare class 8 in the beginning of the booking period and later in the booking period only fare class 1 is open. Contrary to this both the L&H and the EMSRb methods accept requests for class 8 throughout the booking period. Furthermore these methods do not sell any class 1, 2, 3 or 4 tickets. The reason for this pattern of accepted and rejected requests is found by considering the demand in test set 4. In the last data interval before departure, data interval 1, the marginal demand given to the methods without trade-up is

$$\begin{aligned} D_1^1 &= 1.68, \quad D_2^1 = 0.09, \quad D_3^1 = 0.09, \quad D_4^1 = 0.09, \\ D_5^1 &= 0.09, \quad D_6^1 = 0.09, \quad D_7^1 = 0.09, \quad D_8^1 = 0.84, \end{aligned}$$

where D_i^t is the demand for class i in data interval t. The total demand for test set 4 for all data intervals is 33, hence the marginal demand for class 1 and class 8 are significant. The model in the L&H method does not assume trade-up, and thus the decision is to open class 8 close to departure, since it is better to sell a seat to class 8 than not to sell the seat. The model expects that the demand for class 1 will book class 1 only, i.e., that there is no danger of trade-down. This is where the model fails, since in the trade-up market demand for class 1 only occurs through trade-up from class 8, but if class 8 is open the trade-up will never happen. This explains why close to departure, when the trade-up rate is high, no trade-up can be achieved. The HM method only leaves class 1 open and thus it is not surprising that the revenue difference is large. Notice that the methods with trade-up obtains the higher revenue with a lower load factor of the aircraft than the methods without trade-up. I.e., a higher revenue is obtained by accepting fewer requests.

Therefore in general the HM and "AddColHM" methods are much better than both the L&H and EMSRb methods, except for one instance for the "AddColHM" method. This is for test set 2 with $C = 25$, where the "AddColHM" method is 3.49% and 6.42% worse than the L&H and EMSRb method, respectively. To get an idea of why this happens, the acceptance/rejection process is shown for the four methods for test set 2 with $C = 25$ in Figure 5.11. In this figure it is seen that the L&H and EMSRb methods do not accept any class 8 requests, i.e., all tickets sold are due to trade-up. This is not the case for the "AddColHM" method, which accepts many class 8 requests in the beginning of the booking period. Hence, close to departure when passengers are willing to trade up to high-fare classes, these are rejected due to lack of capacity. The load factor of 100% is already reached in decision period 12.

The results depend very much on the demand used in the non-trade-up methods. Looking at the results it may not be a good idea to use marginal

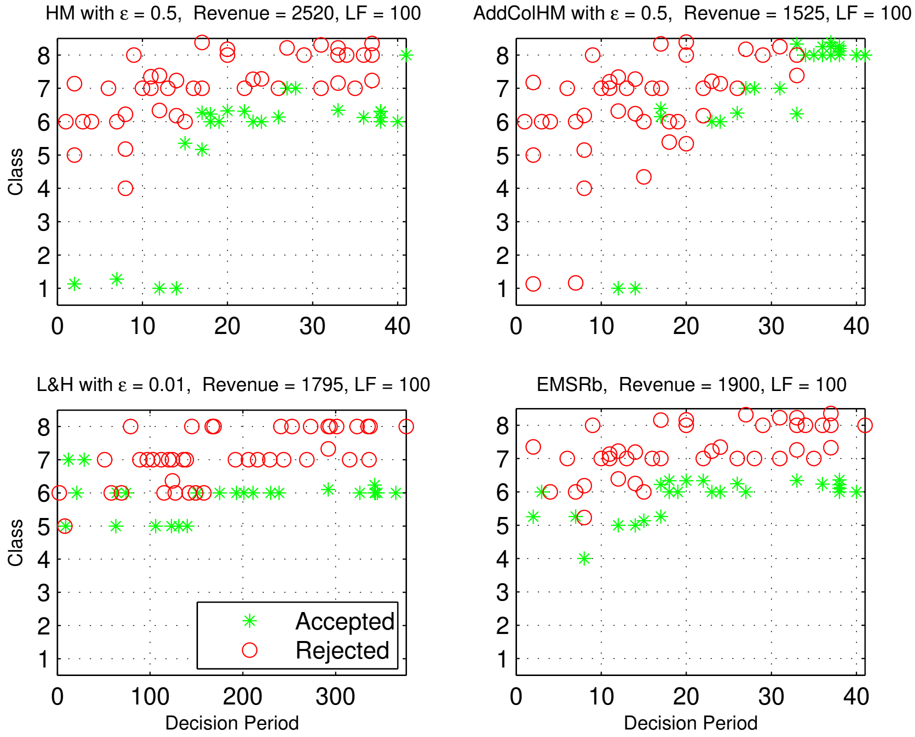


Figure 5.11: Acceptance/Rejection process for the Simplified Problem, Test Set 2.

demands, since using this yields a low revenue. There may be other demand which give better results. For instance, when it is recognized that a non-trade-up model leaves class 8 open close to departure, because it does not expect any risk of trade-down, then this can be counteracted by giving a higher demand forecast for class 1, such that the model at least closes class 8 for small remaining capacities. This way a higher revenue can be obtained by adjusting the demand given as input.

The final conclusion is that when using methods with trade-up instead of methods without trade-up in a trade-up market a gain in revenue can be obtained. Except for the instance in test set 2, the lowest relative percentage which appears in all five tables is approximately 10%. Hence, even in the best case for the L&H and EMSRb methods the difference in the revenue is large. This difference may get smaller if another demand than a marginal demand is used for the non-trade-up methods. There is no doubt, though, that in a market where trade-up occurs, it is important for an airline to use the methods with trade-up instead of the methods without trade-up.

Summary of Results

Due to the extensive number of methods, a reduction in the final methods used in the results is made. This is done by only choosing one simplified B&P method, one simplified C,G&J method and two of eight hybrid methods. The simplified adjusted B&P method with trade-up is chosen. This method is also used in the simplified C,G&J method to calculate the bounds. The two hybrid methods used in the final results are the "Max2" method and the "AddColHM" method.

Table 5.20 sums up the main results of this section. In the top half it shows the results for the HM method for test set 1 and in the lower half it shows the results for the simplified You method for test set 3. The conclusion is that the HM and simplified You methods yield the best results. These methods yield higher revenues than all other methods in most instances, but the running times are also the longest. Table 5.20 also shows that the performance of the "AddColHM" method is worse for small capacities. For larger capacities the method yields revenues which are very similar to those obtained with the HM and simplified You methods.

	Capacity	**You**	**B&P 1**	**C,G&J 1**	**Max2 1**	**AddColHM 1**
HM 1	$C = 25$	0	33.10	14.08	11.91	19.56
HM 1	$C = 40$	0.17	30.35	8.19	6.93	11.84
HM 1	$C = 55$	0.31	24.67	3.55	7.41	1.35
HM 1	$C = 75$	0	28.97	17.18	5.20	0
	Capacity	**HM**	**B&P 3**	**C,G&J 3**	**Max2 3**	**AddColHM 3**
You 3	$C = 25$	0	26.06	5.37	3.28	7.53
You 3	$C = 40$	0	20.91	2.75	3.56	0.48
You 3	$C = 55$	0	20.46	18.25	2.81	0
You 3	$C = 75$	NoD	20.70	20.70	3.09	NoD

Table 5.20: Summary of results for the Simplified SIC problem with Trade-Up.

In the simplified trade-up market the HM and the "AddColHM" methods are compared with the two best methods for the problem without trade-up, i.e., with the EMSRb and L&H methods. From the comparisons, it is seen that the methods for the problem with trade-up on average yield revenues which are 20% - 30% higher than the revenue obtained with the methods without trade-up. The differences in the revenues may get smaller if a different demand than marginal demand is used for the non-trade-up methods. The conclusion is, though, that an increase in revenue is expected by using methods with trade-up in the simplified trade-up market instead of methods without trade-up.

Chapter 6

Summary and Conclusion

In this chapter a short summary of this report is given. Furthermore the main conclusions are recapitulated and the contributions of this work are outlined. Finally, some areas for further work are suggested.

6.1 Summary

The main task given by British Airways for this work was to develop an efficient DP algorithm for the revenue optimization problem which can be used in practice in the presence of trade-up behaviour. Due to the extensive number of flights, which BA has in the system at all times, the meaning of "can be used in practice" is that each optimization of the problem for the entire booking period has to take less than 0.85 seconds.

To make the reader familiar with the revenue optimization problem, especially the Seat Inventory Control problem, different concepts and challenges regarding the SIC problem were introduced in Section 1.1. Furthermore, trade-up behaviour was described in this section.

6.1.1 The SIC Problem without Trade-Up

The SIC problem without trade-up was examined before trade-up was incorporated such that a fundamental understanding of the problem was obtained. Initially, two static methods were given for solving the SIC problem without trade-up, the EMSRa and EMSRb methods and furthermore a dynamic model was set up for this problem. For solving the dynamic model an exact method and an approximation method was proposed, the L&H method and the B&P method, respectively. The four methods were compared by considering both the running time for the methods as well as the revenues obtained from a simulated arrival process. The results showed that the L&H method gave the highest revenue when the assumptions of the DP model were satisfied, but since the running times were 4 seconds on average, this method would have to be developed further to be feasible. The B&P method was slightly faster than the EMSRb method and both of these were much faster than the L&H method. Which one of these gave the best results depended on the request arrivals, and hence no conclusion could be drawn. Thus, the recommendation was to use the EMSRb method for the SIC problem without trade-up, since the booking system would have to be changed if the B&P method is to be used.

It is well known that using Matlab as a general programming tool gives longer running times than using a high-level programming language such as C++. Hence, it is expected that the running times of the methods can be significantly reduced. Furthermore, all numerical experiments were run on a SUN Fire 3800 with a 1200 Mhz processor and 4 GB RAM. When running the same experiments on a HP x4000 with a 1800 Ghz processor and 1.5 GB RAM reductions in running time of approximately 10% to 15% are obtained.

6.1.2 The SIC Problem with Trade-Up

For the SIC problem with trade-up two cases were considered, each with different assumptions. In the first case, the general problem, the assumption is that the buying conditions for the classes in the trade-up market differ, and therefore demand occurs for all classes. In the second case, the simplified problem, the assumption is that there is no difference in the buying conditions for the fare classes, hence passengers always request the lowest-fare class in the market. The latter is the assumption made by British Airways. For both problems it is assumed that if a passenger trades up, then this will always be to the lowest open class, since the buying conditions get less restrictive the higher the fare gets.

In a trade-up market two kinds of classes exist, non-trade-up and trade-up classes. The non-trade-up classes are for instance transfer flights to and from which it is not possible for the customer to trade up. To simplify the problem in this report it was chosen to omit the non-trade-up classes, such that only the trade-up classes in the market were modelled. These trade-up classes form approximately half of the trade-up market. Hence to satisfy the overall time limit on the optimizations given by BA, the optimizations for the problems with trade-up had to be done in less than approximately 0.45 seconds.

General SIC with Trade-Up

For the general SIC problem with trade-up a static method, the EMSRb method with trade-up, was described. Furthermore, a DP model was set up and an exact solution method for solving this, the You method, was described. To obtain an efficient algorithm for the SIC problem with trade-up, the methods considered were mainly approximation algorithms to solve this DP model. As described in the literature review, not many papers deal with heuristics for a DP model with trade-up incorporated. Therefore some ideas from papers dealing with the problem *without* trade-up were applied to the problem *with* trade-up to develop some new approximation methods for solving the DP model with trade-up. These new heuristics were the C,G&J method and the C&H method. Finally, the B&P method with trade-up was described in two different versions, B&P and B&P LP.

The revenues obtained with the methods and the running times for these were compared. The comparisons showed that the highest revenues could be obtained with the You method and the C,G&J method. Hence, it was recommended to use one of these methods for solving the general SIC problem with trade-up. The running times for both methods were a little too long to be feasible for BA. It is expected, though, that if an efficient optimizer is used to solve the LPs in the C,G&J method and if both methods are implemented in another programming language than Matlab, running times can be reduced sufficiently.

The two best solution methods for the SIC problem without trade-up were applied in a trade-up market. The results obtained with these methods were compared with the results obtained with the two best trade-up methods applied in a trade-up market. As expected revenues obtained with the trade-up methods were better than the results obtained with the methods for the problem without trade-up. An average increase in revenue of approximately 7% - 8% were obtained, and hence when dealing with a market in which trade-up occurs, it was recommended to use solution methods which incorporate trade-up.

The Simplified SIC problem

For the simplified SIC problem with trade-up almost the same methods as for the general problem were described with the simplifying assumption. Instead of the EMSRb method with trade-up, the HM method was introduced, in which a different DP model than the DP model in the You method was set up. When implementing the methods and comparing these, it was seen that the simplified You method and the HM method gave similar results.

The highest revenues were obtained with these two methods, but these also had the longest running times. Of the approximation methods with shorter running times the "AddColHM" method, which was a version of the C&H method, generally gave the highest revenues. This method performed poorly, though, when the capacity of the aircraft was much smaller than the total demand in the booking period. The revenues obtained with the simplified B&P method with trade-up were at all times lower than the revenues obtained with any of the other methods. The recommendation was that if the demand is expected to be much larger than the capacity of the aircraft, then either the simplified You method or the HM method should be used, otherwise the "AddColHM" method is preferable, since this is about 1.7 times faster than both the simplified You and HM methods.

The running times for the methods did not satisfy BA's time constraint of 0.45 seconds, but again we believe that by using another programming language the running times can be reduced and satisfy the time constraint.

Finally, in the simplified trade-up market a comparison of two methods with trade-up, the HM and "AddColHM" methods, and the two best methods without trade-up, the EMSRb and the L&H methods, was made. It was seen that the methods with trade-up on average gave revenues which were 20% - 30% higher than the revenues obtained with the methods for the problem without trade-up. It was expected that smaller differences in the revenues would be obtained if a different demand than marginal demand was used for the non-trade-up methods. The conclusion was, as for the general problem, that an increase in revenue is expected by using methods with trade-up in the simplified trade-up market instead of methods without trade-up.

6.2 Conclusion

For a market where trade-up does not occur, then even though the EMSRb and the B&P methods performs similarly, the recommendation is to use the EMSRb method for the SIC problem without trade-up. The reason for this is that the booking system will have to be changed if the B&P method is to be used.

To solve the general SIC problem with trade-up, the final conclusion is that either the You method or the C,G&J method is recommended.

When comparing the best methods for the problem without trade-up with the best methods for the general problem with trade-up, it is concluded that an average increase in revenue of approximately 7% - 8% can be obtained. Hence, in a trade-up market, models which incorporate trade-up must be used.

For the simplified SIC problem with trade-up, the recommendation is that if the demand is expected to be much larger than the capacity of the aircraft, then either the simplified You method or the HM method should be used, otherwise the "AddColHM" method is preferable, since this is about 1.7 times faster than both the You and HM methods.

