UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE E TECNOLOGIE

Scuola di Dottorato in Scienze Matematiche
Dipartimento di Matematica
Dottorato in Matematica e Statistica per
le Scienze Computazionali - XXVI Ciclo

Tesi di Dottorato di Ricerca

Column generation models for optimal package tour composition

Settore scientifico/disciplinare MAT/09

PhD Thesis of:
Adriana Novaes

Tutor:
Prof. Giovanni Righini
Coordinator:
Prof. Giovanni Naldi

A.A. 2013/2014

Acknowledgments

First I want to thank my colleagues Andrea Taverna, Andrea Bettinelli and Emanuele Tresoldi from Optalb for their support and kindness and Prof. Alberto Ceselli for his guide and friendship.

I would like to thank Prof. Bassok, Prof. Vespucci, Prof. Gambarelli, Prof. Ntaimo and Prof. Caldentey for their encouragement and professional example. I am grateful to Maria Teresa Vespucci and Gregorio Tirado Domínguez for their willingness to revise my thesis and for their useful suggestions.

I will always be thankful to my supervisor Giovanni Righini for his inestimable support helping me to remove all rocks we found along the way.

I want to thank my parents for their inspiration, my daughter Letizia and my son Leonardo for joy they give me and above all my husband Marco, because without him I wouldn't have started and I wouldn't have finished this journey.

Contents

List of Figures

List of Tables

Preface

Our work aims to introduce a combinatorial optimization problem orbiting in Revenue Management, called Package Tour Composition (PTC) and to discuss its resolution with a mathematical programming method called column generation method. The classic Network Revenue Management problem considers a set of resources of finite capacity to be allocated to a set of products characterized by a given price and a given demand. The models of Network Revenue Management are applied by airline companies in order to decide how many seats to allocate on each flight leg (resource) to each fare (product) that is characterized by origin, destination and fare class. The model we propose aims to deal with a similar problem in which the demand is not expressed towards a set of products but towards a set of resources. This problem arises, for instance, in the composition of package tours where customer preferences towards events that compose a package tour are more relevant, and easier to be traced, than customer preferences for the whole package.

In the PTC problem customers buy products that are bundles of resources in combinations under various terms and conditions. However demand is linked to resources not to products. The resource composition of each product is a decision variable. As a consequence product price is not known but is the sum of reservation prices of each resource in the bundle. The resource set is partitioned into several subsets corresponding to different resource types. A parameter states how many resources of each type characterize each product type. We refer to resources as 'events' and to products as 'package tours' or simply 'packages'.

The resulting Package Tour Composition problem is a non-linear problem with integer variables that represent the number of tourists assigned to each package tour and binary variables that represent which events are assigned to each package tour. Each event is characterized by a reservation price, a demand and a capacity. Each package tour belongs to a package tour type that is characterized by its event type composition parameter. The number of tourists assigned to each event cannot exceed its actual capacity, which is defined as the minimuml value between the event capacity and the event demand. We also impose that the binary variables respect the composition constraint for every package tour according to its type. The objective function to be maximized is the total revenue, that is the number of packages to be sold times their price.

We propose a column generation model to solve the linear relaxation of the Package Tour Composition problem. The Column Generation technique splits the problem in two sub-problems: the pricing problem and the master problem. The pricing problem dynamically generates, for every package type, several columns containing an event combination according to the package type composition parameter. The master problem chooses which event combinations to use and in which quantity, imposing that event actual capacity is respected, in order to maximize revenue.

Chapter 1 concerns the motivation of our research. At first we analyze the previous literature on the theory of Revenue Management focusing our attention on the most important mathematical models that tackle two main Revenue Management problems: Single Resource Capacity Control and Network Capacity Control. We analyze the assumptions of these models to find improvement directions. After that, we present the state-of-the-art of mathematical models applied to tourist operators industry, in particular in the composition of tour itineraries. We propose a taxonomy to classify several possible Package Tour Composition problem formulations.

In Chapter 2 the Package Tour Composition Model is formally defined and we propose the application of Column Generation method and a Column generation heuristics method to determine an optimal solution to the linear relaxation problem and a rounded solution to the integer problem. Two formulations are compared: the integer master formulation and the binary master formulation. Thereafter we present the dataset description and we display the results of integer and binary master formulations.

In Chapter 3 we illustrate several extensions of the basic models. The extensions take into account market segmentation, inconvenience costs, tourist groups and stochastic demand. For each extension we present computational results obtained with the state-of-the-art mathematical programming solver CPLEX.

Finally Chapter 4 presents some conclusions and possible future research directions.

Chapter 1

Package Tour Problem Background

This chapter introduces the motivation of this thesis and presents the Package Tour Composition Problem (PTCP). We put the PTCP in perspective in the context of Revenue Management, bundling methods and innovation in tourism services. For this purpose we present an overview of the scientific literature concerning Revenue Management and tourism and we outline similarities and differences between the PTCP and some already studied problems.

1.1 The Package Tour Composition Problem

Motivation. The international exposition Expo2015 will bring to Milan 21 million visitors, 30% foreigners. There will be the need of 500000 sleeping accommodations within 90 minutes travel from the city. In addition 100 international tour operators confirmed their interest in offering package tours to the visitors [1].

These numbers motivate first of all the need of promoting territories around Milan that have been so far excluded from mass tourism. Italian tourism facilities such as restaurants and hotels are often managed by families that own and run their own businesses. Small size restaurants, accommodations and retailers are spread throughout the Lombardy region as well. Few of them have an Internet site and the personnel often does not speak a foreign language. There is no coordination between their activities and therefore the sector as a whole is not prepared to make the best of the opportunity offered by the flow of Expo2015 visitors.

This is an example - actually the main initial motivation for this thesis - of the need of coordinated planing on the side of the tourism offer. A centralized coordination could give tour operators better access to the many events and points of interest scattered around in the region, including those that would be almost invisible to the most part of potential visitors.

Our work proposes a mathematical programming model with the aim of determining an optimal set of packages to be sold to tourists, in order to evenly distribute the tourists flow to the many available sites, complying with their limited accommodation capacity. Packages can be defined in order to achieve a suitable mix of event types so that events of different types can promote one another: for instance visitors attracted by music concerts will be also directed to restaurants, while visitors attracted by food and wine will also be presented artistic sites.

On the side of demand the advantage is to avoid congestion in some events/sites (typically within the city of Milan, in the case of Expo2015) and under-exploiting of others (typically in the country around the city), thus improving tourists' experience. On the side of offer the advantage is to allow all events/sites to count on a suitable level of demand, to maximize the overall revenue achievable from the visitors' flow and to evenly distribute this revenue among all the operators.

Problem description. The mathematical model of the PTCP is described in detail in the next chapter. Here we give an informal description of the PTCP to allow for a comparison with the relevant literature. In particular we refer to the scientific literature on revenue management and tourism.

We consider two main sets: events and package tour types. Events are classified into event types. For example: churches, museums, castles, parks, shopping outlets, concerts, restaurants and accommodations.

Package tour types are characterized by a composition pattern. For instance a package tour 'Shopping' can be composed of one castle, one natural park, three restaurants, two nights in a hotel and three shopping outlets; a package tour 'Religious' can be composed of three churches, two museums, three restaurants and two nights in a hotel.

It is important to note that two package tours of the same type can be made of different events. For example two 'Shopping' package tours can lead the tourist to visit different castles, parks, outlets, restaurants and hotels.

Each event is characterized by a finite capacity, a price and an estimated level of demand. Each package tour is characterized by a price, a number of tourists and a subset of events complying with the package type composition pattern. The package tour price is the sum of the prices of the events included in it.

The aim of the optimization process is to define the right number of the right packages in order to make the best - in terms of revenue - of the available capacities.