In a trade-up market, it is seen that the methods with trade-up yield a much higher revenue than those without trade-up. An average increase in revenues of approximately 20% - 30% can be obtained when using methods, which take trade-up into account instead of methods which do not incorporate trade-up.

6.3 Contributions of this Work

The main purpose of this work has been to develop an efficient DP algorithm with trade-up. This goal is reached by initially investigating existing approximation algorithms which solve a dynamic programming model for the SIC problem with trade-up. Furthermore, two new methods for solving the DP model with trade-up are developed by applying ideas obtained from papers dealing with the SIC problem without trade-up. Hence, this work adds new perspectives to the literature on the SIC problem with trade-up.

Besides adding new methods for solving the problem with trade-up, this work also gives an extensive investigation and comparison of both exact and approximation algorithms for solving a DP model with trade-up. In none of the papers in the bibliography of this work, a comparison of several different methods for the SIC problem with trade-up occurs.

Finally, it is investigated how methods for a non-trade-up market perform in a trade-up market compared with methods incorporating trade-up. The conclusion is that in a trade-up market a large gain in revenue can be obtained by applying methods which incorporate trade-up instead of methods without trade-up.

6.4 Further Work

Many areas of the SIC problem with trade-up remain to be investigated, both in this work and in general. The problem examined in this report is the SIC problem with trade-up for a single-leg flight and without multiple bookings. Hence, modelling the problem for a network and including multiple bookings are just some of the areas which can be added to the problem. Furthermore the model can be extended to incorporate cancellations, overbookings or no-shows.

Regarding the trade-up market, a method which deals with both the non-trade-up and trade-up classes is to be developed. In this project only the trade-up classes were included in the methods. The results for the methods with and without trade-up can be used to determine which methods might be worth combining to develop a new method handling both types of classes.

The bounds used in the C,G&J method are determined very superficially in this work and a more thourough investigation of the choice of these should be made. Furthermore, the stochastic LP should be implemented and used as a lower bound.

Finally implementing the methods in another programming language than Matlab is a necessity. This is both due to the slowness of Matlab and the compatibility of the method with the existing booking systems at the airlines.

Appendix A

Rewriting the Value Function

In the following the value function for the SIC problem without trade-up (3.9) page 25 is rewritten to include $\Delta V_t(x)$.

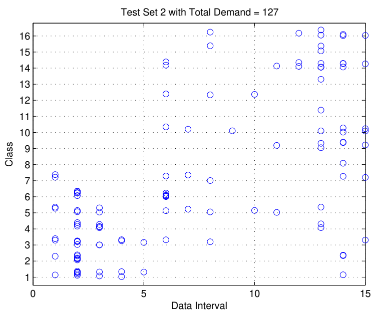
$$
\begin{aligned}
V_t(x) - V_{t-1}(x) &= P_0^t V_{t-1}(x) + P_1^t\big(F_1 + V_{t-1}(x-1)\big) \\
&\quad + \sum_{i=2}^{k} P_i^t \max\big(F_i + V_{t-1}(x-1), V_{t-1}(x)\big) - V_{t-1}(x) \\
&= \big(1 - \sum_{i=1}^{k} P_i^t\big)V_{t-1}(x) + P_1^t\big(F_1 + V_{t-1}(x-1)\big) \\
&\quad + \sum_{i=2}^{k} P_i^t \max\big(F_i + V_{t-1}(x-1), V_{t-1}(x)\big) - V_{t-1}(x) \\
&= V_{t-1}(x) - V_{t-1}(x) + P_1^t\big(F_1 + V_{t-1}(x-1)\big) \\
&\quad - \sum_{i=1}^{k} P_i^t V_{t-1}(x) + \sum_{i=2}^{k} P_i^t \max\big(F_i + V_{t-1}(x-1), V_{t-1}(x)\big) \\
&= P_1^t\big(F_1 + V_{t-1}(x-1)\big) - P_1^t V_{t-1}(x) \\
&\quad + \sum_{i=2}^{k} \big(P_i^t \max\big(F_i + V_{t-1}(x-1), V_{t-1}(x)\big) - P_i^t V_{t-1}(x)\big) \\
&= P_1^t\big(F_1 + V_{t-1}(x-1) - V_{t-1}(x)\big) \\
&\quad + \sum_{i=2}^{k} P_i^t \max\big(F_i + V_{t-1}(x-1) - V_{t-1}(x), V_{t-1}(x) - V_{t-1}(x)\big) \\
&= P_1^t\big(F_1 - \Delta V_{t-1}(x)\big) + \sum_{i=2}^{k} P_i^t \max\big(F_i - \Delta V_{t-1}(x), 0\big)
\end{aligned}
$$

where it is used that $\sum_{i=1}^{k} P_i^t + P_0^t = 1$.

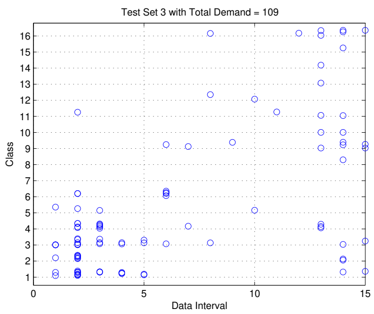
Appendix B

Demand Patterns

In Figures B.1 and B.2 the demand patterns for test sets 2, 3 and 5 for the SIC problem without trade-up can be seen.



(a) Demand for Test Set 2.



(b) Demand for Test Set 3.

Figure B.1: Demand Patterns for Two Test Sets without Trade-Up.

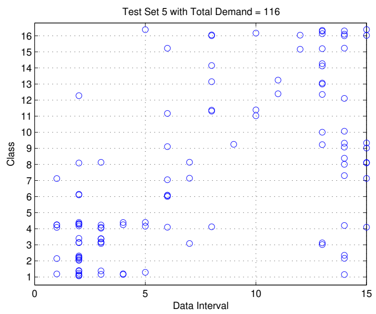
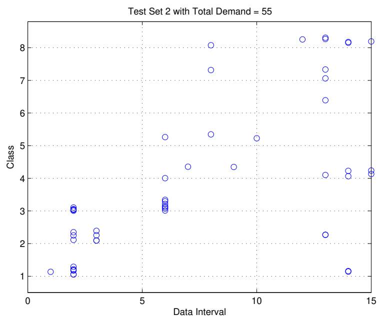
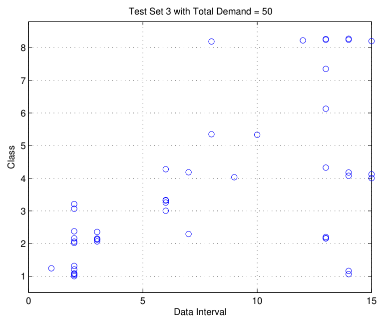


Figure B.2: Demand Pattern for Test Set 5 without Trade-Up.

In Figures B.3 and B.4 the demand patterns for test sets 2, 3 and 5 for the SIC problem with trade-up can be seen.



(a) Demand for Test Set 2.



(b) Demand for Test Set 3.

Figure B.3: Demand Patterns for Two Test Sets with Trade-Up.

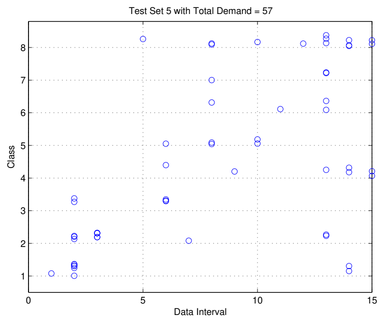


Figure B.4: Demand Pattern for Test Set 5 with Trade-Up.

Appendix C

Overview of Variables in Programs

In Table C.1 a list of the main functions for the SIC problem without trade-up can be seen. Furthermore, in this table the different inputs and outputs used by the programs are given. The description of these variables can be seen in Table C.2.

Program	**Input**	**Output**
`Simulation.m`	$D, F, C, TimeVec, nyj$	$t, class, dataint, n$
`EMSR.m`	$D, C, F, DecPer, class, dataint, n$	$Revenue$
`EMSRb.m`	$D, C, F, DecPer, class, dataint, n$	$Revenue$
`LHBL.m`	D, C, F, nyj	$BLmatrix$
`PopBL.m`	D, C, F, nyj	$BLmatrix$
`calcRevenue.m`	$C, D, F, nyj, class, dataint,$ $n, BLmatrix, DecPer$	$Revenue$

Table C.1: Inputs and Outputs for the Programs without Trade-Up.

In Table C.3 a list of the main functions and their inputs and outputs for the general SIC problem with trade-up can be seen. In Table C.5 the description of the variables which are different from those used for the SIC problem without trade-up described in Table C.2 can be seen.

In Table C.4 a list of the main functions and their inputs and outputs for the simplified SIC problem with trade-up can be seen. The input variables for the programs for this problem have been described in the Tables C.2 and C.5. There is only one difference and that is the demand matrix D in Table C.2, which is now a demand vector for class k.

Variable	**Description**
D	Matrix where the (i, j)th element is the demand forecast for the data interval in the beginning of the booking period to data interval j for fare class i.
$epsilon$	Tolerance such that the probability of more than one arrival in each decision period is smaller than epsilon.
C	Capacity of the aircraft.
F	Vector containing the fare of each class.
$TimeVec$	Vector containing the start times of each data interval. The first element is the start time of the last data interval and the last element is the end time of the first data interval, i.e., departure time.
$DecPer$	Vector containing the decision periods in which the booking limits need to be updated.
t	Vector where the ith element is the arrival time for the ith arrival in the simulation process.
$class$	Vector containing the classes, which are requested in the simulation process.
$dataint$	Vector containing the data intervals in which the arrivals occur in the simulation process.
n	Vector containing the decision periods in which the arrivals occur in the simulation process.
nyj	Vector containing the number of equally sized decision periods in each data interval.
$BLmatrix$	Matrix where the (i, j)th element is the booking limit for fare class i in decision period j.
$Revenue$	The revenue obtained from the simulation process.

Table C.2: Description of Variables in the Programs without Trade-Up.

Program	**Input**	**Output**
`SimulationTradeup.m`	$D, F, C, TimeVec, nyj$	$t, dataint$ $class, n, TUarr$
`EMSRbTUdecperTU.m`	$D, C, F, DecPer, class, dataint$ $n, TUarr, TU2dim, Q, nyj$	$Revenue$
`You.m`	$D, TU3dim, C, F, nyj$	j, H, V
`PopBLTU.m`	$Dtot, C, F, nyj$	M, V
`BPLPTU.m`	$D, C, F, update, TU2dim, TU3dim, nyj$	H, j
`BPTUSpline.m`	$Dtot, F, Cknot, Tknots$	V
`BPLPSplineNew.m`	$D, F, Cknot, Tknots, TU2dim, TU3dim$	V
`hybridMax.m`	$fapprox, fexact, D, C, F, switchTime$ $TU, TU3dim, indicator, nyj1, nyj2$	j, H, M
`hybridAddCol.m`	$fapprox, fexact, D, C, F, switchTime$ $TU, TU3dim, indicator, nyj1, nyj2$	j, H, M
`hybridCombi.m`	$fapprox, fexact, D, C, F, switchTime$ $TU, TU3dim, nyj1, nyj2$	j, H, M
`RevenueYT.m`	C, F, H, j, Q $class, n, dataint, TUarr$	$Revenue$
`matrixRevenue.m`	C, F, M, Q $class, n, dataint, TUarr$	$Revenue$
`RevenueBPLP.m`	$C, F, BLmatrix, Q, class$ $n, dataint, TUarr, update$	$Revenue$
`RevenueSpline.m`	$C, F, LB, UB, Q, class$ $n, dataint, TUarr, nyj$	$Revenue$
`RevenueSplineBPLP.m`	$C, F, LB, UB, TU3dim, class$ $n, dataint, TUarr, nyj$	$Revenue$
`RevenueHybrid.m`	$C, F, H, j, M, Q, class, dataint$ $TUarr, nyj1, n1, nyj2, n2, switchTime$	$Revenue$

Table C.3: Inputs and Outputs for the General Programs with Trade-Up.

Program	**Input**	**Output**
`SimulationTradeupSimple.m`	$D, TU, F, C,$ $TimeVec, Nyj$	$t, dataint,$ $maxclass, n$
`YouTalluriSimple.m`	D, TU, C, F, nyj	M, V
`HMSimple.m`	D, TU, C, F, nyj	M, V
`PopBLTUSimple.m`	D, TU, C, F, nyj	M, V
`BPTUSplineSimple.m`	$D, F, Cknots, Tknots, TU$	V
`hybridSimpleMax.m`	$fapprox, fexact, D, C, F, TU$ $switchTime, indicator, nyj1, nyj2$	M
`hybridSimpleAddCol.m`	$fapprox, fexact, D, C, F$ $switchTime, TU, nyj1, nyj2$	M
`hybridSimpleCombi.m`	$fapprox, fexact, D, C, F$ $switchTime, TU, nyj1, nyj2$	M
`matrixRevenueSimple.m`	$C, F, M, maxclass, nyj1,$ $n1, nyj2, n2, switchTime$	$Revenue$
`RevenueSplineSimple.m`	$C, F, LB, UB, maxclass,$ $n, dataint, nyj$	$Revenue$

Table C.4: Inputs and Outputs for the Simplified Programs with Trade-Up.

Variable	Description
$Dtot$	Matrix where the (i, j)th element corresponds to the total expected number of requests for fare class i in data interval j including those who trade-up to class i from lower classes.
TU	Matrix where the (i, j)th element is the trade-up rate from fare class k to class i and all more expensive fare classes in data interval $h - j + 1$, where h is the total number of data intervals and data interval 1 is closest to departure.
Q	Three dimensional matrix where the third dimension is data intervals. The (i, j)th element in each of the matrices in the third dimension is the trade-up rate from class i to class j. The first matrix in the third dimension corresponds to departure.
$TU2dim$	Matrix where the (i, j)th element is the trade-up rate from fare class $j + 1$ to class j and all more expensive fare classes in data interval i.
$TU3dim$	Three dimensional matrix where the (i, j, h)th element is the trade-up rate from class i to class j and all higher-fare classes in data interval h.
H	Three-dimensional matrix where the first dimension is decision periods, the second dimension is remaining capacity and the third dimension is class. The elements in H are given by (4.12).
j	Three dimensional matrix with the same dimensions as H. Contains the lowest class offered if the request is rejected.
V	Matrix where the (i, j)th element is the maximum expected revenue obtained when the remaining capacity is i in decision period j.
M	Matrix where the (i, j)th element corresponds to the lowest open class with a remaining capacity of i in decision period j.
$TUarr$	Vector containing a uniformly distributed random number between 0 and 1 for each simulated arrival representing the customers willingness to trade-up.
$update$	Vector containing the data intervals in which the booking limits need to be updated.
LB	Matrix where the (i, j)th element is the value of the lower bound with a remaning capacity of i in data interval j.
UB	Matrix where the (i, j)th element is the value of the upper bound with a remaning capacity of i in data interval j.
$Tknots$	Vector containing the data intervals in which the LP for the upper bound in the C,G& J method must be solved.
$Cknots$	Vector of remaining capacities which for each time in $Tknot$ are knots in a spline. First element must be 1 and last element must be C.
$switchTime$	The data interval, in which the hybrid methods changes from an approximation method to an exact method.
$fapprox$	Function which is used to approximate the maximal expected revenue in the beginning of the booking period until $switchTime$ in the C&H method.
$fexact$	Function which is used to calculate the maximal expected revenue from $switchTime$ to departure using an exact method.

Table C.5: Description of Variables in the Programs with Trade-Up.

Appendix D

Parameter Tuning

D.1 SIC without Trade-Up

D.1.1 The L&H Method

In Table D.1 the average running time from 1000 test runs for the L&H method and the L&H method with updates are seen.

In Table D.2 page 146 the number of times where one value of ϵ yields a higher revenue than another is seen for different capacities and test sets for 1000 test runs. The values of ϵ are $\epsilon_1 = 0.001$, $\epsilon_1 = 0.1$, $\epsilon_1 = 0.2$ and $\epsilon_1 = 0.3$.

D.1.2 The B&P Method

In Table D.3 page 147 the average running time from 1000 test runs for the B&P method are seen.

In Table D.4 page 147 the number of times where one value of ϵ yields a higher revenue than another is seen for different capacities and test sets for 1000 test runs with the B&P method. The values of ϵ are $\epsilon_1 = 0.001$, $\epsilon_2 = 0.1$, $\epsilon_3 = 0.2$ and $\epsilon_4 = 0.3$.

D.2 SIC with Trade-Up

D.2.1 The General Methods

The You Method

In Table D.5 page 148 the average running time from 1000 test runs for the General You solution method are seen.

		Average Running Time			
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.1$	$\epsilon_3 = 0.2$	$\epsilon_4 = 0.3$
C = 80	L&H 1	16.02 s	1.42 s	0.94 s	0.73 s
	L&H 2	11.39 s	1.07 s	0.69 s	0.55 s
	L&H 3	9.70 s	0.92 s	0.61 s	0.48 s
	L&H 4	9.72 s	0.90 s	0.60 s	0.49 s
	L&H 5	10.38 s	0.97 s	0.66 s	0.52 s
	L&Hup 1	16.04 s	1.44 s	0.96 s	0.75 s
	L&Hup 2	11.41 s	1.09 s	0.71 s	0.57 s
	L&Hup 3	9.71 s	0.93 s	0.63 s	0.50 s
	L&Hup 4	9.73 s	0.91 s	0.62 s	0.50 s
	L&Hup 5	10.39 s	0.99 s	0.68 s	0.54 s
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.1$	$\epsilon_3 = 0.2$	$\epsilon_4 = 0.3$
C = 100	L&H 1	19.1 s	1.74 s	1.15 s	0.93 s
	L&H 2	14.12 s	1.31 s	0.86 s	0.67 s
	L&H 3	12.13 s	1.13 s	0.75 s	0.59 s
	L&H 4	11.88 s	1.11 s	0.74 s	0.59 s
	L&H 5	12.87 s	1.20 s	0.78 s	0.62 s
	L&Hup 1	19.11 s	1.76 s	1.17 s	0.95 s
	L&Hup 2	14.14 s	1.32 s	0.88 s	0.69 s
	L&Hup 3	12.15 s	1.14 s	0.76 s	0.60 s
	L&Hup 4	11.90 s	1.12 s	0.75 s	0.61 s
	L&Hup 5	12.88 s	1.21 s	0.79 s	0.64 s
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.1$	$\epsilon_3 = 0.2$	$\epsilon_4 = 0.3$
C = 120	L&H 1	22.83 s	2.11 s	1.41 s	1.04 s
	L&H 2	16.82 s	1.53 s	1.03 s	0.79 s
	L&Hup 1	22.85 s	2.14 s	1.43 s	1.06 s
	L&Hup 2	16.84 s	1.54 s	1.05 s	0.80 s
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.1$	$\epsilon_3 = 0.2$	$\epsilon_4 = 0.3$
C=140	L&H 1	25.98 s	2.32 s	1.52 s	1.18 s
	L&Hup 1	26.00 s	2.34 s	1.54 s	1.20 s

Table D.1: Running Times of the L&H Method with Different Values of ϵ.

In Table D.6 page 148 the number of times where one value of ϵ yields a higher revenue than another is seen for different capacities and test sets for 1000 test runs. The values of ϵ are $\epsilon_1 = 0.001$, $\epsilon_2 = 0.01$, $\epsilon_3 = 0.1$ and $\epsilon_4 = 0.2$.

The B&P Method with Trade-Up

In Table D.7 page 149 the average running time from 1000 test runs for the B&P Method for the general SIC problem with trade-up are seen.

In Table D.8 page 150 the number of times where one value of ϵ yields higher revenue than another is seen for different capacities and test sets for 1000 test runs. The values of ϵ are $\epsilon_1 = 0.1$, $\epsilon_2 = 0.2$, $\epsilon_3 = 0.3$, $\epsilon_4 = 0.4$ and $\epsilon_5 = 0.5$.

D.2.2 The Simplified Methods

The You Method

In Table D.9 page 151 the average running time from 1000 test runs for the simplified You solution method are seen.

In Table D.10 page 151 the number of times where one value of ϵ yields a higher revenue than another is seen for different capacities and test sets for 1000 test runs. The values of ϵ are $\epsilon_1 = 0.001$, $\epsilon_1 = 0.01$, $\epsilon_1 = 0.1$ and $\epsilon_1 = 0.2$.

The HM Method

In Table D.11 page 152 the average running time from 1000 test runs for the HM method are seen.

In Table D.12 page 152 the number of times where one value of ϵ yields a higher revenue than another is seen for different capacities and test sets for 1000 test runs. The values of ϵ are $\epsilon_1 = 0.2$, $\epsilon_1 = 0.3$, $\epsilon_1 = 0.4$ and $\epsilon_1 = 0.5$.

The B&P Method with Trade-Up

In Table D.13 page 153 the average running time from 1000 test runs for the simplified B&P solution method are seen.

In Table D.14 page 154 the number of times where one value of ϵ yields higher revenue than another is seen for different capacities and test sets for 1000 test runs. The values of ϵ are $\epsilon_1 = 0.1$, $\epsilon_2 = 0.2$, $\epsilon_3 = 0.3$, $\epsilon_4 = 0.3$ and $\epsilon_5 = 0.5$.

		No. of Times the Inequality is Fulfilled					
	Test	$\epsilon_1 > \epsilon_2$	$\epsilon_1 > \epsilon_3$	$\epsilon_1 > \epsilon_4$	$\epsilon_2 > \epsilon_3$	$\epsilon_2 > \epsilon_4$	$\epsilon_3 > \epsilon_4$
Capacity	Set	$\epsilon_1 < \epsilon_2$	$\epsilon_1 < \epsilon_3$	$\epsilon_1 < \epsilon_4$	$\epsilon_2 < \epsilon_3$	$\epsilon_2 < \epsilon_4$	$\epsilon_3 < \epsilon_4$
C=80	L&H 1	607	678	711	720	808	772
		388	322	288	256	189	197
	L&H 2	555	651	702	708	782	773
		439	349	296	260	213	201
	L&H 3	532	624	712	689	787	767
		462	371	287	288	208	199
	L&H 4	637	731	800	816	868	863
		359	267	198	174	129	122
	L&H 5	631	703	763	721	820	819
		353	291	235	232	176	152
	L&Hup 1	407	509	549	550	645	607
		584	491	451	444	353	368
	L&Hup 2	476	565	634	594	692	628
		515	433	364	393	298	322
	L&Hup 3	433	552	585	582	606	468
		558	448	413	380	384	493
	L&Hup 4	516	636	732	710	812	823
		481	361	267	272	188	171
	L&Hup 5	476	590	649	610	680	636
		515	405	347	346	310	282
C=100	L&H 1	596	689	760	762	856	849
		401	311	240	231	144	142
	L&H 2	581	673	753	746	826	815
		410	321	244	233	166	150
	L&H 3	430	479	516	422	499	421
		420	380	346	214	199	163
	L&H 4	464	497	525	470	498	411
		377	348	323	187	197	129
	L&H 5	542	650	703	712	761	672
		439	343	290	241	204	147
	L&Hup 1	453	573	677	609	741	724
		546	423	323	384	258	269
	L&Hup 2	464	572	672	660	751	730
		530	421	325	309	237	224
	L&Hup 3	438	480	495	441	484	372
		535	495	483	264	278	307
	L&Hup 4	422	490	510	448	485	362
		525	461	442	268	240	159
	L&Hup 5	444	580	654	674	743	686
		541	413	342	317	249	202
C=120	L&H 1	563	677	763	800	872	873
		433	323	236	188	124	117
	L&H 2	419	461	476	414	437	338
		519	482	467	161	169	112
	L&Hup 1	455	591	695	656	782	778
		543	408	304	335	215	202
	L&Hup 2	402	454	477	409	442	331
		585	541	518	261	242	147
C=140	L&H 1	534	691	760	782	840	836
		458	306	238	208	154	120
	L&Hup 1	407	598	725	717	814	825
		579	396	274	276	182	147

Table D.2: Tuning of ϵ for the L&H Method with 1000 Test Runs.

		Average Running Time			
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.1$	$\epsilon_3 = 0.2$	$\epsilon_4 = 0.3$
C = 80	B&P 1	4.69 s	0.46 s	0.30 s	0.25 s
	B&P 2	4.12 s	0.43 s	0.27 s	0.24 s
	B&P 3	4.00 s	0.39 s	0.27 s	0.22 s
	B&P 4	3.90 s	0.40 s	0.27 s	0.22 s
	B&P 5	3.96 s	0.38 s	0.28 s	0.23 s
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.1$	$\epsilon_3 = 0.2$	$\epsilon_4 = 0.3$
C = 100	B&P 1	6.22 s	0.58 s	0.40 s	0.32 s
	B&P 2	5.47 s	0.53 s	0.35 s	0.30 s
	B&P 3	5.32 s	0.51 s	0.35 s	0.28 s
	B&P 4	5.10 s	0.49 s	0.35 s	0.28 s
	B&P 5	5.09 s	0.50 s	0.32 s	0.29 s
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.1$	$\epsilon_3 = 0.2$	$\epsilon_4 = 0.3$
C = 120	B&P 1	7.87 s	0.72 s	0.48 s	0.38 s
	B&P 2	6.86 s	0.63 s	0.43 s	0.34 s
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.1$	$\epsilon_3 = 0.2$	$\epsilon_4 = 0.3$
C=140	B&P 1	9.55 s	0.87 s	0.57 s	0.46 s

Table D.3: Running Times of the B&P Method with Different values of ϵ.

		No. of Times the Inequality is Fulfilled					
Capacity	Test Set	$\epsilon_1 > \epsilon_2$ $\epsilon_1 < \epsilon_2$	$\epsilon_1 > \epsilon_3$ $\epsilon_1 < \epsilon_3$	$\epsilon_1 > \epsilon_4$ $\epsilon_1 < \epsilon_4$	$\epsilon_2 > \epsilon_3$ $\epsilon_2 < \epsilon_3$	$\epsilon_2 > \epsilon_4$ $\epsilon_2 < \epsilon_4$	$\epsilon_3 > \epsilon_4$ $\epsilon_3 < \epsilon_4$
C=80	B&P 1	212 210	284 260	340 333	219 199	223 236	192 242
	B&P 2	195 184	269 237	295 299	205 169	215 193	213 205
	B&P 3	171 172	268 243	339 322	208 177	257 222	229 218
	B&P 4	159 157	221 193	270 241	171 139	197 150	209 196
	B&P 5	175 171	263 225	296 274	208 162	236 185	209 192
C=100	B&P 1	192 211	248 259	285 312	191 179	206 195	193 196
	B&P 2	174 183	243 267	266 270	192 168	213 161	204 176
	B&P 3	132 114	211 159	239 231	151 105	193 157	159 174
	B&P 4	137 125	181 164	225 209	124 105	166 137	157 142
	B&P 5	182 164	268 232	305 294	212 181	253 193	212 185
C=120	B&P 1	130 180	205 248	206 286	157 155	156 178	159 183
	B&P 2	144 130	202 192	219 190	138 135	150 130	174 136
C=140	B&P 1	184 181	269 268	269 313	208 193	204 195	207 217

Table D.4: Tuning of ϵ for the B&P Method with 1000 Test Runs.

		Average Running Time			
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.01$	$\epsilon_3 = 0.1$	$\epsilon_4 = 0.2$
C = 25	YT 1	46.88 s	8.44 s	2.19 s	1.46 s
	YT 2	23.76 s	5.71 s	1.59 s	0.98 s
	YT 3	22.14 s	5.89 s	1.29 s	1.07 s
	YT 4	10.29 s	2.90 s	0.89 s	0.61 s
	YT 5	24.99 s	5.60 s	1.52 s	1.02 s
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.01$	$\epsilon_3 = 0.1$	$\epsilon_4 = 0.2$
C = 40	YT 1	83.53 s	15.01 s	4.22 s	3.10 s
	YT 2	44.82 s	9.42 s	2.70 s	1.56 s
	YT 3	38.36 s	10.34 s	2.53 s	2.07 s
	YT 5	43.51 s	9.00 s	2.35 s	1.60 s
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.01$	$\epsilon_3 = 0.1$	$\epsilon_4 = 0.2$
C = 55	YT 1	113.87 s	20.20 s	4.92 s	3.43 s
	YT 2	65.40 s	13.52 s	3.70 s	2.42 s
	YT 5	63.20 s	13.04 s	3.30 s	2.23 s
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.01$	$\epsilon_3 = 0.1$	$\epsilon_4 = 0.2$
C = 75	YT 1	146.41 s	27.46 s	6.47 s	4.14 s

Table D.5: Running Times for the You Method with Different Values of ϵ.

		No. of Times the Inequality is Fulfilled					
Capacity	Test Set	$\epsilon_1 > \epsilon_2$ $\epsilon_1 < \epsilon_2$	$\epsilon_1 > \epsilon_3$ $\epsilon_1 < \epsilon_3$	$\epsilon_1 > \epsilon_4$ $\epsilon_1 < \epsilon_4$	$\epsilon_2 > \epsilon_3$ $\epsilon_2 < \epsilon_3$	$\epsilon_2 > \epsilon_4$ $\epsilon_2 < \epsilon_4$	$\epsilon_3 > \epsilon_4$ $\epsilon_3 < \epsilon_4$
C=25	YT 1	7 5	39 21	89 31	40 19	90 29	76 20
	YT 2	426 350	509 487	513 485	518 464	543 454	560 413
	YT 3	347 370	442 513	525 469	435 490	534 447	466 366
	YT 4	312 595	560 435	664 335	590 410	679 317	688 265
	YT 5	398 403	527 460	606 391	563 434	620 379	658 330
C=40	YT 1	389 595	586 413	674 325	606 383	697 302	745 229
	YT 2	347 291	458 531	517 476	482 501	539 453	605 368
	YT 3	380 514	438 527	447 533	457 496	456 507	426 361
	YT 5	392 570	497 500	555 440	517 469	583 412	657 319
C=55	YT 1	340 501	528 470	647 351	542 456	667 329	716 281
	YT 2	387 412	394 453	386 468	362 349	364 362	209 176
	YT 5	270 306	341 366	363 350	323 277	347 275	267 158
C=75	YT 1	335 444	453 480	480 455	493 411	508 400	309 126

Table D.6: Tuning of ϵ for the General You Method.