1.2 Scientific literature on Revenue Management

1.2.1 Revenue Management

Definition. The early definition of Revenue Management (RM) is historically anchored to the airline applications and states that RM, also called Yield Management, aims at *maximizing passenger revenue by selling the right seats to the right customers at the right time* [2]. Pfeifer[3] defined yield management as the *process by which discount fares are allocated to scheduled flights for the purposes of balancing demand and increasing revenues.* Orkin defined yield management as *controlling the tradeoff between average rate and occupancy.* Cross [4] defined Revenue Management as *using price incentives and inventory controls to maximize the value of existing processes.*

Weatherford and Bodily[5] proposed a taxonomy and a research overview of Revenue Management problems. They introduced the term PARM - Perishable Asset Revenue Management. However since that publication the techniques and applications in Revenue Management undertook a substantial development and that terminology and taxonomy were abandoned. McGill and Van Ryzin [6] made a remarkable research overview of Revenue Management problems. They discussed the major areas of revenue management research (forecasting, overbooking, seat inventory control and pricing) focusing on the airline industry and defining the RM goal as *maximizing profits* *seeking booking policies that maximize revenue.*

After 2000 the RM literature focus was broaden to other areas outside the airline industry and the definition of RM was also enlarged. Shelby and Brunel [7] stated that *Revenue Management is concerned with making efficient use of a given fixed resource that becomes worthless after a given time. It uses controls such as booking or sales limits at various price levels.* According to Modarres and Sharifyazdi [8] *any revenue management problem contains the following common characteristics: capacity is perishable and limited (it cannot be enhanced easily in short term), demand is stochastic and there are different customer classes. The available perishable asset can be sold at different prices, through different booking classes (usually at different periods).*

More general definitions point out the application of RM outside the airline industry using products and resources in a more general manner. Zhang [9] definition argues RM entails controlling the availability/pricing of different products that use the same set of resources in order to maximize revenue. Bijvank et al 2011 states RM *involves the allocation of scarce resources to stochastic demand for products that consume one or more of these resources, with the aim of maximizing total expected revenue.*

Several papers give an overview of Revenue Management techniques. As mentioned Weatherford and Bodily [5] propose a taxonomy with 14 elements and they review the research literature related to each element. Mc Gill and van Ryzin [6] analyze the major areas of revenue management research (forecasting, overbooking, seat inventory control and pricing) focusing on the airline industry.

Pak and Piersman [10] present an overview of operational research techniques applied to the airline industry. Briten and Caldentey [11] present an overview of pricing models for RM. Chiang, Chen and Xu [12] provide a comprehensive review describing applications, major RM problems and issues for future research.

The major references on RM are the two textbooks *Revenue Management - Theory and Practice* by Talluri van Ryzin [13], that provides an extensive review of the RM literature with 585 references, and *Pricing and Revenue optimization* by Robert Philips [14].

To our purpose we highlight Talluri and van Ryzin [13] definition, that is: Revenue Management *is concerned with demand management decisions and the methodology and systems required to make them.* There are three kinds of decisions involved: structural decisions, price decisions and quantity decisions.

1. Structural decisions: e.g. which selling format to use (such as posted prices, negotiations or auctions); which segmentation or differentiation mechanisms to use (if any); which terms of trade to offer (including

volume discounts and cancelation or refund options); how to bundle products.

2. Price decisions: e.g. how to set posted prices, individual offer prices, and reserve prices (in auctions); how to price across product categories; how to price over time; how to markdown (discount) over the product lifetime.

3. Quantity decisions: e.g. whether to accept or reject an offer to buy; how to allocate output or capacity to different segments, products or channels; when to withhold a product from the market and sale at later points in time.

Revenue Management techniques and applications are usually divided into *pricing strategies* and *capacity control strategies*. The different application of these two types of strategies depends on the extent to which a firm is able to vary price and quantity, according to the changes in market conditions. For example price-based RM is widely used in the retail industry, because it is very easy to make price changes in this sector. On the other hand quantity-based RM is widely used by airlines because for marketing and administrative reasons most airlines advertise and price fare products and therefore it is difficult to make price changes

Structural decisions are normally strategic decisions that are taken relatively infrequently. Therefore they play a secondary role in the RM literature. Our research focuses on structural decisions, because it addresses the decision on how to define the right number of packages including the right subsets of events.

The basic problems in capacity-based RM can be distinguished into *single resource capacity* and *network capacity*. Single resource capacity problems concern the optimal allocation of a limited resource to different classes of demand. Two prototypical examples are: controlling the sale of different fare classes on a single flight leg of an airline and the sales of rooms in a hotel for a given date at different price classes. Network capacity problems concern the optimal allocation of capacity when customers buy bundle of resources. In these problems a lack of availability of any one resource in the bundle limits sales. Classical examples of network RM are the origin-destination itinerary fare class combination problem in the airline industry and the problem of assigning room capacity on consecutive days in the hotel industry.

We focus our attention on network RM models and especially on bundling models, which are analyzed in more depth in the next subsections.

Nesting setting and control rules Before presenting such assumptions it is crucial to clarify the concepts of nesting setting rules and control rules. The access each demand class has to each capacity class can be set according to three different nesting rules: non-nested capacity allocation, nested capacity allocation and *theft nesting* capacity allocation. In the non-nested approach each demand class has access just to the correspondent capacity class. The nested capacity allocation is the most applied allocation rule: higher classes demand has access to lower class capacity if its own class capacity is depleted. The last approach, theft nesting is rarely used: higher capacity has access to lower capacity also when the own class capacity is still available. The Figure 1.1 displays the control rules applied to protect capacity to high value customer: booking limits, protection level and bid prices. The representation of such control policies are displayed in the figure 1.1.

In a nested control policy the capacity available to different classes overlaps in a hierarchical manner - with higher-ranked classes having access to all the capacity reserved for lower. The first rule defines the nested booking limit b_j for each class j as the maximum number of units of capacity we are willing to sell up to class j. For example for high class 1 the nested booking limit is $b_1 = 30$, that is the entire capacity. For medium class 2 the nested booking limit is $b_2 = 18$. For low class 3 the nested booking limit is $b_3 = 8$, that is the class 3 capacity. We accept at most 30 bookings for classes 1,2 and 3, at most 18 for classes 2 and 3 combined and at most 8 for class 3.

The second rule sets protection level y_j specifies an amount of capacity to reserve (protect) for a particular class j or set of classes. For example a nested control policy we might set a protection level y_1 of 12 for class 1 (meaning 12 units of capacity are protected for sale to class 1), a protection leve y_2 of 22 for classes 1 and 2 combined, and a protection level y_3 of 30 for class 1,2 and 3 combined.

The third rule sets threshold prices that are defined in a bid price table (based on the remaining capacity or time). A request is accepted just if the revenue exceeds the threshold price $\pi(x)$. Bid prices can be used in the same nested-allocation policy as booking limits and protection levels. In Figure **??**fig:controls the bid price $\pi(x)$ is plotted as a function of the remaining capacity x. When there are 12 or fewer units remaining, the bid price is over \$75 but less than \$100, so that only class 1 demand is accepted. With 13 to 22 units remaining, the bid price is over \$50 but less than \$75, so that only class 1 and 2 are accepted. With more than 22 units of capacity available, the bid price drops bellow \$50, so that all three classes are accepted.

1.2.2 Network Revenue Management

In Network Revenue Management problem customers buy bundles of resources in combinations under various terms and conditions. When products are sold in bundles, the lack of availability of any one resource in the bundle limits sales. This creates interdependencies among resources and hence it becomes necessary to jointly manage the capacity controls on all resources. In the airline industry this is also called the passenger-mix problem or ODF

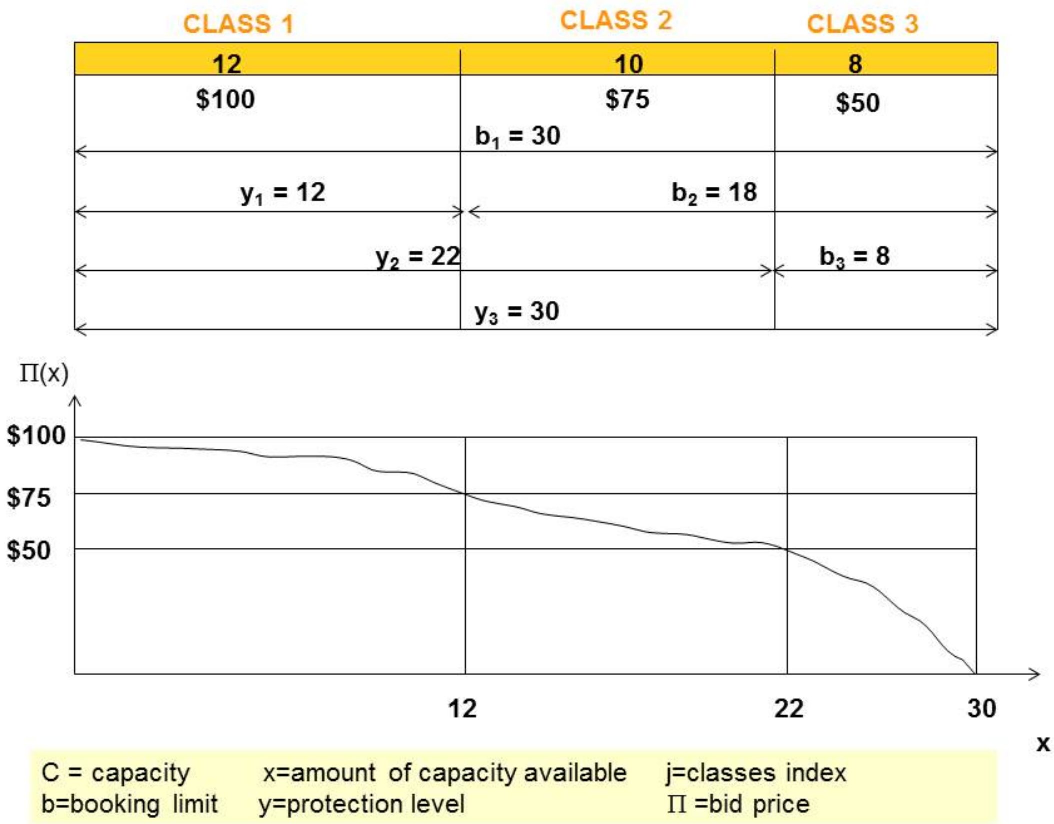

Figure 1.1: Nesting setting and control rules

(origin-destination-fare) control, where several origin-destination fare products share capacity in the airline network flights. In the hotel industry this is also called length-of-stay control problem, where customers with different lengths of stay compete for the same day-room pairs. In the Network Revenue Management models the price of the products are assumed to be given and they are not associated with the resources present in the bundle.