		Average Running Time				
Capacity	Test Set	$\epsilon_1 = 0.1$	$\epsilon_2 = 0.2$	$\epsilon_3 = 0.3$	$\epsilon_4 = 0.4$	$\epsilon_5 = 0.5$
C = 25	BP 1	0.48 s	0.10 s	0.29 s	0.09 s	0.09 s
	BP 2	0.13 s	0.10 s	0.10 s	0.08 s	0.09 s
	BP 3	0.12 s	0.09 s	0.08 s	0.08 s	0.08 s
	BP 4	0.16 s	0.16 s	0.08 s	0.07 s	0.07 s
	BP 5	0.17 s	0.11 s	0.10 s	0.08 s	0.09 s
Capacity	Test Set	$\epsilon_1 = 0.1$	$\epsilon_2 = 0.2$	$\epsilon_3 = 0.3$	$\epsilon_4 = 0.4$	$\epsilon_5 = 0.5$
C = 40	BP 1	0.53 s	0.14 s	0.15 s	0.11 s	0.11 s
	BP 2	0.17 s	0.12 s	0.11 s	0.10 s	0.11 s
	BP 3	0.15 s	0.11 s	0.10 s	0.10 s	0.09 s
	BP 5	0.21 s	0.14 s	0.12 s	0.17 s	0.09 s
Capacity	Test Set	$\epsilon_1 = 0.1$	$\epsilon_2 = 0.2$	$\epsilon_3 = 0.3$	$\epsilon_4 = 0.4$	$\epsilon_5 = 0.5$
C = 55	BP 1	0.59 s	0.17 s	0.21 s	0.26 s	0.12 s
	BP 2	0.22 s	0.16 s	0.13 s	0.12 s	0.12 s
	BP 5	0.25 s	0.17 s	0.15 s	0.11 s	0.11 s
Capacity	Test Set	$\epsilon_1 = 0.1$	$\epsilon_2 = 0.2$	$\epsilon_3 = 0.3$	$\epsilon_4 = 0.4$	$\epsilon_5 = 0.5$
C = 75	BP 1	0.69 s	0.22 s	0.22 s	0.16 s	0.15 s

Table D.7: Running Times for the B&P Method with TU for Different Values of ϵ.

Test Set	No. of Times the Inequality is Fulfilled									
	$\epsilon_1 > \epsilon_2$	$\epsilon_1 > \epsilon_3$	$\epsilon_1 > \epsilon_4$	$\epsilon_1 > \epsilon_5$	$\epsilon_2 > \epsilon_3$	$\epsilon_2 > \epsilon_4$	$\epsilon_2 > \epsilon_5$	$\epsilon_3 > \epsilon_4$	$\epsilon_3 > \epsilon_5$	$\epsilon_4 > \epsilon_5$
	$\epsilon_1 < \epsilon_2$	$\epsilon_1 < \epsilon_3$	$\epsilon_1 < \epsilon_4$	$\epsilon_1 < \epsilon_5$	$\epsilon_2 < \epsilon_3$	$\epsilon_2 < \epsilon_4$	$\epsilon_2 < \epsilon_5$	$\epsilon_3 < \epsilon_4$	$\epsilon_3 < \epsilon_5$	$\epsilon_4 < \epsilon_5$
	C = 25									
BP 1	69	87	110	111	80	103	98	79	90	69
	33	34	36	22	43	34	17	48	48	60
BP 2	285	378	412	442	298	339	395	235	321	262
	196	170	180	157	178	158	127	175	154	152
BP 3	282	353	369	423	259	280	371	232	347	320
	150	143	162	168	184	156	169	183	192	191
BP 4	220	221	258	298	84	165	193	140	197	109
	48	36	39	41	84	67	51	49	52	52
BP 5	342	442	502	561	382	432	523	354	501	439
	168	146	172	211	204	195	224	246	224	228
	C = 40									
BP 1	363	442	536	558	362	487	509	434	450	331
	153	188	199	166	232	216	167	200	222	262
BP 2	239	275	306	351	227	264	339	222	298	268
	171	161	169	192	169	161	165	173	153	168
BP 3	309	344	381	419	303	334	328	266	352	296
	184	153	175	180	256	204	172	226	220	214
BP 5	285	364	393	486	316	347	449	305	441	354
	143	123	157	192	191	176	210	233	221	228
	C = 55									
BP 1	331	446	530	534	394	478	511	389	426	326
	176	192	213	173	242	224	151	211	218	277
BP 2	184	224	287	335	224	285	320	217	294	257
	155	119	135	130	157	158	122	152	148	181
BP 5	283	354	435	424	274	388	388	348	362	295
	157	163	190	190	206	164	208	198	214	256
	C = 75									
BP 1	299	373	434	476	322	415	455	341	395	317
	180	179	198	198	182	198	198	196	229	303

Table D.8: Tuning of ϵ for the General B&P Method with TU.

		Average Running Time			
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.01$	$\epsilon_3 = 0.1$	$\epsilon_4 = 0.2$
C = 25	You 1	2.35 s	0.65 s	0.24 s	0.17 s
	You 2	1.51 s	0.45 s	0.17 s	0.13 s
	You 3	1.28 s	0.39 s	0.19 s	0.12 s
	You 4	0.86 s	0.30 s	0.14 s	0.10 s
	You 5	1.55 s	0.45 s	0.18 s	0.14 s
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.01$	$\epsilon_3 = 0.1$	$\epsilon_4 = 0.2$
C = 40	You 1	3.69 s	1.01 s	0.31 s	0.23 s
	You 2	2.26 s	0.64 s	0.23 s	0.20 s
	You 3	2.00 s	0.59 s	0.24 s	0.15 s
	You 5	2.48 s	0.70 s	0.24 s	0.18 s
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.01$	$\epsilon_3 = 0.1$	$\epsilon_4 = 0.2$
C = 55	You 1	4.99 s	1.28 s	0.42 s	0.27 s
	You 2	3.01 s	0.88 s	0.29 s	0.21,s
	You 5	3.22 s	0.90 s	0.30 s	0.22 s
Capacity	Test Set	$\epsilon_1 = 0.001$	$\epsilon_2 = 0.01$	$\epsilon_3 = 0.1$	$\epsilon_4 = 0.2$
C = 75	You 1	6.57 s	1.71 s	0.51 s	0.35 s

Table D.9: Running Times of the Simplified You Method for Different Values of ϵ.

		No. of Times the Inequality is Fulfilled					
Capacity	Test Set	$\epsilon_1 > \epsilon_2$ $\epsilon_1 < \epsilon_2$	$\epsilon_1 > \epsilon_3$ $\epsilon_1 < \epsilon_3$	$\epsilon_1 > \epsilon_4$ $\epsilon_1 < \epsilon_4$	$\epsilon_2 > \epsilon_3$ $\epsilon_2 < \epsilon_3$	$\epsilon_2 > \epsilon_4$ $\epsilon_2 < \epsilon_4$	$\epsilon_3 > \epsilon_4$ $\epsilon_3 < \epsilon_4$
C=25	You 1	338 632	597 390	729 263	686 304	772 217	798 198
	You 2	500 314	641 352	730 264	658 338	738 256	721 241
	You 3	519 416	551 383	646 329	596 338	637 282	586 335
	You 4	127 342	252 408	290 368	225 326	275 304	132 80
	You 5	482 458	670 316	751 239	693 289	781 210	730 209
C=40	You 1	382 491	733 248	821 169	761 225	836 150	781 159
	You 2	364 271	436 330	457 310	385 309	421 276	340 141
	You 3	248 163	255 231	263 233	211 207	219 204	121 124
	You 5	399 354	488 356	501 345	442 314	461 302	331 199
C=55	You 1	420 356	371 414	403 384	292 333	341 292	233 194
	You 2	29 27	30 42	33 40	22 25	28 20	15 4
	You 5	20 16	24 23	24 23	14 21	17 18	13 7
C=75	You 1	6 8	7 7	8 6	6 4	7 3	5 3

Table D.10: Tuning of ϵ for the Simplified You Method.

		Average Running Time			
Capacity	Test Set	$\epsilon_1 = 0.2$	$\epsilon_2 = 0.3$	$\epsilon_3 = 0.4$	$\epsilon_4 = 0.5$
C = 25	HM 1	0.89 s	0.68 s	0.68 s	0.77 s
	HM 2	0.76 s	0.65 s	0.54 s	0.52 s
	HM 3	0.72 s	0.74 s	0.55 s	0.53 s
	HM 4	0.52 s	0.50 s	0.42 s	0.44 s
	HM 5	0.71 s	0.58 s	0.60 s	0.58 s
Capacity	Test Set	$\epsilon_1 = 0.2$	$\epsilon_2 = 0.3$	$\epsilon_3 = 0.4$	$\epsilon_4 = 0.5$
C = 40	HM 1	1.42 s	1.33 s	1.02 s	1.01 s
	HM 2	1.02 s	0.85 s	0.88 s	0.72 s
	HM 3	0.88 s	0.74 s	0.79 s	0.64 s
	HM 5	0.99 s	0.81 s	0.73 s	0.68 s
Capacity	Test Set	$\epsilon_1 = 0.2$	$\epsilon_2 = 0.3$	$\epsilon_3 = 0.4$	$\epsilon_4 = 0.5$
C = 55	HM 1	1.79 s	1.40 s	1.35 s	1.14 s
	HM 2	1.27 s	1.17 s	0.90 s	0.85 s
	HM 5	1.26 s	1.03 s	0.92 s	0.82 s
Capacity	Test Set	$\epsilon_1 = 0.2$	$\epsilon_2 = 0.3$	$\epsilon_3 = 0.4$	$\epsilon_4 = 0.5$
C = 75	HM 1	2.26 s	1.96 s	1.58 s	1.41 s

Table D.11: Running Times for the HM Method with Different Values of ϵ.

		No. of Times the Inequality is Fulfilled					
Capacity	Test Set	$\epsilon_1 > \epsilon_2$ $\epsilon_1 < \epsilon_2$	$\epsilon_1 > \epsilon_3$ $\epsilon_1 < \epsilon_3$	$\epsilon_1 > \epsilon_4$ $\epsilon_1 < \epsilon_4$	$\epsilon_2 > \epsilon_3$ $\epsilon_2 < \epsilon_3$	$\epsilon_2 > \epsilon_4$ $\epsilon_2 < \epsilon_4$	$\epsilon_3 > \epsilon_4$ $\epsilon_3 < \epsilon_4$
C=25	HM 1	111 84	186 159	145 233	144 136	124 226	141 251
	HM 2	119 173	218 262	233 251	170 191	203 204	157 136
	HM 3	170 263	232 429	293 470	211 370	252 457	224 354
	HM 4	88 137	150 159	153 161	64 27	67 33	5 8
	HM 5	98 117	76 188	160 302	142 210	177 301	158 235
C=40	HM 1	161 157	230 239	214 344	192 227	172 333	161 288
	HM 2	152 221	219 302	300 352	206 271	289 358	245 343
	HM 3	40 81	91 137	94 165	98 117	97 143	10 44
	HM 5	109 152	91 240	238 355	137 223	205 337	207 314
C=55	HM 1	293 389	358 395	368 437	300 375	302 440	290 366
	HM 2	9 17	16 39	23 29	18 30	26 26	31 19
	HM 5	5 9	3 11	7 23	12 9	10 19	7 19
C=75	HM 1	4 2	3 4	4 5	0 2	1 5	1 4

Table D.12: Tuning of ϵ for the HM Method.

		Average Running Time				
Capacity	Test Set	$\epsilon_1 = 0.1$	$\epsilon_2 = 0.2$	$\epsilon_3 = 0.3$	$\epsilon_4 = 0.4$	$\epsilon_5 = 0.5$
C = 25	BP 1	0.21 s	0.13 s	0.10 s	0.09 s	0.09 s
	BP 2	0.17 s	0.10 s	0.09 s	0.08 s	0.10 s
	BP 3	0.20 s	0.09 s	0.09 s	0.08 s	0.08 s
	BP 4	0.16 s	0.08 s	0.08 s	0.07 s	0.08 s
	BP 5	0.17 s	0.11 s	0.09 s	0.08 s	0.09 s
Capacity	Test Set	$\epsilon_1 = 0.1$	$\epsilon_2 = 0.2$	$\epsilon_3 = 0.3$	$\epsilon_4 = 0.4$	$\epsilon_5 = 0.5$
C = 40	BP 1	0.21 s	0.15 s	0.13 s	0.11 s	0.12 s
	BP 2	0.16 s	0.13 s	0.11 s	0.10 s	0.11 s
	BP 3	0.19 s	0.11 s	0.11 s	0.10 s	0.10 s
	BP 5	0.17 s	0.13 s	0.12 s	0.10 s	0.10 s
Capacity	Test Set	$\epsilon_1 = 0.1$	$\epsilon_2 = 0.2$	$\epsilon_3 = 0.3$	$\epsilon_4 = 0.4$	$\epsilon_5 = 0.5$
C = 55	BP 1	0.31 s	0.20 s	0.16 s	0.14 s	0.14 s
	BP 2	0.21 s	0.15 s	0.13 s	0.12 s	0.12 s
	BP 5	0.23 s	0.16 s	0.13 s	0.11 s	0.11 s
Capacity	Test Set	$\epsilon_1 = 0.1$	$\epsilon_2 = 0.2$	$\epsilon_3 = 0.3$	$\epsilon_4 = 0.4$	$\epsilon_5 = 0.5$
C = 75	BP 1	0.34 s	0.24 s	0.20 s	0.17 s	0.16 s

Table D.13: Running Times of the Simplified B&P Method with TU for Different Values of ϵ.

Test Set	No. of Times the Inequality is Fulfilled									
	$\epsilon_1 > \epsilon_2$ $\epsilon_1 < \epsilon_2$	$\epsilon_1 > \epsilon_3$ $\epsilon_1 < \epsilon_3$	$\epsilon_1 > \epsilon_4$ $\epsilon_1 < \epsilon_4$	$\epsilon_1 > \epsilon_5$ $\epsilon_1 < \epsilon_5$	$\epsilon_2 > \epsilon_3$ $\epsilon_2 < \epsilon_3$	$\epsilon_2 > \epsilon_4$ $\epsilon_2 < \epsilon_4$	$\epsilon_2 > \epsilon_5$ $\epsilon_2 < \epsilon_5$	$\epsilon_3 > \epsilon_4$ $\epsilon_3 < \epsilon_4$	$\epsilon_3 > \epsilon_5$ $\epsilon_3 < \epsilon_5$	$\epsilon_4 > \epsilon_5$ $\epsilon_4 < \epsilon_5$
	C = 25									
BP 1	498 260	552 295	614 244	638 281	419 342	529 282	625 236	491 352	597 281	509 363
BP 2	339 283	452 271	506 251	494 264	394 296	472 262	455 274	393 254	394 335	317 385
BP 3	422 273	539 231	542 250	545 253	458 287	466 257	466 242	356 341	362 339	291 285
BP 4	545 252	576 206	588 246	581 282	430 301	513 302	550 289	447 289	516 285	431 244
BP 5	513 336	573 274	548 333	589 327	486 358	499 346	581 320	435 412	507 352	500 333
	C = 40									
BP 1	544 300	616 325	632 283	622 328	514 377	550 353	589 332	470 439	535 401	508 402
BP 2	451 321	520 287	519 312	511 354	444 320	487 331	474 333	403 336	410 405	366 451
BP 3	433 305	533 245	523 289	550 348	463 280	462 250	553 294	405 320	455 348	437 349
BP 5	495 352	564 270	557 358	587 331	478 364	554 347	594 331	490 393	539 361	512 332
	C = 55									
BP 1	470 289	516 324	567 274	575 302	443 347	509 302	576 292	463 339	526 323	483 362
BP 2	228 143	286 160	313 127	306 142	247 177	272 143	254 141	244 142	236 206	153 241
BP 5	323 219	382 154	382 169	421 196	325 215	350 205	403 219	268 240	342 234	319 224
	C = 75									
BP 1	458 227	494 261	499 272	500 277	408 300	447 269	504 256	399 326	433 308	415 319

Table D.14: Tuning of ϵ for the Simplified B&P Method with TU.

Appendix E

Results for the SIC Problem without Trade-Up

For all tables in this chapter, which show the relative difference in percentage for two methods the following holds. If a number different from zero appears in the tables, it cannot be rejected that the difference is greater than zero, i.e., it cannot be rejected that one method is significantly better than another with $\alpha = 5\%$. A bold number indicates that the method listed in the row in the top of the table is significantly better than the method listed in the column. If the number is not bold the method listed in the column yields a significantly higher revenue than the method in th top row. The symbol *NoD* means that no difference between the methods was observed for all 1000 runs.

In Table E.1 page 156 the average running times from 1000 test runs for six different solution methods for the SIC problem without trade-up with $\epsilon_1 = 0.01$ and $\epsilon_2 = 0.1$ are seen for test sets 2, 3 and 5 and all capacities.

The update of the booking limits is in Tables E.1 page 156, E.2 page 157, E.3 page 158 and E.4 page 159 are done at some prespecified times given by the airline and not in each decision period. The times of these updates are given in the vector *DecPer* and the times at which each data interval begins are given in the vector *DataintStart*. These vectors are given by

$$\begin{aligned} DataintStart &= [150, 100, 70, 49, 42, 35, 28, 21, 14, 10, 7, 5, 3, 2, 1, 0] \\ DecPer &= [150, 143, 136, 129, 122, 115, 108, 101, 96, 91, 86, 81, 76, 71, 68, \\ &\quad 65, 62, 59, 56, 53, 50, 48, 46, 44, 42, 40, 38, 36, 34, 32, 30, 28, 27, \\ &\quad 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, \\ &\quad 8, 7, 6, 5, 4, 3, 2, 1]. \end{aligned}$$

In Tables E.2 page 157, E.3 page 158 and E.4 page 159 the relative dif-

	$C = 80$		$C = 100$		$C = 120$		$C = 140$	
	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2
L&H 2	2.91 s	0.85 s	3.54 s	1.03 s	4.24 s	1.25 s	4.80 s	1.40 s
L&Hup 2	2.93 s	0.86 s	3.56 s	1.04 s	4.26 s	1.26 s	4.81 s	1.42 s
B&P 2	1.38 s	0.41 s	1.72 s	0.49 s	2.12 s	0.63 s	2.51 s	0.71 s
B&Pup 2	0.31 s	0.19 s	0.32 s	0.20 s	0.34 s	0.22 s	0.34 s	0.23 s
EMSRa 2	1.00 s	0.99 s	1.02 s	1.01 s	1.04 s	1.03 s	1.05 s	1.04 s
EMSRb 2	1.04 s	1.03 s	1.05 s	1.04 s	1.08 s	1.07 s	1.08 s	1.07 s
L&H 3	4.80 s	1.40 s	3.13 s	0.91 s	3.69 s	1.07 s	4.19 s	1.22 s
L&Hup 3	4.81 s	1.42 s	3.14 s	0.92 s	3.70 s	1.09 s	4.20 s	1.23 s
B&P 3	2.51 s	0.71 s	1.61 s	0.49 s	1.99 s	0.57 s	2.31 s	0.68 s
B&Pup 3	0.34 s	0.23 s	0.30 s	0.21 s	0.32 s	0.22 s	0.33 s	0.24 s
EMSRa 3	1.05 s	1.04 s	0.95 s	0.95 s	0.99 s	0.98 s	1.01 s	1.00 s
EMSRb 3	1.08 s	1.07 s	0.99 s	0.98 s	1.02 s	1.01 s	1.04 s	1.03 s
L&H 5	2.64 s	0.77 s	3.21 s	0.93 s	3.81 s	1.11 s	4.42 s	1.27 s
L&Hup 5	2.66 s	0.78 s	3.23 s	0.95 s	3.82 s	1.12 s	4.43 s	1.28 s
B&P 5	1.29 s	0.36 s	1.57 s	0.46 s	1.92 s	0.57 s	2.30 s	0.66 s
B&Pup 5	0.29 s	0.18 s	0.30 s	0.19 s	0.31 s	0.20 s	0.33 s	0.22 s
EMSRa 5	0.92 s	0.92 s	0.95 s	0.94 s	0.97 s	0.97 s	1.00 s	1.00 s
EMSRb 5	0.96 s	0.96 s	0.98 s	0.98 s	1.01 s	1.00 s	1.04 s	1.03 s

Table E.1: Running Times for the Problem without Trade-Up, Test Sets 2,3 and 5.

ferences in % from running the different methods 1000 times on test sets 2, 3 and 5, respectively, are seen.

In Table E.5 page 159 the running times for the EMSRa and EMSRb methods when these are updated in each decision period can be seen. The running times are measured for all test sets and different capacities and with $\epsilon_1 = 0.01$ and $\epsilon_2 = 0.1$.

In Tables E.6 page 160, E.7 page 160, E.8 page 161, E.9 page 161 and E.10 page 162 the results from running the methods 1000 times on each test set, are seen. The booking limits are updated in each decision period, hence only results for L&H, B&P, EMSRa and EMSRb are shown since in this case L&H and B&P are the same as L&Hup and B&Pup, respectively.

	L&Hup 2		**B&P 2**		**B&Pup 2**		**EMSRa 2**		**EMSRb 2**	
	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2
C=80										
L&H 2	1.62	1.54	0.11	**0.20**	1.61	1.32	3.66	3.28	0.45	0
L&Hup 2	-	-	**1.61**	**1.32**	0	**0.23**	2.03	1.75	**1.18**	**1.43**
B&P 2	-	-	-	-	1.50	1.51	3.54	3.51	0.35	0.33
B&Pup 2	-	-	-	-	-	-	2.04	2.00	**1.17**	**1.20**
EMSRa 2	-	-	-	-	-	-	-	-	**3.20**	**3.19**
C=100										
L&H 2	1.16	1.10	0.16	**0.32**	1.27	0.83	2.07	1.53	0.38	**0.10**
L&Hup 2	-	-	**1.27**	**0.82**	0.11	**0.28**	0.92	0.45	**0.79**	**1.20**
B&P 2	-	-	-	-	1.12	1.14	1.90	1.87	0.23	0.24
B&Pup 2	-	-	-	-	-	-	0.80	0.74	**0.90**	**0.91**
EMSRa 2	-	-	-	-	-	-	-	-	**1.69**	**1.65**
C=120										
L&H 2	0.81	0.81	0.20	**0.21**	1.05	0.65	0.47	0	0.16	**0.26**
L&Hup 2	-	-	**1.04**	**0.65**	0.23	**0.17**	**0.37**	**0.78**	**0.66**	**1.08**
B&P 2	-	-	-	-	0.85	0.86	0.27	0.27	**0.04**	**0.05**
B&Pup 2	-	-	-	-	-	-	**0.60**	**0.61**	**0.89**	**0.90**
EMSRa 2	-	-	-	-	-	-	-	-	**0.31**	**0.31**
C=140										
L&H 2	0.22	0.18	0.01	0.01	0.21	0.20	0.04	0.04	0.03	0.03
L&Hup 2	-	-	**0.21**	**0.20**	0	0.02	**0.17**	**0.14**	**0.18**	**0.15**
B&P 2	-	-	-	-	0.20	0.20	0.03	0.03	0.02	0
B&Pup 2	-	-	-	-	-	-	**0.17**	**0.16**	**0.18**	**0.18**
EMSRa 2	-	-	-	-	-	-	-	-	0	0

Table E.2: Rel. Diff. in % for Test Set 2 without Trade-Up.

	L&Hup 3		B&P 3		B&Pup 3		EMSRa 3		EMSRb 3	
	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2
C=80										
L&H 3	1.08	0.92	0.48	**0.16**	1.01	0.36	4.13	3.41	0.42	**0.28**
L&Hup 3	-	-	**1.00**	**0.35**	0	**0.57**	3.06	2.52	**0.67**	**1.19**
B&P 3	-	-	-	-	0.54	0.51	3.61	3.58	**0.07**	**0.12**
B&Pup 3	-	-	-	-	-	-	3.10	3.10	**0.60**	**0.62**
EMSRa 3	-	-	-	-	-	-	-	-	**3.70**	**3.73**
C=100										
L&H 3	0.79	0.74	0.40	**0.05**	0.83	0.36	2.35	1.82	0.27	**0.24**
L&Hup 3	-	-	**0.83**	**0.36**	0.04	**0.38**	1.56	1.10	**0.53**	**0.97**
B&P 3	-	-	-	-	0.44	0.41	1.92	1.87	**0.14**	**0.18**
B&Pup 3	-	-	-	-	-	-	1.50	1.48	**0.57**	**0.59**
EMSRa 3	-	-	-	-	-	-	-	-	**2.08**	**2.08**
C=120										
L&H 3	0.19	0.14	0.03	0	0.13	0.10	0.17	0.15	0.06	0.04
L&Hup 3	-	-	**0.13**	**0.10**	**0.06**	**0.04**	0	0	**0.13**	**0.10**
B&P 3	-	-	-	-	0.10	0.10	0.14	0.14	0.04	0.04
B&Pup 3	-	-	-	-	-	-	0	0.05	**0.06**	**0.06**
EMSRa 3	-	-	-	-	-	-	-	-	**0.10**	**0.11**
C=140										
L&H 3	0.00	0.00	NoD	NoD	0.00	0.00	NoD	NoD	0	0
L&Hup 3	-	-	**0.00**	**0.00**	0	0	**0.00**	**0.00**	**0.00**	**0.00**
B&P 3	-	-	-	-	0.00	0.00	NoD	NoD	0	0
B&Pup 3	-	-	-	-	-	-	**0.00**	**0.00**	0	0
EMSRa 3	-	-	-	-	-	-	-	-	0	0

Table E.3: Rel. Diff. in % for Test Set 3 without Trade-Up.

	L&Hup 5		B&P 5		B&Pup 5		EMSRa 5		EMSRb 5	
	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2
C=80										
L&H 5	1.78	1.37	0	**0.30**	1.60	1.21	4.99	4.60	0.50	0
L&Hup 5	-	-	**1.60**	**1.19**	**0.18**	**0.18**	3.20	3.24	**1.28**	**1.21**
B&P 5	-	-	-	-	1.55	1.49	4.96	4.96	0.46	0.46
B&Pup 5	-	-	-	-	-	-	3.39	3.45	**1.10**	**1.04**
EMSRa 5	-	-	-	-	-	-	-	-	**4.48**	**4.48**
C=100										
L&H 5	1.19	0.97	0.27	**0.28**	1.34	0.78	1.30	0.75	0.25	**0.31**
L&Hup 5	-	-	**1.34**	**0.77**	0.15	**0.20**	0	**0.27**	**0.95**	**1.28**
B&P 5	-	-	-	-	1.07	1.05	1.03	1.03	**0.03**	**0.03**
B&Pup 5	-	-	-	-	-	-	0	0	**1.10**	**1.08**
EMSRa 5	-	-	-	-	-	-	-	-	**1.05**	**1.06**
C=120										
L&H 5	0.56	0.41	0.06	**0.04**	0.63	0.49	0.24	0.13	0.09	0
L&Hup 5	-	-	**0.63**	**0.49**	0.07	0.08	**0.33**	**0.29**	**0.47**	**0.44**
B&P 5	-	-	-	-	0.57	0.54	0.18	0.17	0	0
B&Pup 5	-	-	-	-	-	-	**0.40**	**0.37**	**0.54**	**0.52**
EMSRa 5	-	-	-	-	-	-	-	-	**0.15**	**0.15**
C=140										
L&H 5	0.03	0.03	0	0	0.03	0.03	0.01	0.01	0.01	0.01
L&Hup 5	-	-	**0.03**	**0.03**	**0.00**	0.00	**0.03**	**0.02**	**0.03**	**0.02**
B&P 5	-	-	-	-	0.03	0.03	0	0.01	0.01	0.01
B&Pup 5	-	-	-	-	-	-	**0.02**	**0.02**	**0.02**	**0.02**
EMSRa 5	-	-	-	-	-	-	-	-	0	0

Table E.4: Rel. Diff. in % for Test Set 5 without Trade-Up.

	$C = 80$		$C = 100$		$C = 120$		$C = 140$	
	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2
EMSRa 1	3.32 s	2.79 s	3.45 s	2.89 s	3.58 s	3.00 s	3.61 s	3.03 s
EMSRb 1	3.54 s	2.97 s	3.64 s	3.06 s	3.77 s	3.16 s	3.77 s	3.16 s
EMSRa 2	2.46 s	2.07 s	2.59 s	2.18 s	2.70 s	2.27 s	2.80 s	2.35 s
EMSRb 2	2.65 s	2.23 s	2.76 s	2.32 s	2.82 s	2.37 s	2.91 s	2.44 s
EMSRa 3	2.12 s	1.79 s	2.23 s	1.87 s	2.32 s	1.95 s	2.41 s	2.03 s
EMSRb 3	2.26 s	1.91 s	2.34 s	1.97 s	2.41 s	2.03 s	2.49 s	2.10 s
EMSRa 4	2.16 s	1.82 s	2.23 s	1.88 s	2.35 s	1.98 s	2.40 s	2.02 s
EMSRb 4	2.31 s	1.94 s	2.36 s	1.98 s	2.44 s	2.05 s	2.49 s	2.10 s
EMSRa 5	2.21 s	1.87 s	2.44 s	2.06 s	2.68 s	2.27 s	2.73 s	2.31 s
EMSRb 5	2.41 s	2.04 s	2.58 s	2.17 s	2.80 s	2.36 s	2.84 s	2.39 s

Table E.5: Running Times with Update in each Decision Period.