Network RM problems are usually modeled by a network made of m resources and a firm that sells n products. An incidence matrix $A = [a_{ij}]$ is given, where $a_{ij} = 1$ if resource i is used by product j and $a_{ij} = 0$ otherwise. Thus the j^{th} column of A, denoted by A_j is the *incidence vector* for product j.

The state of the network is defined by the vector $x = (x_1, ..., x_m)$ and indicates available resource capacities. Time is discrete: there are T periods, and index t represents the current time for the decision-maker (with the time index running forward, so $t = T$ is the time of service). Within each time period t we assume that at most one request for a product can arrive, that is the discretization of time is sufficiently fine so that the probability of more than one request is negligible.

The demand in period t is modeled as the realization of a single random vector $\mathrm{P}(t) = (P_1(t), ..., P_n(t))$. If $P_j(t) = p_j > 0$, this indicates that a request for product j has occurred that is associated with price p_j; if $P_j(t) = 0$, this indicates there is no request for product j at time t.

The sequence $\{\mathrm{P}(t),\ t \geq 1\}$ is assumed to be independent across time t with known joint distribution in each period t. Let the prices associated with the products be $p = (p_1, ..., p_n)$.

At a given current time t, given the current residual capacity x and the current request $\mathrm{P}(t)$, the quantity-based RM decision is as follows: Do we or do we not accept the current request?

Let the n-vector $\mathrm{u}(t)$ denote this decision, where $u_j(t) = 1$ if a request for product j in period t is accepted, and $u_j(t) = 0$ otherwise. The decision $u(t)$ is a function of the residual capacity vector x and the price p_j and hence $u(t) = u(t, x, p)$. The vector $u(t)$ is restricted to the set $\mathcal{U}(x) = \{u \in \{0,1\}^n : Au \leq x\}$.

Dynamic Programming formulation. To formulate a *dynamic programming algorithm* to determine optimal decisions $u^*(t, x, p)$, let $V_t(x)$ denote the maximum expected revenue to go, given the residual capacity x in period t. Then $V_t(x)$ must satisfy the Bellman equation

$$V_t^{DP}(x) = \mathbb{E}\Big[\max_{u \in \mathcal{U}(x)} \big\{ P(t)^\intercal u(t, x, p) + V_{t+1}(x - Au) \big\}\Big] \qquad (1.1)$$

with boundary conditions

$$V_{T+1}(x) = 0 \quad \forall x. \tag{1.2}$$

The set $V_t(x)$ is finite for all finite x and an optimal control u^* satisfies

$$u_j^*(t, x, p_j) = \begin{cases} 1 \ if \ p_j \leq V_{t+1}(x) - V_{t+1}(x - A_j) \ and \ A_j \leq x, \\ 0 \ otherwise. \end{cases}$$

An optimal control policy for accepting requests is of the form: accept a request for a product j (at price p_j) if and only if there is sufficient remaining capacity and

$$p_j \geq V_{t+1}(x) - V_{t+1}(x - A_j),$$

where p_j is the price of product j.

This reflects the rather intuitive notion that we accept a booking request for product j only if its price exceeds the *opportunity cost* of the reduction in resource capacities required to satisfy the request.

Deterministic Linear Programming formulation As explained in Talluri van Ryzin [13] in the *Deterministic Linear Programming* formulation the aggregate demand coming at time t for each product j is denoted by D_j: it is the overall demand over the periods $t, t+1, ..., T$. Its mean value is indicated by μ_j. Let $D = (D_1, ..., D_n)$ and $\mu = \mathbb{E}[D]$ denote the vectors of demand and mean demand respectively. The problem is displayed in figure 1.2. The deterministic linear programming (DLP) method uses the approximation

$$V_t^{DLP} = \max p^T y \tag{1.3}$$

$$s.t. \ Ay \leq x \tag{1.4}$$

$$0 \leq y \leq \mu. \tag{1.5}$$

The decision variables y $= (y_1, ..., y_n)$ represent the partitioned allocation of capacity for each of the n products. The approximation effectively treats demand as if it were deterministic and equal to its mean μ and makes an optimal partitioned allocation accordingly. Using Jensen's inequality[15] one can show that $V_t^{LP}(x)$ is in fact an upper bound on the optimal value function [16, 17, 18].

There are three kinds of nested control: partitioned, virtual and bid-price. They are further explained in the following sections. Occasionally, the optimal primal solution to (1.3) is used to construct a partitioned control directly . More often, the primal allocation are discarded and one uses only the optimal dual variables, denoted by π^{LP}, associated with the constraints $Ay \leq x$ as bid prices .

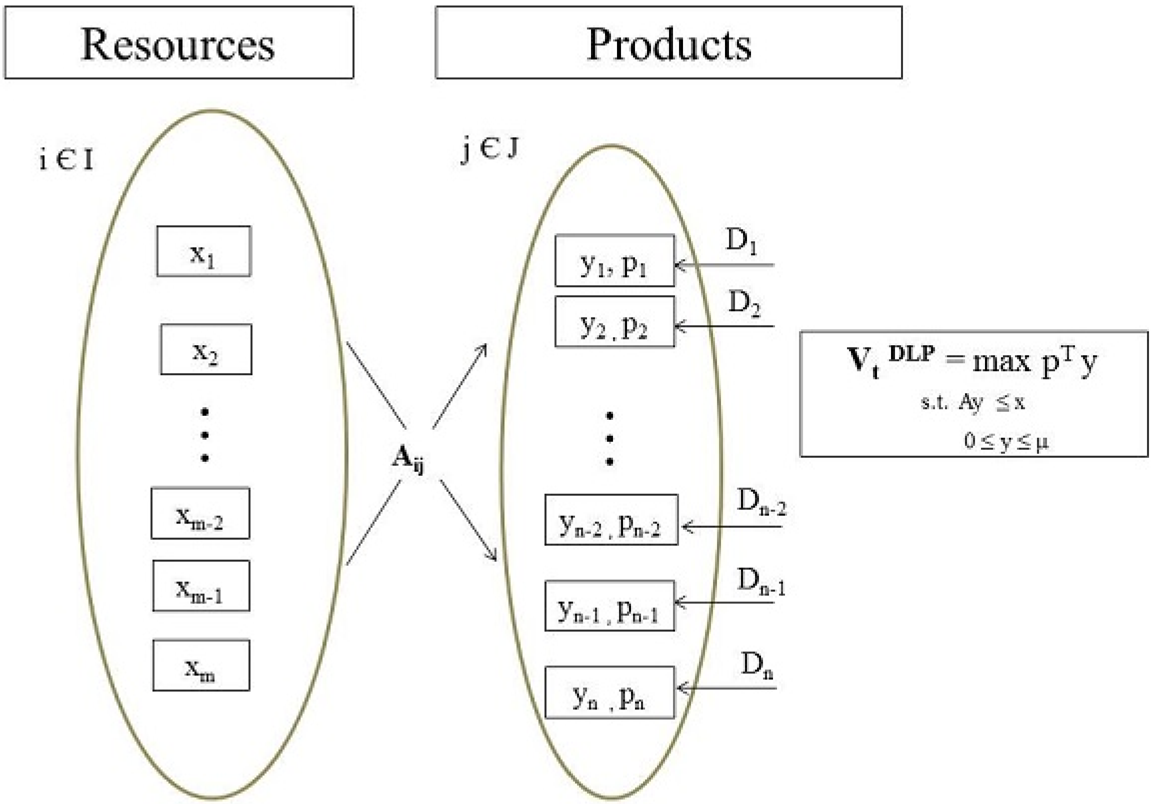

Figure 1.2: Network RM Problem: DLP formulation

Original contribution. In Network RM problems customers buy bundle of resources in combinations under various terms and conditions. This is also true in the PTCP we study. Nevertheless in our model the demand is linked to resources (events) not to products (packages). In addition the composition of the products, described by the incidence matrix A, is a known parameter in Network RM while it is a decision variable in our model. Consequently products (packages) of the same type may have different prices in the PTCP, because in general they include different resources (events).