	B&P 1		**EMSRa 1**		**EMSRb 1**	
	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2
C=80						
L&H 1	0.19	**0.06**	3.72	3.45	0.37	0
B&P 1	-	-	3.53	3.53	0.18	0.17
EMSRa 1	-	-	-	-	**3.35**	**3.37**
C=100						
L&H 1	0.16	**0.12**	2.16	1.89	0.24	**0.04**
B&P 1	-	-	2.00	2.03	0.09	0.09
EMSRa 1	-	-	-	-	**1.93**	**1.95**
C=120						
L&H 1	0.10	**0.34**	1.24	0.79	0.19	**0.27**
B&P 1	-	-	1.14	1.14	0.09	0
EMSRa 1	-	-	-	-	**1.06**	**1.07**
C=140						
L&H 1	0.26	**0.25**	0.51	**0.05**	0.21	**0.36**
B&P 1	-	-	0.25	0.22	**0.07**	**0.10**
EMSRa 1	-	-	-	-	**0.31**	**0.32**

Table E.6: Rel. Diff. in % for Test Set 1 with Update in each Decision Period.

	B&P 2		**EMSRa 2**		**EMSRb 2**	
	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2
C=80						
L&H 2	0.15	**0.09**	3.82	3.55	0.27	0
B&P 2	-	-	3.66	3.66	0.13	0.11
EMSRa 2	-	-	-	-	**3.54**	**3.56**
C=100						
L&H 2	0.14	**0.39**	2.34	1.79	0.15	**0.40**
B&P 2	-	-	2.19	2.20	0	0
EMSRa 2	-	-	-	-	**2.19**	**2.21**
C=120						
L&H 2	0.21	**0.16**	0.54	0.15	0.08	**0.30**
B&P 2	-	-	0.33	0.31	**0.14**	**0.14**
EMSRa 2	-	-	-	-	**0.46**	**0.44**
C=140						
L&H 2	0	0	0.08	0.06	0	0
B&P 2	-	-	0.07	0.06	0	0
EMSRa 2	-	-	-	-	**0.07**	**0.06**

Table E.7: Rel. Diff. in % for Test Set 2 with Update in each Decision Period.

	B&P 3		**EMSRa 3**		**EMSRb 3**	
	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2
C=80						
L&H 3	0.42	**0.11**	3.73	3.12	0.19	**0.41**
B&P 3	-	-	3.27	3.24	**0.24**	**0.30**
EMSRa 3	-	-	-	-	**3.54**	**3.57**
C=100						
L&H 3	0.35	**0.05**	1.99	1.59	0.12	**0.30**
B&P 3	-	-	1.63	1.64	**0.23**	**0.25**
EMSRa 3	-	-	-	-	**1.88**	**1.91**
C=120						
L&H 3	0.04	0	0.23	0.20	0.02	0
B&P 3	-	-	0.18	0.19	**0.02**	0
EMSRa 3	-	-	-	-	**0.21**	**0.20**
C=140						
L&H 3	0	NoD	0	0	0	NoD
B&P 3	-	-	0	0	0	NoD
EMSRa 3	-	-	-	-	0	0

Table E.8: Rel. Diff. in % for Test Set 3 with Update in each Decision Period.

	B&P 4		**EMSRa 4**		**EMSRb 4**	
	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2
C=80						
L&H 4	0.63	**0.58**	4.90	3.50	0.15	**1.15**
B&P 4	-	-	4.19	4.12	**0.49**	**0.56**
EMSRa 4	-	-	-	-	**4.74**	**4.74**
C=100						
L&H 4	0.26	**0.35**	2.24	1.54	0	**0.64**
B&P 4	-	-	1.96	1.90	**0.22**	**0.29**
EMSRa 4	-	-	-	-	**2.21**	**2.22**
C=120						
L&H 4	0.02	0	0.14	0.12	0.03	0
B&P 4	-	-	0.11	0.12	0	0
EMSRa 4	-	-	-	-	**0.11**	**0.11**
C=140						
L&H 4	NoD	NoD	NoD	NoD	NoD	NoD
B&P 4	-	-	NoD	NoD	NoD	NoD
EMSRa 4	-	-	-	-	NoD	NoD

Table E.9: Rel. Diff. in % for Test Set 4 with Update in each Decision Period.

	Relative Difference in Percentage					
	B&P 5		**EMSRa 5**		**EMSRb 5**	
	ϵ_1	ϵ_2	ϵ_1	ϵ_2	ϵ_1	ϵ_2
C=80						
L&H 5	0.11	**0.22**	4.62	4.29	0.29	**0.03**
B&P 5	-	-	4.52	4.56	0.18	0.20
EMSRa 5	-	-	-	-	**4.33**	**4.35**
C=100						
L&H 5	0.27	**0.29**	1.27	0.67	0.10	**0.49**
B&P 5	-	-	1.00	0.97	**0.18**	**0.20**
EMSRa 5	-	-	-	-	**1.18**	**1.16**
C=120						
L&H 5	0.07	0	0.27	0.19	0.06	**0.03**
B&P 5	-	-	0.20	0.20	0	**0.03**
EMSRa 5	-	-	-	-	**0.21**	**0.22**
C=140						
L&H 5	0.01	0	0.02	0.01	0	0
B&P 5	-	-	0.01	0.01	0	0
EMSRa 5	-	-	-	-	**0.01**	**0.01**

Table E.10: Rel. Diff. in % for Test Set 5 with Update in each Decision Period.

Appendix F

Results for the General SIC Problem with Trade-Up

For all tables in this chapter, which show the relative difference in percentage for two methods the following holds. If a number different from zero appears in the tables, it cannot be rejected that the difference is greater than zero, i.e., it cannot be rejected that one method is significantly better than another with $\alpha = 5\%$. A bold number indicates that the method listed in the row in the top of the table is significantly better than the method listed in the column. If the number is not bold the method listed in the column yields a significantly higher revenue than the method in th top row. The symbol *NoD* means that no difference between the methods was observed for all 1000 runs.

F.1 Comparison of the Hybrid Methods

In Tables F.1, F.2, F.3, F.4 page 166 and F.5 page 166 the relative differences in % when comparing the six different hybrid methods for the general SIC problem with trade-up are seen. There is one table for each test set.

F.2 Results

In Table F.6 page 167 the average running times from 1000 test runs for six different methods for the general SIC problem with trade-up are shown. The times are shown for test sets 2, 3 and 5.

In Tables F.7 page 167, F.8 page 168 and F.9 page 168 the relative differences in % from running the different methods 1000 times on test sets 2, 3 and 5, respectively, are seen.

	Max2	**AddCol1**	**AddCol2**	**Combi1**	**Combi2**
C=25					
Max1	**7.10**	4.76	**7.18**	**7.02**	8.49
Max2	-	11.99	**0.08**	0	15.58
AddCol1	-	-	**12.08**	**11.92**	3.60
AddCol2	-	-	-	0.14	15.67
Combi1	-	-	-	-	15.52
C=40					
Max1	**5.94**	11.96	0	0	18.59
Max2	-	17.95	3.98	6.15	24.34
AddCol1	-	-	**13.38**	**11.86**	5.53
AddCol2	-	-	-	2.31	19.39
Combi1	-	-	-	-	18.51
C=55					
Max1	**5.27**	12.63	0	0.18	18.18
Max2	-	18.50	4.55	5.45	24.19
AddCol1	-	-	**13.25**	**12.40**	4.64
AddCol2	-	-	-	1.19	18.42
Combi1	-	-	-	-	17.91
C=75					
Max1	**2.35**	6.53	**1.28**	0.14	**2.39**
Max2	-	9.17	1.10	2.48	0.39
AddCol1	-	-	**8.06**	**6.38**	**8.66**
AddCol2	-	-	-	1.42	0
Combi1	-	-	-	-	**2.52**

Table F.1: Rel. Diff. in % for Hybrid Methods for Test Set 1.

F.3 Comparison of Non-Trade-Up with Trade-Up Methods

In Tables F.10, F.11 and F.12 page 169 the relative differences in % from comparing the two best solution methods for the SIC problem without trade-up with the two best solution methods for the general SIC problem with trade-up 1000 times are seen for test sets 2, 3 and 5, respectively.

	Max2	**AddCol1**	**AddCol2**	**Combi1**	**Combi2**
C=25					
Max1	**8.91**	3.10	**6.94**	**2.46**	0
Max2	-	11.95	2.71	6.32	6.76
AddCol1	-	-	**9.33**	**5.64**	**5.73**
AddCol2	-	-	-	4.46	3.75
Combi1	-	-	-	-	1.40
C=40					
Max1	**1.34**	4.99	**2.37**	**0.09**	4.09
Max2	-	6.57	**1.43**	1.25	5.49
AddCol1	-	-	**7.50**	**5.15**	**1.27**
AddCol2	-	-	-	2.31	6.45
Combi1	-	-	-	-	4.25
C=55					
Max1	3.70	1.54	0	0	**1.30**
Max2	-	**2.53**	**3.00**	**3.89**	**4.99**
AddCol1	-	-	**1.06**	**1.67**	**2.64**
AddCol2	-	-	-	0	**2.08**
Combi1	-	-	-	-	**1.12**
C=75					
Max1	2.76	**1.46**	0.98	**1.46**	**2.57**
Max2	-	**4.28**	**1.92**	**4.28**	**5.33**
AddCol1	-	-	2.40	0	**1.10**
AddCol2	-	-	-	**2.40**	**3.46**
Combi1	-	-	-	-	**1.10**

Table F.2: Rel. Diff. in % for Hybrid Methods for Test Set 2.

	Max2	**AddCol1**	**AddCol2**	**Combi1**	**Combi2**
C=25					
Max1	**9.80**	7.04	**6.18**	**0.96**	0.87
Max2	-	17.35	3.72	8.94	9.99
AddCol1	-	-	**13.08**	**8.25**	**6.99**
AddCol2	-	-	-	5.41	5.92
Combi1	-	-	-	-	2.00
C=40					
Max1	**4.24**	8.61	**3.45**	0.28	1.02
Max2	-	13.46	0.91	4.53	5.17
AddCol1	-	-	**12.62**	**8.30**	**7.87**
AddCol2	-	-	-	3.72	4.36
Combi1	-	-	-	-	0.73
C=55					
Max1	**1.20**	0.99	1.66	0.03	**4.52**
Max2	-	2.22	2.32	1.25	**3.51**
AddCol1	-	-	**0.11**	**0.90**	**5.40**
AddCol2	-	-	-	0	**5.82**
Combi1	-	-	-	-	**4.47**
C=75					
Max1	**0.45**	**0.80**	2.48	**0.80**	**4.31**
Max2	-	0	2.37	0	**4.09**
AddCol1	-	-	3.29	0	**3.53**
AddCol2	-	-	-	**3.29**	**6.42**
Combi1	-	-	-	-	**3.53**

Table F.3: Rel. Diff. in % for Hybrid Methods for Test Set 3.

	Max2	AddCol1	AddCol2	Combi1	Combi2
C=25					
Max1	**3.18**	19.74	**4.82**	0	18.37
Max2	-	22.23	**2.09**	3.16	20.55
AddCol1	-	-	**25.76**	**19.75**	**2.38**
AddCol2	-	-	-	4.80	24.14
Combi1	-	-	-	-	18.38
C=40					
Max1	**4.21**	0.70	4.25	0.11	**5.73**
Max2	-	4.90	8.39	4.29	**1.45**
AddCol1	-	-	4.10	**0.54**	**6.19**
AddCol2	-	-	-	**4.55**	**10.12**
Combi1	-	-	-	-	**5.63**
C=55					
Max1	**5.61**	**0.94**	5.19	**0.94**	**6.37**
Max2	-	4.75	10.96	4.75	**0.71**
AddCol1	-	-	6.32	NoD	**5.46**
AddCol2	-	-	-	**6.32**	**11.78**
Combi1	-	-	-	-	**5.46**
C=75					
Max1	**5.70**	**0.95**	4.81	**0.95**	**6.50**
Max2	-	4.82	10.60	4.82	**0.75**
AddCol1	-	-	5.88	NoD	**5.58**
AddCol2	-	-	-	**5.88**	**11.44**
Combi1	-	-	-	-	**5.58**

Table F.4: Rel. Diff. in % for Hybrid Methods for Test Set 4.

	Max2	AddCol1	AddCol2	Combi1	Combi2
C=25					
Max1	**4.86**	9.81	**4.37**	**1.66**	8.31
Max2	-	14.69	0.53	3.25	12.55
AddCol1	-	-	**14.21**	**11.75**	**2.68**
AddCol2	-	-	-	2.75	12.07
Combi1	-	-	-	-	10.24
C=40					
Max1	**4.46**	8.25	**3.49**	0	8.58
Max2	-	12.93	1.19	4.74	13.06
AddCol1	-	-	**11.51**	**8.15**	0
AddCol2	-	-	-	3.80	11.53
Combi1	-	-	-	-	8.48
C=55					
Max1	1.70	2.97	0	0	0
Max2	-	1.77	**1.10**	**1.66**	**1.94**
AddCol1	-	-	**2.75**	**2.93**	**2.95**
AddCol2	-	-	-	0	0
Combi1	-	-	-	-	0
C=75					
Max1	2.35	**1.42**	1.68	**1.44**	**2.18**
Max2	-	**3.81**	**0.75**	**3.83**	**4.48**
AddCol1	-	-	3.08	**0.02**	**0.76**
AddCol2	-	-	-	**3.10**	**3.75**
Combi1	-	-	-	-	**0.74**

Table F.5: Rel. Diff. in % for Hybrid Methods for Test Set 5.

	$C = 25$	$C = 40$	$C = 55$	$C = 75$
EMSRb 2	0.20 s	0.24 s	0.35 s	0.33 s
You 2	1.47 s	2.29 s	3.19 s	4.36 s
B&P 2	0.09 s	0.12 s	0.15 s	0.19 s
B&P LP 2	15.32 s	24.44 s	33.77 s	46.68 s
C,G&J 2	3.89 s	5.67 s	7.62 s	10.20 s
C&H 2	0.90 s	1.35 s	1.80 s	2.47 s
EMSRb 3	0.22 s	0.27 s	0.20 s	0.25 s
You 3	1.30 s	2.05 s	2.79 s	3.80 s
B&P 3	0.09 s	0.11 s	0.13 s	0.17 s
B&P LP 3	14.06 s	22.67 s	31.44 s	42.71 s
C,G&J 3	3.58 s	5.33 s	7.04 s	9.29 s
C&H 3	0.81 s	1.24 s	1.65 s	2.22 s
EMSRb 5	0.24 s	0.30 s	0.30 s	0.30 s
You 5	1.47 s	2.38 s	3.29 s	4.47 s
B&P 5	0.09 s	0.11 s	0.14 s	0.18 s
B&P LP 5	14.65 s	24.24 s	33.70 s	45.91 s
C,G&J 5	3.75 s	5.64 s	7.55 s	9.99 s
C&H 5	0.90 s	1.39 s	1.89 s	2.54 s

Table F.6: Running Times for the General Methods, Test Set 2, 3 and 5.

	You	**B&P**	**B&P LP**	**C,G&J**	**C&H**
C=25					
EMSRb	**13.45**	**7.62**	**6.69**	**12.24**	**13.62**
You	-	7.17	8.06	2.57	0
B&P	-	-	1.00	**4.71**	**7.17**
B&P LP	-	-	-	**5.54**	**8.06**
C,G&J	-	-	-	-	**2.59**
C=40					
EMSRb	**5.20**	0	0	**4.27**	**3.65**
You	-	4.18	3.31	1.01	2.53
B&P	-	-	**0.90**	**3.30**	**2.92**
B&P LP	-	-	-	**2.44**	**2.15**
C,G&J	-	-	-	-	0
C=55					
EMSRb	**10.99**	**8.67**	**8.71**	**10.60**	**7.10**
You	-	2.28	2.25	0	3.75
B&P	-	-	**0.07**	**1.94**	1.58
B&P LP	-	-	-	**1.90**	1.66
C,G&J	-	-	-	-	3.42
C=75					
EMSRb	**13.57**	**11.93**	**11.95**	**12.98**	**7.38**
You	-	1.59	1.58	0.59	5.89
B&P	-	-	**0.01**	**1.06**	4.33
B&P LP	-	-	-	**1.05**	4.35
C,G&J	-	-	-	-	5.35

Table F.7: Rel. Diff. in % for Test Set 2 for the General Problem.

	You	B&P	B&P LP	C,G&J	C&H
C=25					
EMSRb	**4.59**	8.34	6.60	**4.20**	**5.17**
You	-	12.20	10.40	0	**0.53**
B&P	-	-	**1.76**	**10.89**	**12.78**
B&P LP	-	-	-	**9.13**	**10.99**
C,G&J	-	-	-	-	0
C=40					
EMSRb	**3.33**	5.62	4.12	**3.82**	**3.46**
You	-	8.25	6.62	**0.81**	**0.89**
B&P	-	-	**1.56**	**8.88**	**8.92**
B&P LP	-	-	-	**7.24**	**7.36**
C,G&J	-	-	-	-	**0.09**
C=55					
EMSRb	**9.46**	**3.17**	**3.51**	**9.40**	**5.50**
You	-	7.19	6.78	0.07	3.76
B&P	-	-	**0.38**	**7.11**	**3.62**
B&P LP	-	-	-	**6.70**	**3.23**
C,G&J	-	-	-	-	3.69
C=75					
EMSRb	**9.73**	**3.98**	**3.98**	**9.73**	**5.23**
You	-	6.57	6.57	NoD	4.28
B&P	-	-	0	**6.57**	**2.53**
B&P LP	-	-	-	**6.57**	**2.53**
C,G&J	-	-	-	-	4.28

Table F.8: Rel. Diff. in % for Test Set 3 for the General Problem.

	You	B&P	B&P LP	C,G&J	C&H
C=25					
EMSRb	**32.49**	**28.56**	**28.94**	**34.59**	**29.84**
You	-	4.98	4.16	**0.94**	1.76
B&P	-	-	0	**5.53**	**3.59**
B&P LP	-	-	-	**4.74**	**2.75**
C,G&J	-	-	-	-	2.83
C=40					
EMSRb	0.19	5.13	3.72	0	**1.11**
You	-	5.21	3.72	**0.00**	**0.99**
B&P	-	-	**1.37**	**4.66**	**6.00**
B&P LP	-	-	-	**3.29**	**4.58**
C,G&J	-	-	-	-	**1.33**
C=55					
EMSRb	**6.01**	**2.82**	**3.57**	**5.99**	**3.49**
You	-	3.40	2.61	**0.07**	2.44
B&P	-	-	**0.80**	**3.42**	**1.28**
B&P LP	-	-	-	**2.64**	**0.55**
C,G&J	-	-	-	-	2.44
C=75					
EMSRb	**8.70**	**7.18**	**7.20**	**8.50**	**3.74**
You	-	1.76	1.74	0.20	4.83
B&P	-	-	0	**1.58**	3.34
B&P LP	-	-	-	**1.56**	3.36
C,G&J	-	-	-	-	4.64

Table F.9: Rel. Diff. in % for Test Set 5 for the General Problem.

	L&H	EMSRb
C=25		
You	10.20	11.84
C,G&J	7.84	9.59
C=40		
You	5.17	6.22
C,G&J	4.65	5.72
C=55		
You	3.16	3.38
C,G&J	2.62	2.84
C=75		
You	1.70	1.71
C,G&J	1.06	1.07

Table F.10: Rel. Diff. in % when Comparing Non-TU and TU Methods, Test Set 2.

	L&H	EMSRb
C=25		
You	15.53	16.84
C,G&J	14.86	16.20
C=40		
You	9.40	9.89
C,G&J	10.38	10.92
C=55		
You	7.24	7.43
C,G&J	7.21	7.39
C=75		
You	6.92	6.92
C,G&J	6.92	6.92

Table F.11: Rel. Diff. in % when Comparing Non-TU and TU Methods, Test Set 3.

	L&H	EMSRb
C=25		
You	12.42	14.98
C,G&J	13.48	16.30
C=40		
You	6.98	8.03
C,G&J	6.69	7.81
C=55		
You	4.41	4.67
C,G&J	4.43	4.69
C=75		
You	1.98	1.99
C,G&J	1.80	1.81

Table F.12: Rel. Diff. in % when Comparing Non-TU and TU Methods, Test Set 5.

Appendix G

Results for the Simplified SIC Problem with Trade-Up

For all tables in this chapter, which show the relative difference in percentage for two methods the following holds. If a number different from zero appears in the tables, it cannot be rejected that the difference is greater than zero, i.e., it cannot be rejected that one method is significantly better than another with $\alpha = 5\%$. A bold number indicates that the method listed in the row in the top of the table is significantly better than the method listed in the column. If the number is not bold the method listed in the column yields a significantly higher revenue than the method in th top row. The symbol *NoD* means that no difference between the methods was observed for all 1000 runs.

G.1 Comparison of the Hybrid Methods

In Tables G.1 page 172, G.2 page 173, G.3 page 174, G.4 page 175 and G.5 page 176 the relative differences in % when comparing the seven different hybrid methods for the simplified SIC problem with trade-up are seen.

G.2 Results

In Table G.6 page 177 the average running times from 1000 test runs for six different methods for the simplified SIC problem with trade-up are shown. The times are shown for test sets 2, 3 and 5.

In Tables G.7 page 177, G.8 page 178 and G.9 page 178 the relaive differences in % from running the different methods 1000 times on test sets 2, 3 and 5, respectively, are seen.

	Max1	**Max2**	**Combi**	**AddColHM**	**Max1HM**	**Max2HM**	**CombiHM**
C=25							
AddCol	44.26	**14.06**	28.64	**6.19**	44.24	**14.15**	29.55
Max1	-	**64.19**	**12.48**	**53.14**	0	**64.36**	**11.87**
Max2	-	-	46.32	8.63	64.18	0	47.33
Combi	-	-	-	**35.76**	12.48	**46.52**	0.82
AddColHM	-	-	-	-	53.13	**8.72**	36.71
Max1HM	-	-	-	-	-	**64.35**	**11.87**
Max2HM	-	-	-	-	-	-	47.53
C=40							
AddCol	**2.39**	**6.66**	12.23	**2.14**	**3.27**	**6.35**	18.13
Max1	-	**4.92**	15.16	0	**0.91**	**4.60**	21.26
Max2	-	-	20.35	4.63	3.81	0.45	26.60
Combi	-	-	-	**14.39**	**16.31**	**20.05**	6.07
AddColHM	-	-	-	-	**1.49**	**4.29**	20.50
Max1HM	-	-	-	-	-	**3.47**	22.47
Max2HM	-	-	-	-	-	-	26.30
C=55							
AddCol	5.30	6.47	0	**0.51**	6.16	7.87	2.97
Max1	-	1.15	**5.25**	**5.68**	0.79	2.45	**3.05**
Max2	-	-	**6.42**	**6.81**	0	1.30	**4.21**
Combi	-	-	-	**0.66**	6.12	7.82	2.89
AddColHM	-	-	-	-	6.54	8.22	3.41
Max1HM	-	-	-	-	-	1.66	**3.92**
Max2HM	-	-	-	-	-	-	**5.59**
C=75							
AddCol	5.73	5.71	NoD	0	6.00	6.25	0
Max1	-	0	**5.73**	**5.71**	0.25	0.52	**5.71**
Max2	-	-	**5.71**	**5.68**	0.54	0.51	**5.68**
Combi	-	-	-	0	6.00	6.25	0
AddColHM	-	-	-	-	5.98	6.23	0
Max1HM	-	-	-	-	-	0	**5.98**
Max2HM	-	-	-	-	-	-	**6.23**

Table G.1: Rel. Diff. in % for Simplified Hybrid Methods for Test Set 1.

G.3 Comparison of Non-Trade-Up with Trade-Up Methods

In Tables G.10, G.11 and G.12 page 179 the relative differences in % from comparing the two best solution methods for the SIC problem without trade-up with the two best solution methods for the SIC problem with trade-up 1000 times are seen for test sets 2, 3 and 5, respectively.

	Max1	**Max2**	**Combi**	**AddColHM**	**Max1HM**	**Max2HM**	**CombiHM**
C=25							
AddCol	6.77	**13.84**	6.53	**2.35**	4.61	**15.07**	8.27
Max1	-	**20.28**	0	**8.73**	**2.47**	**21.71**	3.66
Max2	-	-	20.79	12.13	17.45	**1.10**	22.75
Combi	-	-	-	**8.23**	**4.27**	**22.18**	1.98
AddColHM	-	-	-	-	6.35	**13.25**	9.85
Max1HM	-	-	-	-	-	**18.76**	6.01
Max2HM	-	-	-	-	-	-	24.18
C=40							
AddCol	5.19	3.02	0.69	**0.32**	5.67	3.81	1.25
Max1	-	**2.61**	**4.58**	**5.46**	0.50	**1.89**	**4.21**
Max2	-	-	**2.49**	**3.27**	3.06	0.77	**2.14**
Combi	-	-	-	**0.98**	5.07	3.28	0.57
AddColHM	-	-	-	-	5.90	4.05	1.43
Max1HM	-	-	-	-	-	**2.32**	**4.67**
Max2HM	-	-	-	-	-	-	**2.93**
C=55							
AddCol	3.96	2.56	0	0	4.38	3.19	0.05
Max1	-	**1.52**	**3.96**	**3.94**	0.40	**0.94**	**3.92**
Max2	-	-	**2.56**	**2.53**	1.90	0.61	**2.51**
Combi	-	-	-	0	4.37	3.19	0
AddColHM	-	-	-	-	4.35	3.16	0
Max1HM	-	-	-	-	-	**1.32**	**4.33**
Max2HM	-	-	-	-	-	-	**3.15**
C=75							
AddCol	3.01	0.82	NoD	NoD	3.01	0.82	NoD
Max1	-	**2.24**	**3.01**	**3.01**	NoD	**2.24**	**3.01**
Max2	-	-	**0.82**	**0.82**	2.24	NoD	**0.82**
Combi	-	-	-	NoD	3.01	0.82	NoD
AddColHM	-	-	-	-	3.01	0.82	NoD
Max1HM	-	-	-	-	-	**2.24**	**3.01**
Max2HM	-	-	-	-	-	-	**0.82**

Table G.2: Rel. Diff. in % for Simplified Hybrid Methods for Test Set 2.

	Max1	**Max2**	**Combi**	**AddColHM**	**Max1HM**	**Max2HM**	**CombiHM**
C=25							
AddCol	9.80	**10.04**	5.27	**5.10**	6.90	**9.64**	7.69
Max1	-	**19.63**	0	**14.61**	**2.84**	**19.30**	2.30
Max2	-	-	15.28	5.18	15.70	0	18.03
Combi	-	-	-	**9.38**	**2.33**	**14.82**	2.62
AddColHM	-	-	-	-	11.35	**4.64**	11.95
Max1HM	-	-	-	-	-	**15.28**	4.85
Max2HM	-	-	-	-	-	-	17.61
C=40							
AddCol	2.82	3.39	0	**0.41**	3.71	4.28	0.89
Max1	-	0	**2.69**	**3.05**	0.85	1.49	**2.13**
Max2	-	-	**3.24**	**3.61**	0.58	0.84	**2.72**
Combi	-	-	-	**0.47**	3.57	4.12	0.79
AddColHM	-	-	-	-	3.93	4.50	1.16
Max1HM	-	-	-	-	-	0.66	**3.02**
Max2HM	-	-	-	-	-	-	**3.61**
C=55							
AddCol	2.31	2.87	**0.00**	0	2.31	2.87	0
Max1	-	0.62	**2.31**	**2.31**	NoD	0.62	**2.30**
Max2	-	-	**2.87**	**2.86**	**0.62**	NoD	**2.86**
Combi	-	-	-	0	2.31	2.87	0
AddColHM	-	-	-	-	2.31	2.86	0.00
Max1HM	-	-	-	-	-	0.62	**2.30**
Max2HM	-	-	-	-	-	-	**2.86**
C=75							
AddCol	2.23	2.75	NoD	NoD	2.23	2.75	NoD
Max1	-	0.56	**2.23**	**2.23**	NoD	0.56	**2.23**
Max2	-	-	**2.75**	**2.75**	**0.56**	NoD	**2.75**
Combi	-	-	-	NoD	2.23	2.75	NoD
AddColHM	-	-	-	-	2.23	2.75	NoD
Max1HM	-	-	-	-	-	0.56	**2.23**
Max2HM	-	-	-	-	-	-	**2.75**

Table G.3: Rel. Diff. in % for Simplified Hybrid Methods for Test Set 3.