For example in the airline industry customers are concerned with origins and destinations of their flights, while the flight legs are a secondary feature of the product they purchase. A traveler that flies from Milan to Rio de Janeiro is interested just in the origin and destination cities. It is not so important for him to use the flight legs Milan-Paris or Paris-Rio, or Milan-Madrid and Madrid-Rio. On the contrary in the PTCP customers place their demand on single events.

In Network RM resources are not categorized, as flight legs in the airline industry, instead in the PTC problem resources are classified in resources types. Finally in Network RM demand is decomposed into a set of discrete time periods before the expiration date, while in the PTCP consider the aggregated demand in the entire period.

We summarized the major differences between classical Network RM models and the PTCP model in Figure 1.3.

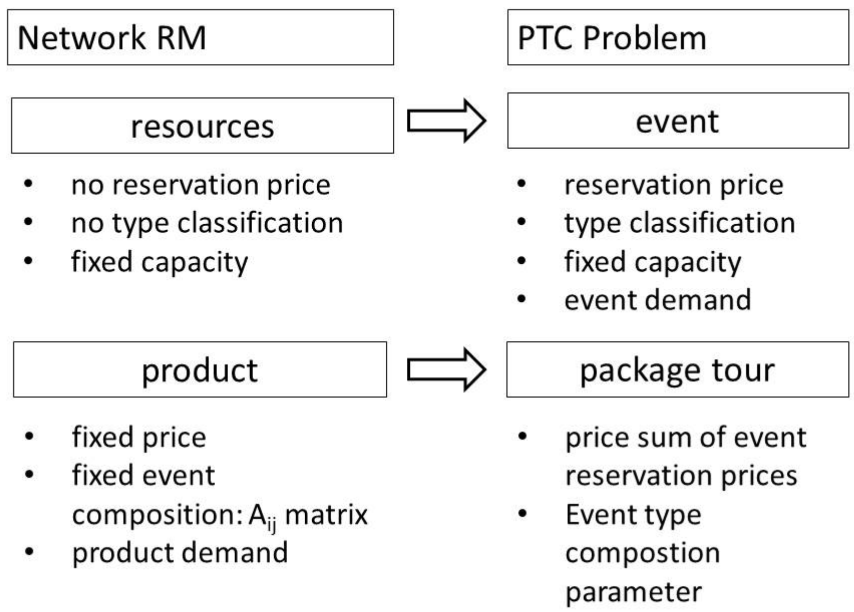

Figure 1.3: Comparison between the PTCP and Networtk RM.

1.2.3 Bundling

Bundling is the sale of two or more separate products. Specifically separate products *are products for which separate market exists*[19]. Following Ghosh, *Bundling is the practice of offering two or more products as a package* [20]. According to Cataldo, *Bundling has been widely investigated from consumer behavior, economic and marketing points of view* [21]. According to Kobayashi, *The majority of cases where bundling is observed, the reason why separate goods are sold as a package is easily explained by economies of scope in production or reduction in transactions and information costs, with an obvious benefit to the seller, the buyer or both* [22]. Stremersch and Tellis [23] identify two key dimensions that enable a comprehensive classification of bundling strategies: pure vs. mixed bundling and price vs. product bundling.

They also formulate rules for evaluating the legality of each of these strategies and they propose a framework of twelve propositions which suggest which bundling strategy is optimal in various contexts through a simulation model that uses an optimization routine based on a genetic algorithm.

Pure vs. mixed bundling. The decision-maker has to choose which bundling strategy to use. Three bundling strategies defined and investigated in the literature are:

- *unbundling* [24] or *pure component strategy* [25] is defined as offering the products for sale separately;
- *pure bundling* is defined as offering products for sale exclusively in bundle form; pure bundling is sometimes called 'tying' in the economic and legal literature;
- *mixed bundling* is defined as offering products in both bundle and unbundled form [24].

These terms were first defined by Adam and Yellen [26]. They formulated the bundling problem and referred to it as *commodity bundling.* They analyzed the pure components strategy, pure bundling strategy and mixed bundling strategy in a monopoly and presented some implications of commodity bundling for public policy analysis. Schmalensee [27], Venkatesh and Mahajan [28], Hanson and Martin [19] and Venkatesh and Kamakura [29] investigated which bundling strategy could be more profitable, arriving at the conclusion that mixed bundling is better than the other two strategies.

In this thesis we consider a monopoly with a pure strategy approach. The main application is the composition of package tours with capacity constraints. In these packages events are sold together, so that tourists are alleviated from the task of deciding which events/locations to select from a subset of potential events/locations of a same type.

Price vs. product bundling. Another bundling classification criterion is the distinction between price bundling and product bundling.

Price bundling is the sale of two or more separate products in a package at a discount, without any integration of the products. The major application of price bundling is in the retail industry as highlighted by Banciu [25]). In addition several articles entail the application of price bundling in the Internet [30].

Product bundling can be defined as the *integration of two or more products or services at any price*. The term 'integration' means providing the consumer with value added as *compactness (as an integrated stereo systems), seamless interaction (PC systems), non duplication coverage (one-stop assurance), reduced risk (mutual fund) interconnectivity (telecom systems), enhanced performance (personalized dieting and exercise program) or convenience from an integrated bill (telecom calling plan)* [23].

Price bundling can be viewed as a *promotional tool* while *product bundling* can be seen as a *long term differentiation strategy.*

The Package Tour Composition Problem analyzed in this thesis is related to product bundling. Moreover we are particularly interested in models that use an optimization approach that simultaneously decides bundle design and price. Hanson and Martin [19] show that the single firm bundle pricing problem is naturally viewed as a disjunctive program which is formulated as a mixed integer linear programming problem. Multiple components and a variety of cost and reservation price conditions are investigated with this approach. Nevertheless in Hanson and Martin the model aims to determine bundles prices considering at most 21 single products. On the contrary in the model we propose the price of each bundle is given by the sum of the prices of the products included in the bundle with no discounts, but the elements (events/points of interest for tourists) are categorized in types and they are up to 100.

Customized bundling. *Customized bundling* is characterized by bundles being determined by the customer's choice. Hitt and Chen [31] results suggest that customized bundling has a number of advantages - both in theory and practice - over other bundling strategies in many relevant settings. They argue that *the use of customized bundling greatly simplifies the complexity of the problem especially of large number of goods.* Wu et al. [32] propose non-linear mixed-integer programming models to solve the customized bundle-pricing problem in which consumers are allowed to choose up to a given number of products out of a larger set.

Original contribution. In the scientific literature on bundling the focus is mainly on the economic aspects with very limited investigation from an operational research perspective. Very few models address the problem of the optimal composition of product bundles and no paper considers consumer economic evaluation of products in the bundle.

Britan and Ferrer[33] propose a model to determine the composition and price of a bundle so as to maximize total expected profit in the case of a company that operates in a competitive market and for a single bundle. Cataldo et al. [21] analyze the same problem for multiples bundles in a competitive market. However they deal with price bundling instead of product bundling. The number of elements in the bundle is up to 4. The similarity with the approach we propose is the structure of a two-phase problem: one phase is the pricing problem and the second is an optimal composition problem. In the model we propose we use column generation that is also a two-phases approach. Nevertheless our model phase 1 deals with composition of feasible bundling packages and phase 2 with the selection of packages with the largest value that become part of the optimal solution, considering also capacity constraints.

Mayer et al. [34] analyze a service provider's mixed bundling problem for services such as sporting events and holidays packages. Their objective is to maximize total revenue and the service provider has to determine static prices for each single product at the beginning of the selling period. The optimal package price must be chosen for a bundle that includes one unit of each single product. Because of capacity constraints, the availability of products can change over time so that consumers are forced to switch from their preferred subset of products to an alternative, following dynamic substitution. This work proposed a mixed-integer linear program and a meta-heuristic algorithm based on variable neighborhoods as a solution method. Mayers et al. consider not only the reservation prices but also the contingency ρ, that is the consumer relative perception of services, and they indicate whether consumers consider the single products to be substitutes ($\rho > 0$), complements ($\rho < 0$) or independent of each other ($\rho = 0$). The use of this parameter is necessary for the dynamic substitution mechanism employed in their model. The model we propose differs for the composition and pricing of packages and above all for the solution method since we use mathematical programming. Moreover in Mayers research the consumers are not considered as an aggregated demand, as in our thesis work, but they are grouped into a discrete set of individual consumers. In our model we consider a large number of products partitioned into several subsets according to their types in order to cope with the variety inherent to the tourist points of interest.

1.3 Scientific literature on tourism

In this section we give an overview of the scientific literature on tourism, based upon scientific papers that concern tourist district analysis, composition of packages for tourists or tour design. In particular we analyze three main topics: destination management, information technology applications and recommendation systems. In the last subsection we also present two package tour composition models already studied in the literature.