	Max1	**Max2**	**Combi**	**AddColHM**	**Max1HM**	**Max2HM**	**CombiHM**
C=25							
AddCol	9.79	5.76	3.88	**0.42**	11.63	7.74	3.88
Max1	-	**4.07**	**6.81**	**10.01**	1.56	**2.31**	**6.81**
Max2	-	-	0	**5.95**	5.67	1.77	0
Combi	-	-	-	**4.50**	8.65	0	NoD
AddColHM	-	-	-	-	11.83	7.92	4.50
Max1HM	-	-	-	-	-	**3.85**	**8.65**
Max2HM	-	-	-	-	-	-	0
C=40							
AddCol	4.67	2.03	NoD	0	5.61	2.91	NoD
Max1	-	**2.64**	**4.67**	**4.68**	0.87	**1.82**	**4.67**
Max2	-	-	**2.03**	**2.04**	3.53	0.83	**2.03**
Combi	-	-	-	0	5.61	2.91	NoD
AddColHM	-	-	-	-	5.61	2.91	0
Max1HM	-	-	-	-	-	**2.69**	**5.61**
Max2HM	-	-	-	-	-	-	**2.91**
C=55							
AddCol	5.29	2.17	NoD	NoD	5.29	2.17	NoD
Max1	-	**3.16**	**5.29**	**5.29**	NoD	**3.16**	**5.29**
Max2	-	-	**2.17**	**2.17**	3.16	NoD	**2.17**
Combi	-	-	-	NoD	5.29	2.17	NoD
AddColHM	-	-	-	-	5.29	2.17	NoD
Max1HM	-	-	-	-	-	**3.16**	**5.29**
Max2HM	-	-	-	-	-	-	**2.17**
C=75							
AddCol	4.89	2.24	NoD	NoD	4.89	2.24	NoD
Max1	-	**2.65**	**4.89**	**4.89**	NoD	**2.65**	**4.89**
Max2	-	-	**2.24**	**2.24**	2.65	NoD	**2.24**
Combi	-	-	-	NoD	4.89	2.24	NoD
AddColHM	-	-	-	-	4.89	2.24	NoD
Max1HM	-	-	-	-	-	**2.65**	**4.89**
Max2HM	-	-	-	-	-	-	**2.24**

Table G.4: Rel. Diff. in % for Simplified Hybrid Methods for Test Set 4.

	Max1	**Max2**	**Combi**	**AddColHM**	**Max1HM**	**Max2HM**	**CombiHM**
C=25							
AddCol	16.04	**6.09**	12.77	**2.07**	16.09	**5.95**	18.11
Max1	-	**23.02**	0	**18.40**	0	**23.01**	5.21
Max2	-	-	19.89	4.74	23.11	0	25.52
Combi	-	-	-	**14.47**	0	**19.88**	5.35
AddColHM	-	-	-	-	18.43	**4.59**	19.83
Max1HM	-	-	-	-	-	**23.06**	5.30
Max2HM	-	-	-	-	-	-	25.58
C=40							
AddCol	2.61	4.50	0	0	3.05	5.71	1.60
Max1	-	1.97	**2.59**	**2.79**	0	3.14	**1.38**
Max2	-	-	**4.47**	**4.66**	**1.57**	1.14	**3.25**
Combi	-	-	-	0	3.03	5.67	1.59
AddColHM	-	-	-	-	3.17	5.85	1.68
Max1HM	-	-	-	-	-	2.72	**1.79**
Max2HM	-	-	-	-	-	-	**4.44**
C=55							
AddCol	1.79	2.95	0	0	2.02	3.66	0
Max1	-	1.21	**1.80**	**1.79**	0.22	1.90	**1.78**
Max2	-	-	**2.96**	**2.96**	**0.99**	0.67	**2.94**
Combi	-	-	-	0	2.02	3.66	0
AddColHM	-	-	-	-	2.02	3.66	0
Max1HM	-	-	-	-	-	1.68	**2.01**
Max2HM	-	-	-	-	-	-	**3.65**
C=75							
AddCol	1.52	1.71	NoD	NoD	1.52	1.71	NoD
Max1	-	0	**1.52**	**1.52**	NoD	0	**1.52**
Max2	-	-	**1.71**	**1.71**	0	NoD	**1.71**
Combi	-	-	-	NoD	1.52	1.71	NoD
AddColHM	-	-	-	-	1.52	1.71	NoD
Max1HM	-	-	-	-	-	0	**1.52**
Max2HM	-	-	-	-	-	-	**1.71**

Table G.5: Rel. Diff. in % for Simplified Hybrid Methods for Test Set 5.

	$C = 25$	$C = 40$	$C = 55$	$C = 75$
You 2	0.48 s	0.70 s	0.92 s	1.23 s
HM 2	0.62 s	0.80 s	1.00 s	1.33 s
B&P 2	0.11 s	0.15 s	0.18 s	0.24 s
C,G&J 2	0.21 s	0.22 s	0.22 s	0.23 s
Max2 2	0.36 s	0.44 s	0.54 s	0.67 s
AddColHM 2	0.34 s	0.44 s	0.54 s	0.69 s
You 3	0.43 s	0.63 s	0.84 s	1.09 s
HM 3	0.55 s	0.73 s	0.96 s	1.22 s
B&P 3	0.11 s	0.14 s	0.18 s	0.22 s
C,G&J 3	0.22 s	0.21 s	0.22 s	0.22 s
Max2 3	0.31 s	0.40 s	0.50 s	0.62 s
AddColHM 3	0.30 s	0.40 s	0.50 s	0.64 s
You 5	0.48 s	0.71 s	0.95 s	1.28 s
HM 5	0.58 s	0.78 s	1.02 s	1.33 s
B&P 5	0.11 s	0.15 s	0.19 s	0.25 s
C,G&J 5	0.22 s	0.24 s	0.23 s	0.25 s
Max2 5	0.35 s	0.46 s	0.59 s	0.74 s
AddColHM 5	0.32 s	0.43 s	0.56 s	0.73 s

Table G.6: Running Times for the Simplified Methods, Test sets 2, 3 and 5.

	HM	**B&P**	**C,G&J**	**Max2**	**AddColHM**
C=25					
You	0	10.54	0	4.20	16.27
HM	-	10.33	0	4.15	16.21
B&P	-	-	**11.32**	**6.75**	7.71
C,G&J	-	-	-	5.67	16.89
Max2	-	-	-	-	11.88
C=40					
You	0	15.13	**6.68**	4.32	0
HM	-	14.71	**7.06**	3.92	0
B&P	-	-	**21.13**	**10.78**	**14.91**
C,G&J	-	-	-	11.05	7.21
Max2	-	-	-	-	**4.12**
C=55					
You	0	16.90	12.33	2.58	0
HM	-	16.88	12.31	2.56	0
B&P	-	-	**4.59**	**14.34**	**16.86**
C,G&J	-	-	-	**9.96**	**12.30**
Max2	-	-	-	-	**2.55**
C=75					
You	NoD	19.45	19.41	0.94	NoD
HM	-	19.45	19.41	0.94	NoD
B&P	-	-	**0.04**	**18.52**	**19.45**
C,G&J	-	-	-	**18.48**	**19.41**
Max2	-	-	-	-	**0.94**

Table G.7: Rel. Diff. in % for Test Set 2 for the Simplified Problem.

	HM	B&P	C,G&J	Max2	AddColHM
C=25					
You	0	26.06	5.37	3.28	7.53
HM	-	25.95	5.42	2.87	7.64
B&P	-	-	**20.09**	**22.99**	**18.66**
C,G&J	-	-	-	**2.93**	7.40
Max2	-	-	-	-	5.33
C=40					
You	0	20.91	2.75	3.56	0.48
HM	-	20.87	2.71	3.46	0.43
B&P	-	-	**18.40**	**17.46**	**20.41**
C,G&J	-	-	-	0	0
Max2	-	-	-	-	**3.14**
C=55					
You	0	20.46	18.25	2.81	0
HM	-	20.46	18.26	2.81	0
B&P	-	-	**2.01**	**17.73**	**20.45**
C,G&J	-	-	-	**15.64**	**18.25**
Max2	-	-	-	-	**2.80**
C=75					
You	NoD	20.70	20.70	3.09	NoD
HM	-	20.70	20.70	3.09	NoD
B&P	-	-	0	**17.74**	**20.70**
C,G&J	-	-	-	**17.74**	**20.70**
Max2	-	-	-	-	**3.09**

Table G.8: Rel. Diff. in % for Test Set 3 for the Simplified Problem.

	HM	B&P	C,G&J	Max2	AddColHM
C=25					
You	**0.11**	25.77	2.65	7.15	11.46
HM	-	25.73	2.78	7.14	11.73
B&P	-	-	**23.14**	**18.22**	**15.55**
C,G&J	-	-	-	7.30	11.03
Max2	-	-	-	-	4.89
C=40					
You	**0.16**	17.69	**5.62**	5.54	1.11
HM	-	17.76	**5.60**	5.57	1.22
B&P	-	-	**21.05**	**12.06**	**16.81**
C,G&J	-	-	-	11.42	6.69
Max2	-	-	-	-	**4.72**
C=55					
You	0	14.70	8.56	2.95	0
HM	-	14.67	8.54	2.92	0
B&P	-	-	**6.26**	**11.78**	**14.67**
C,G&J	-	-	-	**5.92**	**8.53**
Max2	-	-	-	-	**2.91**
C=75					
You	NoD	17.36	17.30	1.74	NoD
HM	-	17.36	17.30	1.74	NoD
B&P	-	-	**0.06**	**15.58**	**17.36**
C,G&J	-	-	-	**15.52**	**17.30**
Max2	-	-	-	-	**1.74**

Table G.9: Rel. Diff. in % for Test Set 5 for the Simplified Problem.

	L&H	**EMSRb**
C=25		
HM	13.26	9.69
AddColHM	**3.49**	**6.42**
C=40		
HM	16.20	13.81
AddColHM	16.06	13.65
C=55		
HM	15.51	14.22
AddColHM	15.50	14.20
C=75		
HM	19.75	19.70
AddColHM	19.75	19.70

Table G.10: Rel. Diff. in % when Comp. Non-TU with Simple TU Methods, Test Set 2.

	L&H	**EMSRb**
C=25		
HM	34.87	23.39
AddColHM	28.42	17.30
C=40		
HM	26.75	23.57
AddColHM	26.57	23.38
C=55		
HM	18.80	18.34
AddColHM	18.80	18.34
C=75		
HM	20.83	20.83
AddColHM	20.83	20.83

Table G.11: Rel. Diff. in % when Comp. Non-TU with Simple TU Methods, Test Set 3.

	L&H	**EMSRb**
C=25		
HM	30.56	22.30
AddColHM	20.56	12.68
C=40		
HM	20.94	17.50
AddColHM	19.81	16.41
C=55		
HM	13.88	11.99
AddColHM	13.86	11.96
C=75		
HM	17.73	17.57
AddColHM	17.73	17.57

Table G.12: Rel. Diff. in % when Comp. Non-TU with Simple TU Methods, Test Set 5.

Appendix H

Programs

All the programs described in this report are enclosed on the CD.

Bibliography

[1] Staffan Algers, Sven-Eric Andersson, and Joakim Kohler. Impact of deviation and recapture upon class allocations. *Status report, 33rd Agifors Symposium*, pages 237–255, 1993.

[2] S.-E. Andersson. Passenger choice analysis for seat capacity control: A pilot project in scandinavian airlines. *Int. Trans. Opl. Res.*, 5:471–486, 1998.

[3] Peter P. Belobaba. Airline yield management: An overview of seat inventory control. *Transportation Science*, 21:63–73, 1987.

[4] Peter P. Belobaba. Application of a probabilistic decision model to airline seat inventory control. *Operations Research*, 37:183–197, 1989.

[5] Peter P. Belobaba. Optimal vs. heuristic methods for nested seat allocation. *Proceedings of AGIFORS Reservations and Yield Management Study Group*, pages 28–53, 1992.

[6] Peter P. Belobaba and Lawrence R. Weatherford. Comparing decision rules that incorporate customer diversion in perishable asset revenue management situations. *Decision Sciences*, 27:343–363, 1996.

[7] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control, volume one*. Athena Scientific, Belmont, Massachusetts, USA, 1995.

[8] Dimitris Bertsimas and Ioana Popescu. Revenue management in a dynamic network environment. *Transportation Science*, 37:257–277, 2003.

[9] S.E. Bodily and L.R. Weatherford. Perishable-asset revenue management: Generic and multiple-price yield management with diversion. *Omega, Int.J. Management Science*, 23:173–185, 1994.

[10] T. Botimer. Efficiency considerations in airline pricing and yield management. *Transpn. Res.-A.*, 30:307–317, 1996.

[11] Victoria C.P. Chen, Dirk Gunther, and Ellis L. Johnson. A markov decision problem based approach to the airline ym problem. *Working Paper, http://www.isye.gatech.edu/research/files/lec9812.pdf*, 1998.

[12] Knut Conradsen. *En Introduktion til Statistik*. IMM, DTU, Lyngby, DK, 1999.

[13] William L. Cooper. Asymptotic behavior of a class of an allocation policy for revenue management. *Operations Research*, 50:720–727, 2002.

[14] William L. Cooper and Tito Homem de Mello. A class of hybrid methods for revenue management. *Working Paper No. 03-015, http://www.iems.nwu.edu/grad/workingpapers/WP_03_015.pdf*, 2003.

[15] G. Gallego and G. van Ryzin. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, 40:999–1020, 1994.

[16] Hans-Martin Gutmann. British airways. 2004.

[17] Richard A. Johnson. *Miller & Freund's Probability and Statistics for Engineers*. Prentice Hall International, Inc., Upper Saddle River, USA, 2000.

[18] C. Lautenbacher and jr. S. Stidham. The underlying decision process in the single-leg airline yield management problem. *Transportation Science*, 33:136–146, 1999.

[19] Tak C. Lee and Marvin Hersh. A model for dynamic airline seat inventory control with multiple seat bookings. *Transportation Science*, 27:252–265, 1993.

[20] K. Littlewood. Forecasting and control of passenger bookings. *AGIFORS Symposium Proceedings*, 12:95–117, 1972.

[21] J. McGill and G. van Ryzin. Revenue management: Research overview and prospects. *Transportation Science*, 33:233–256.

[22] K. Pak and N. Piersma. Overview of or techniques for airline revenue management. *Statistica Neerlandica*, 56:479–495, 2002.

[23] Frank L. Severance. *System Modeling and Simulation*. John Wiley & Sons Ltd, West Sussex, England, 2001.

[24] Janakiram Subramanian, Shaler Stidham Jr., and Conrad J. Lautenbacher. Airline yield management with overbooking, cancellations and no-shows. *Transportation Science*, 33:147–167, 1999.

[25] Kalyan Talluri and Garrett van Ryzin. Revenue management under a general discrete choice model of customer behavior. *Accepted by Management Science*, pages 1–33, 2001.

[26] L.R. Weatherford and P.P. Belobaba. Revenue impacts of fare input and demand forecast accuracy in airline yield management. *Journal of the Operational Research Society*, 53:811–821, 2002.

[27] Laurence A. Wolsey. *Integer Programming.* John Wiley & Sons, Inc., New York, USA, 1998.

[28] Peng-Sheng You. Airline seat management with rejction-for-possible-upgrade decision. *Transportation Research Part B*, pages 507–524, 2001.

[29] W. Zhao and Y.-S. Zheng. A dynamic model for airline seat allocation with passenger diversion and no-shows. *Transportation Science*, 35:80–98, 2001.