1.3.1 Destination management

The fundamental product in tourism is the destination experience; therefore competition is centered on destinations. Although competition occurs between airlines, tour operators, hotels, and other tourism services, this inter-enterprise competition is dependent upon and derived from the choices tourists make between alternative destinations [35]. The role of destination management is to manage and support the integration of different resources, activities and stake-holders through suitable policies and actions. Hence destination management (DM) requires governmental, decisional and functional competencies (planning, organization and control of business activities), which should generally be performed by public administrations [36].

Some contributions focused on the challenges of strategic management [37, 38, 39, 40]. Initially the concept of destination tended to coincide with large geographical areas, but later the research focus shifted towards a local level [39] and various types of local destinations were identified and analyzed [41, 42].

In this thesis we focus on tourist districts that can be recognized by the following constituent elements according to the canonical (or Marshall [43]) approach: (i) a well-defined geographical area, (ii) a set of small-medium enterprises backed up by larger-sized firms and (iii) a shared culture.

Sainaghi [44] raises the following questions: *In a tourist district where hundreds of independent players compete, often with radically different development visions, who should take on DM? How can these players come together to work toward certain goals? How can actions undertaken at a district level be brought into line with those done by individual local firms?.* His work proposes a methodology called Dynamic Destination Management Model (DDMM). One of the important areas included in DDMM is *new product development* that concerns *the creation of service and event packages.*Sainaghi defines packages that *can vary in size, and are usually limited to a core of services which are seen as central to the reference segment (accommodations, cultural attractions, and other recreational services) and may leave room for accessory components.*

Sainaghi claims that *new product development always 'blends' supervisory processes with activities carried out by firms and local public bodies* and underlines the problems related to supervisory packages of tourist districts as follows:

- lack of promo-commercialization channels;
- need to reach a sufficient 'critical mass' in terms of number of beds;

- different business needs by different district firms;
- difference in standards of quality.

Sainaghi methodology does not follow a mathematical approach; however it underlines the importance of the creation of service and event packages for tourist district enhancement that is the main motivation of this thesis.

1.3.2 Information technology in tourism

Several recent studies underline the role of information technology in the design of innovative tourism services. This includes the adoption of certain technologies, as discussed in McCabe et al. [45], and the review of the progress in information technology for tourism management, as illustrated in Buhalis and Law [46]. Buhalis and Law's findings state that the technological revolution led by the development of the Internet dramatically changed the market conditions for tourism organizations and lead to re-engineering of the entire process of developing, managing and marketing tourism products and destinations. They claim that *the future of e-tourism will support organizations to interact with their customers dynamically and consumers are becoming powerful and are increasingly able to determine elements of their tourism products. They are also much more sophisticated and experienced and therefore more difficult to please.*

In this thesis we aim to generate several unique solutions that consist of bundles of tourism products characterized by a certain pattern of product types (es. a certain number of restaurants, museums, parks, shopping outlets, concerts). The solution proposed by our model is able to cope with the complexity of consumer behavior because takes into account the demand towards each touristic package component.

1.3.3 Package tour composition and routing

Two recent papers address mathematical programming models for package tour composition; they mainly deal with routing problems.

Rodrìgues et al. [47] proposed a system that considers many objectives which tourists may have when planning their trip: minimization of distance traveled, minimization of cost, maximization of utility (estimated on the basis of tourist preferences and the importance of each activity). They also adjusting the time spent on each type of visit according to tourists' wishes. Furthermore they take into consideration multiple constraints for better adjusting their model to reality: in particular they use a multi-criteria method (MOAMP), which is a meta-heuristic algorithm for multi-objective programming based on tabu search which combines tools for solving selection problems with tools for solving the Traveling Salesman Problem.

Cardoso Neto et al.[48] study tourists routes at Vina del Mar in Chile. Their objective is to determine the best routes for five existing itineraries, that is five sets of locations to be visited, minimizing the time spent in traveling. The authors used an exact method by solving the mathematical model of the Traveling Salesman Problem and a heuristic method based on cheapest insertion.

Original contribution. The Package Tour Composition Problem presented in this thesis links destination management with revenue management. It provides a mathematical approach to the destination management problems described by Sainaghi [44] .

Our approach is not focused on technology but it provides the basis for decision support systems that tour operators can employ to take advantage of ICT. Our models are intended to support the organization of the tourist operators so that their revenues can be enhanced by offering a large range of unique solutions, taking into account consumers reservation prices, demand and availability for each point of interest or event.

An important feature of our model is the coordination between different destinations so that the flow of tourists is not concentrated on few overcrowded locations, but it is evenly distributed to avoid congestions and to allow all operators to count on a significant demand.

In this way not only profitability is maximized but also the utilization of resources is optimized and the experience from the viewpoint of the tourists is improved.

In conclusion we note the small number of exact mathematical models addressing package tour composition problems in the tourism literature.

1.4 Conclusions

The Package Tour Composition Problem presented in this thesis is a step in the direction of giving a scientifically sound answer to a well-known and compelling economical problem. The model we have defined deals with proposing several product solutions that not only maximize tour operators revenue but also maximize utility for tourists and event capacity usage. It is especially intended for Italian tourism districts of secondary importance, characterized by a high number of small operators.

In the Revenue Management literature and in the tourism literature no other model presents the characteristics of the PTCP.

In Chapter 2 we propose a mathematical model that solves a basic version of the PTCP and in Chapter 3 we propose some extensions.

Chapter 2

The Package Tour Composition Problem

In this chapter we propose mathematical programming models of the Package Tour Composition (PTC) Problem. In Section 1 we present a non-linear formulation of the PTC problem and some related computational results that show the need for alternative approaches. This motivates the use of column generation as a solution method. In Section 2 and Section 3 we describe two column generation models. The model described in Section 2 has a master problem with integer relaxed variables, while that described in Section 3 has a master problem with binary relaxed variables. In both cases a heuristic technique is proposed to find an integer feasible solution, using the package tours generated by the relaxed problem. In Section 4 we describe the instances we used in our computational tests and present computational results. In Section 5 we propose a parametric formulation and analyze model robustness.

2.1 The Package Tour Composition Problem: a non-linear model

The Package Tour Composition Problem deals with two entities: a given ground set of events and a variable set of package tours to be defined. The aim of the model is to generate package tours made by suitable event subsets to be visited by the tourists and also to decide the number of package tours of each kind to sell in order to maximize revenue.

In Figure 2.1 we show a scheme that represents the PTC problem.

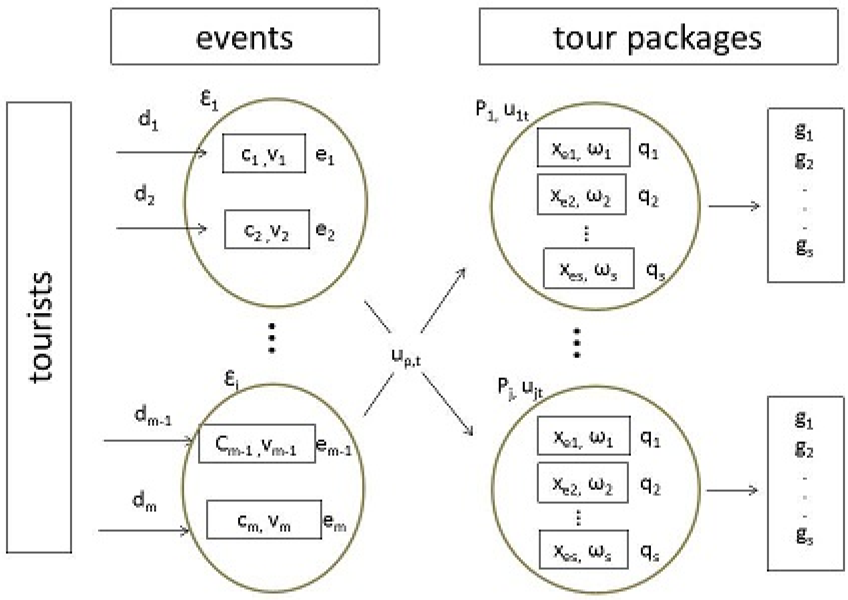

Figure 2.1: PTC Problem

Let $\mathcal{E}$ denote the set of available events. Each event $e \in \mathcal{E}$ is characterized by a capacity c_e, a value v_e and a demand d_e. The event capacity c_e represents the maximum number of tourists that can be accommodated at the event. The demand d_e is the predicted number of tourists that are willing to visit the event. The value v_e is the reservation price tourists are willing to pay for attending the event. Therefore the package tour generated by the model reflects the aggregated demand preferences: events with high demand, high value and high capacity are more likely to be included in the package tours generated by the model.

The event set $\mathcal{E}$ is partitioned into subsets $\mathcal{E}_t$ that correspond to event types. We indicate with $\mathcal{T}$ = the set of event types. This feature reflects the particular characteristics of a group of events. As discussed in Chapter 1 a possible definition of types could be: museums, parks, shopping outlets, churches, castles, restaurants, hotels, etc.

Differently from the classical Network Revenue management problem the composition of package tours is not known, but it is a variable. The generated package tours are grouped into package tour types; we indicate with $\mathcal{P}$ the set of package tour types, with $\mathcal{Q}_p$ the set of package tours of each type $p \in \mathcal{P}$ and with $\mathcal{Q}$ the set of all package tours. Package tour types are characterized by given composition parameters u_{pt} stating the number of events of each event type $t \in \mathcal{T}$ that belong to package tours of type $p \in \mathcal{P}$.

Therefore all package tours of the same type $p \in \mathcal{P}$ are made by the same types of events, but not necessarily by the same events.

The decisions are represented by two sets of discrete variables. Binary variables x_{eq} state whether an event $e \in \mathcal{E}$ belongs to package tours $q \in \mathcal{Q}$. Integer variables g_q indicate the number of identical package tours $q \in \mathcal{Q}$ to be put on sale.

Each package tour $q \in \mathcal{Q}$ is also characterized by a price, given by the sum of the values of the events it includes, i.e. $\sum_{e \in \mathcal{E}} v_e x_{eq}$ $\forall q \in \mathcal{Q}$.

2.1.1 A mathematical programming formulation

Data. The following data are given.

- A set $\mathcal{E}$ of events.
- A value v_e for each event $e \in \mathcal{E}$.
- A capacity c_e for each event $e \in \mathcal{E}$.
- A demand d_e for each event $e \in \mathcal{E}$.
- A set $\mathcal{T}$ of event types.
- A partition of the ground set $\mathcal{E}$ into subsets: $\mathcal{E} = \bigcup_{t \in \mathcal{T}} \mathcal{E}_t$.
- A set $\mathcal{P}$ of package tour types.
- A number u_{pt} of events for each event type $t \in \mathcal{T}$ and for each package type $p \in \mathcal{P}$ (composition pattern).

Variables. The following variables are used.

- A set $\mathcal{X}_p$ of packages for each package type $p \in \mathcal{P}$.
- A binary assignment variable x_{eq} for each event $e \in \mathcal{E}$ and each package $q \in \mathcal{X} = \bigcup_{p \in \mathcal{P}} \mathcal{X}_p$; it takes value 1 if and only if event e belongs to package q.

- An integer variable g_q for each package $q \in \mathcal{X}$; it indicates the number of packages q to be sold.

Note that the number of packages to be sold for each package type is not known: the cardinality of sets $\mathcal{X}_p$ must be estimated in some way to allow formulating the model. For instance in the non-linear model we present in paragraph 2.1.2 we consider the cardinality of $\mathcal{X}_p$ as the maximum value it can assume considering the combination of the elements of the subset $\mathcal{E}_t$ and the composition pattern u_{pt}.

Constraints. The following constraints define feasible solutions.

- Capacity constraints: the number of packages including each event $e \in \mathcal{E}$ cannot be larger than the event capacity c_e.

$$\sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{X}_p} x_{eq} g_q \leq c_e \quad \forall e \in \mathcal{E}.$$

- Demand constraints: the amount of packages including each event $e \in \mathcal{E}$ cannot be larger than the event demand d_e.

$$\sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{X}_p} x_{eq} g_q \leq d_e \quad \forall e \in \mathcal{E}.$$

- Package composition constraint: for each package type $p \in \mathcal{P}$, every package tour $q \in \mathcal{X}_p$ must comply with the composition pattern defined by the parameter u_{pt}.

$$\sum_{e \in \mathcal{E}_t} x_{eq} = u_{pt} \quad \forall p \in \mathcal{P},\ \forall q \in \mathcal{X}_p,\ \forall t \in \mathcal{T}.$$

Objective function. The objective is to maximize the overall value of the packages. The value of each package $q \in \mathcal{X}_p$ is the sum of the values of its events multiplied by the number g_q of packages to be sold.

$$z = \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{X}_p} \sum_{e \in \mathcal{E}} v_e x_{eq} g_q$$

A non-linear formulation. Following the definitions above, the PTC problem turns out to be a non-linear problem with integer and binary variables. The number of its variables must be determined according to an estimate of the cardinality of subsets $\mathcal{X}_p$. The model is as follows.

$$\text{maximize } z = \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{X}_p} \sum_{e\in\mathcal{E}} v_e x_{eq} g_q \tag{2.1}$$

$$\text{s.t. } \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{X}_p} x_{eq} g_q \leq \min\{d_e, c_e\} \quad \forall e \in \mathcal{E} \tag{2.2}$$

$$\sum_{e\in\mathcal{E}_t} x_{eq} = u_{pt} \quad \forall p \in \mathcal{P}\ \forall q \in \mathcal{X}_p\ \forall t \in \mathcal{T} \tag{2.3}$$

$$x_{eq} \text{ binary} \quad \forall e \in \mathcal{E}\ \forall q \in \mathcal{X}_p \tag{2.4}$$

$$g_q \text{ integer} \quad \forall p \in \mathcal{P}\ \forall q \in \mathcal{X}_p. \tag{2.5}$$

2.1.2 Solution of the non-linear model

We used a non-linear solver (BONMIN) to solve PTC problem instances. The number of the elements of $\mathcal{Q}_p\ \forall p$ is given by the number of simple combinations of u_{pt} events chosen from the subset $\mathcal{E}_t$ for each type $t \in \mathcal{T}$, that is

$$|Q_p| = \prod_{t\in\mathcal{T}} \binom{|\mathcal{E}_t|}{u_{pt}}$$

Table 2.1 reports the number of elements of $\mathcal{Q}_p\ \forall p$ for each instance. The instances were randomly generated and the cardinality of event set $\mathcal{E}$, event type set $\mathcal{T}$ and package type set $\mathcal{P}$ are presented in Table 2.2.

Table 2.2 reports also the computing time and error messages. Tests were done in a PC dual core 2.20 Ghz with 4GB of RAM. The timeout considered is 10 hours.

Table 2.2 Non-linear model execution time

Instance	package types	event types	events	time (sec.)	error message
BM01	2	3	6	58.67	-
BM02	2	3	10	timeout	-
BM03	2	3	25	189.7	too expensive
BM07	6	3	6	31.51	-
BM08	6	3	10	23740.30	-
BM09	6	3	25	58.86	too expensive

As we can see the solver was able to provide a solution just for very small instances.

We also remark that the solutions provided by the non-linear solver, when integrality conditions were relaxed, were fractional even though equivalent integer solutions exist.

Table 2.1 Number of elements of $\mathcal{Q}_p$ $\forall p$ for each instance

BM01				
u_{pt}	1	2	3	$\|Q_p\|$
1	1	1	1	6
2	2	1	2	2
BM02				
u_{pt}	1	2	3	$\|Q_p\|$
1	1	1	1	36
2	2	1	2	54
BM03				
u_{pt}	1	2	3	$\|Q_p\|$
1	1	1	1	576
2	2	1	2	8064
BM07				
u_{pt}	1	2	3	$\|Q_p\|$
1	1	1	1	6
2	2	1	2	2
3	1	2	1	4
4	2	1	1	4
5	1	1	2	4
6	2	2	1	2
BM08				
u_{pt}	1	2	3	$\|Q_p\|$
1	1	1	1	36
2	2	1	2	54
3	1	2	1	36
4	2	1	1	36
5	1	1	2	54
6	2	2	1	36
BM09				
u_{pt}	1	2	3	$\|Q_p\|$
1	1	1	1	576
2	2	1	2	8064
3	1	2	1	2016
4	2	1	1	2016
5	1	1	2	2304
6	2	2	1	7056

This motivates the use of an alternative approach based on column generation in which the variables are not enumerated a priori but generated dynamically.

2.1.3 Motivation for column generation

Column generation is often employed to solve the linear relaxation of discrete optimization problems that can be decomposed into smaller and easier sub-problems when some of their constraints are relaxed. The price to pay for the decomposition is the exponential number of variables (columns) of the extended formulation and this motivates the dynamic generation of its columns. A description of column generation is given in Appendix 1.

Here we have a different motivation: in our case columns generation is used to solve an extended linear model resulting from the linearization of a non-linear compact model. Column generation allows us to linearize model (2.1)-(2.5) and to solve its continuous relaxation to proven optimality, while keeping integer values for the x variables and only relaxing the integrality conditions on the g variables.

Model (2.1)-(2.5) is linearized at the expense of considering an exponential number of columns. We indicate by $\mathcal{Q}_p$ the set of all feasible packages of type $p \in \mathcal{P}$:

$$\mathcal{Q}_p = \{x \in \mathcal{B}^{|\mathcal{E}|} : \sum_{e \in \mathcal{E}_t} x_e = u_{pt} \forall t \in \mathcal{T}\} \quad \forall p \in \mathcal{P}.$$

Now we can reformulate model (2.1)-(2.5) as follows:

$$\text{maximize } z = \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}_p} \sum_{e \in \mathcal{E}} v_e x_{eq} g_q \tag{2.6}$$

$$\text{s.t. } \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}_p} x_{eq} g_q \leq \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{2.7}$$

$$g_q \text{ integer} \qquad \forall p \in \mathcal{P} \ \forall q \in \mathcal{Q}_p. \tag{2.8}$$

This extended formulation has an exponential number of variables, since the cardinality of each set $\mathcal{Q}_p$ is given by a (feasible) combination of events: for each event type $t \in \mathcal{T}$ a given number u_{pt} of events of type t must be chosen from $\mathcal{E}_t$ and this can be done in a number of different ways that is combinatorial in the size of $\mathcal{E}_t$. However model (2.6)-(2.8) is linear, because the binary x variables of model (2.1)-(2.5) are replaced by binary coefficients in model (2.6)-(2.8).

In the remainder we present two slightly different models of the PTC problem that are amenable to column generation: an integer master formulation (IMP) and a binary master formulation (BMP). In the integer master formulation, derived from model (2.6)-(2.8), the master problem has integer (relaxed) variables and the pricing problem has binary variables; on the contrary in the binary master formulation the master problem has binary (relaxed) variables and the pricing problem has integer variables. The solutions of the two proposed relaxed models provide dual bounds (upper bounds) for the PTC problem. In order to also find primal bounds (lower bounds) corresponding to feasible integer solutions, a heuristic technique is proposed. It consists of fixing a subset of columns generated by the column generation algorithm and to run a general-purpose integer linear programming solver to solve the master problem without relaxing the integrality restrictions. This column-generation-based heuristic algorithm turns out to be very fast and to provide feasible solutions that are very close to optimality.

2.2 Column generation with integer master problem variables (IMP)

We recall the extended linear formulation of the PTC problem obtained in the previous section.

$$\text{maximize } z = \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}_p} \sum_{e\in\mathcal{E}} v_e x_{eq} g_q \tag{2.9}$$

$$\text{s.t.} \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}_p} x_{eq} g_q \le \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{2.10}$$

$$g_q \text{ integer} \qquad \forall q \in \mathcal{Q} \tag{2.11}$$

Relaxing the integrality restrictions (2.11) on integer variables g into non-negativity conditions $g \ge 0$ we obtain the following relaxed model:

$$\text{maximize } z = \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}_p} \sum_{e\in\mathcal{E}} v_e x_{eq} g_q \tag{2.12}$$

$$\text{s.t.} \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}_p} x_{eq} g_q \le \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{2.13}$$

$$g_q \ge 0 \qquad \forall q \in \mathcal{Q} \tag{2.14}$$

This linear programming model has an exponential number of variables, because the cardinality of set $\mathcal{Q}$ grows combinatorially with the number of events and types. Column generation allows to generate the elements of $\mathcal{Q}$ dynamically.

2.2.1 Linear restricted integer master problem (LRIMP)

Using column generation, model (2.12)-(2.14) is decomposed into a master problem and a pricing problem. The master problem is characterized by the continuous variables g and by the capacity and demand constraint set (2.13). The pricing problem is characterized by binary variables x and by the package composition constraint set, which defines the set Q_p of feasible packages for each package type $p \in \mathcal{P}$. The linear restricted master problem (LRIMP) uses only a restricted subset $\bar{\mathcal{Q}}_p \subset \mathcal{Q}_p$ for each $p \in \mathcal{P}$ and the elements of subsets $\bar{\mathcal{Q}}_p$ for each package type $p \in \mathcal{P}$ are generated dynamically.

Master problem:

$$\text{maximize } z = \sum_{p \in \mathcal{P}} \sum_{q \in \bar{\mathcal{Q}}_p} w_q g_q \tag{2.15}$$

$$\text{s.t. } \sum_{p \in \mathcal{P}} \sum_{q \in \bar{\mathcal{Q}}_p} x_{eq} g_q \leq \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{2.16}$$

$$g_q \geq 0 \qquad \forall p \in \mathcal{P}\ \forall q \in \bar{\mathcal{Q}}_p. \tag{2.17}$$

The value w_q of each package $q \in \bar{\mathcal{Q}}_p$ is given by $w_q = \sum_{e \in \mathcal{E}} v_e x_{eq}$. We indicate with λ_e the non-negative dual variable associated with each of the constraints 2.16.

Pricing problem:

The pricing sub-problem can be decomposed and solved independently for each package type $p \in \mathcal{P}$.

$$\text{maximize } \overline{w}_p = \sum_{e \in \mathcal{E}} (v_e - \lambda_e) x_e \tag{2.18}$$

$$\text{s.t. } \sum_{e \in \mathcal{E}_t} x_e = u_{pt} \qquad \forall t \in \mathcal{T} \tag{2.19}$$

$$x_e \text{ binary} \qquad \forall e \in \mathcal{E}. \tag{2.20}$$

The values of the variables x in the columns with positive reduced cost $\overline{w}$ become the coefficients of the same column when it is inserted into the master problem. For this reason we use symbols x_e in the pricing sub-problem and x_{eq} in the master problem, with a little abuse of notation. For the same reason we use $\overline{w}_p$ to indicate the reduced cost of the optimal column for type $p \in \mathcal{P}$ in the pricing sub-problem and w_q to indicate the value of each column $q \in \mathcal{Q}_p$ in the master problem.

The pricing sub-problem is a discrete optimization problem that can be further decomposed into several sub-problems, one for each event type $t \in \mathcal{T}$. For each event type $t \in \mathcal{T}$, it is optimal to select the u_{pt} events of type t with the largest value of $(v_e - \lambda_e)$. Hence it is trivial to prove that this pricing sub-problem has the integrality property and it can be solved in polynomial time.

In usual applications of column generation this could be seen as a serious drawback, because it means that the dual bound provided by solving an extended formulation (with an exponential number of columns) with column generation cannot be better than the optimal value of the linear relaxation of the corresponding compact linear formulation. This is not the case for the model we are considering, since our extended formulation (2.9)-(2.11) is not derived from a linear compact formulation, but from the linearization of a non-linear compact formulation.

2.2.2 Heuristic restricted integer master problem (HRIMP)

Column generation provides an optimal solution to model (2.12)-(2.14), but such a solution is in general infeasible for model (2.9)-(2.11) owing to the relaxation of constraints (2.11). In order to find a feasible solution with integer values for the g variables, we use a heuristic method. We consider an integer linear programming problem whose columns are those that have been generated while solving model (2.12)-(2.14) with column generation. We denote as $\tilde{\mathcal{Q}}_p$ this set of columns for each package type $p \in \mathcal{P}$. The model we solve is the following.

$$\text{maximize } z = \sum_{p \in \mathcal{P}} \sum_{q \in \tilde{\mathcal{Q}}_p} w_q g_q \tag{2.21}$$

$$\text{s.t. } \sum_{p \in \mathcal{P}} \sum_{q \in \tilde{\mathcal{Q}}_p} x_{eq} g_q \leq \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{2.22}$$

$$g_q \text{ integer} \qquad \forall p \in \mathcal{P} \ \forall q \in \tilde{\mathcal{Q}}_p. \tag{2.23}$$

We call HRIMP this restricted master problem with integer variables and we solve it directly with an integer linear programming solver.

2.3 Column generation with binary master problem variables (BMP)

In this section we present a second model for solving the PTC problem with column generation, where the master problem has binary (relaxed) variables and the pricing problem has integer variables. Hence each column in the master problem does not represent an individual package (to be put on sale a variable number of times), but a unique bunch of packages of the same type, i.e. made according to the same events pattern but not necessarily including the same events. The motivation for this alternative formulation consits in searching for more homogeneous solutions and in using a relaxed formulation that allows to introduce constraints that limit the number of tourist to be assigned to each package tour, as in the model with groups presented in Chapter 3.

2.3.1 Formulation

Data. The data are the same as those used to define the IMP model in the previous section.

Variables. The following variables are used.

- A set $\mathcal{G}_p$ of bunches of packages for each package type $p \in \mathcal{P}$.
- An integer variable n_q for each bunch of packages $q \in \mathcal{G} = \bigcup_{p \in \mathcal{P}} \mathcal{G}_p$, indicating how many individual packages are included in bunch $q \in \mathcal{G}$.
- Integer variables y_{eq} indicating how many individual packages in bunch $q \in \mathcal{G}$ include event $e \in \mathcal{E}$.
- A binary variable h_q for each bunch of packages $q \in \mathcal{G}$, indicating whether it belongs to the solution or not.

Constraints. The following constraints define feasible solutions.

- Capacity constraints: the overall number of tourists assigned to each event $e \in \mathcal{E}$ cannot be larger than the event capacity c_e.

$$\sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{G}_p} y_{eq} h_q \leq c_e \quad \forall e \in \mathcal{E}.$$

- Demand constraints: the overall number of tourists assigned to each event $e \in \mathcal{E}$ cannot be larger than the event demand d_e.

$$\sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{G}_p} y_{eq} h_q \leq d_e \quad \forall e \in \mathcal{E}.$$

- Package composition constraint: for each package type $p \in \mathcal{P}$ all packages in each bunch $q \in \mathcal{G}_p$ must comply with the composition pattern defined by u_{pt}.

$$\sum_{e \in \mathcal{E}_t} y_{eq} = u_{pt} n_q \quad \forall p \in \mathcal{P},\ \forall q \in \mathcal{G}_p,\ \forall t \in \mathcal{T}.$$

- Non-repetition constraint: the number of individual packages of each type $p \in \mathcal{P}$ including each event $e \in \mathcal{E}$ cannot be larger than the number of individual packages in the bunch $q \in \mathcal{G}_p$.

$$y_{eq} <= n_q \quad \forall p \in \mathcal{P},\ \forall q \in \mathcal{G}_p,\ \forall e \in \mathcal{E}.$$

- In this model we also require solutions to include at most one column for each package type. This can be imposed without loss of generality, because every two columns representing package bunches of the same type can be replaced by a single column, whose value is the sum of the values of the two original columns. Therefore these constraints are useful to avoid the presence of a combinatorial number of equivalent solutions.
$$\sum_{q \in \mathcal{G}_p} h_q \leq 1 \quad \forall p \in \mathcal{P}.$$

Objective function. The objective is to maximize the overall value of the packages. The value of each package bunch $q \in \mathcal{G}_p$ is the sum of the values of the packages in it.

$$z = \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{G}_p} \sum_{e \in \mathcal{E}} v_e y_{eq} h_q.$$

A non-linear formulation. Following the definitions above, the following model is obtained. As in the previous case, it is a non-linear model with integer and binary variables.

$$\text{maximize } z = \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{G}_p} \sum_{e \in \mathcal{E}} v_e y_{eq} h_q \tag{2.24}$$

$$\text{s.t. } \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{G}_p} y_{eq} h_q \leq \min\{d_e, c_e\} \quad \forall e \in \mathcal{E} \tag{2.25}$$

$$\sum_{e \in \mathcal{E}_t} y_{eq} = u_{pt} n_q \quad \forall p \in \mathcal{P} \; \forall q \in \mathcal{G}_p \; \forall t \in \mathcal{T} \tag{2.26}$$

$$y_{eq} <= n_q \quad \forall q \in \mathcal{G} \; \forall e \in \mathcal{E} \tag{2.27}$$

$$\sum_{q \in \mathcal{G}_p} h_q \leq 1 \quad \forall p \in \mathcal{P} \tag{2.28}$$

$$h_q \text{ binary} \quad \forall q \in \mathcal{G} \tag{2.29}$$

$$y_{eq}\text{integer} \quad \forall e \in \mathcal{E} \; \forall q \in \mathcal{G} \tag{2.30}$$

$$n_q\text{integer} \quad \forall q \in \mathcal{G}_p. \tag{2.31}$$

2.3.2 An extended linear formulation

The non-linear compact model (2.24)-(2.31) can be linearized into an extended formulation, including all feasible integer columns $q \in \mathcal{Q} = \bigcup_{p \in \mathcal{P}} \mathcal{Q}_p$, where

$$\mathcal{Q}_p = \{(y, n) \in \mathcal{Z}_+^{|\mathcal{E}|} \times \mathcal{Z}_+ : \sum_{e \in \mathcal{E}_t} y_e = u_{pt} n \; \forall t \in \mathcal{T}, y_e <= n\} \quad \forall p \in \mathcal{P}.$$

The linear extended formulation reads as follows.

$$\text{maximize } z = \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}_p} \sum_{e\in\mathcal{E}} v_e y_{eq} h_q \tag{2.32}$$

$$\text{s.t. } \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}_p} y_{eq} h_q \le \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{2.33}$$

$$\sum_{q\in\mathcal{Q}_p} h_q \le 1 \qquad \forall p \in \mathcal{P} \tag{2.34}$$

$$h_q \text{ binary} \qquad \forall q \in \mathcal{Q} \tag{2.35}$$

Relaxing the integrality restrictions (2.35) on binary variables h into constraints of the form $0 \le h \le 1$, we obtain the following relaxed model:

$$\text{maximize } z = \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}_p} \sum_{e\in\mathcal{E}} v_e y_{eq} h_q \tag{2.36}$$

$$\text{s.t. } \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}_p} y_{eq} h_q \le \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{2.37}$$

$$\sum_{q\in\mathcal{Q}_p} h_q \le 1 \qquad \forall p \in \mathcal{P} \tag{2.38}$$

$$0 \le h_q \le 1 \qquad \forall q \in \mathcal{Q} \tag{2.39}$$

This linear programming model has an exponential number of variables h, because the cardinality of set $\mathcal{Q}$ grows combinatorially with the number of events and types. Column generation allows to generate the elements of $\mathcal{Q}$ dynamically.

2.3.3 Linear restricted binary master problem (LRBMP)

We use column generation to solve the continuous relaxation of the extended linear formulation (2.36)-(2.39).

The master problem is characterized by the continuous variables h and by the capacity and demand constraint set (2.37). The pricing sub-problem is characterized by integer variables y and n and by package composition constraints and non-repetition constraints. To avoid unboundedness in the pricing sub-problem and infeasibility in the master problem, we must also insert an additional constraint on the maximum number of packages in a bunch, as follows:

$$y_e <= \min\{c_e, d_e\} \quad \forall e \in \mathcal{E}.$$

The linear restricted binary master problem (LRBMP) includes only a restricted subset $\bar{\mathcal{Q}}_p \subset \mathcal{Q}_p$ for each $p \in \mathcal{P}$ and the elements of subsets $\bar{\mathcal{Q}}_p$ for each package type $p \in \mathcal{P}$ are generated dynamically.

Master problem:

$$\text{maximize } z = \sum_{p\in\mathcal{P}} \sum_{q\in\bar{\mathcal{Q}}_p} w_q h_q \tag{2.40}$$

$$\text{s.t. } \sum_{p\in\mathcal{P}} \sum_{q\in\bar{\mathcal{Q}}_p} y_{eq} h_q \leq \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{2.41}$$

$$0 \leq h_q \leq 1 \qquad \forall p \in \mathcal{P}\ \forall q \in \bar{\mathcal{Q}}_p. \tag{2.42}$$

The value w_q of each bunch of packages $q \in \bar{\mathcal{Q}}_p$ is given by $w_q = \sum_{e\in\mathcal{E}} v_e y_{eq}$. We indicate with μ_e the non-negative dual variable associated with each of the constraints (2.41).

Pricing problem:

The pricing sub-problem can be decomposed and solved independently for each package type $p \in \mathcal{P}$.

$$\text{maximize } \overline{w}_p = \sum_{e\in\mathcal{E}} (v_e - \mu_e) y_e \tag{2.43}$$

$$\text{s.t. } \sum_{e\in\mathcal{E}_t} y_e = u_{pt} n \qquad \forall t \in \mathcal{T} \tag{2.44}$$

$$y_e \leq \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{2.45}$$

$$y_e \leq n \qquad \forall e \in \mathcal{E} \tag{2.46}$$

$$y_e \text{ integer} \qquad \forall e \in \mathcal{E} \tag{2.47}$$

$$n \text{ integer} \tag{2.48}$$

2.3.4 Heuristic restricted binary master problem (HRBMP)

To compute an integer feasible solution we use the same technique described for the integer master problem. We consider an binary linear programming problem whose columns are those that have been generated while solving model (2.36)-(2.39) with column generation. We denote as $\tilde{\mathcal{Q}}_p$ this set of columns for each package type $p \in \mathcal{P}$. The model we solve is the following.

$$\text{maximize } z = \sum_{p\in\mathcal{P}} \sum_{q\in\tilde{\mathcal{Q}}_p} \tilde{w}_q h_q \tag{2.49}$$

$$\text{s.t. } \sum_{p\in\mathcal{P}} \sum_{q\in\tilde{\mathcal{Q}}_p} y_{eq} h_q \leq \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{2.50}$$

$$h_q \text{ binary} \qquad \forall p \in \mathcal{P}\ \forall q \in \tilde{\mathcal{Q}}_p. \tag{2.51}$$

We call HRBMP this restricted master problem with binary variables and we solve it directly with an integer linear programming solver.

2.4 Computational tests

2.4.1 Standard instances description

We constructed six data-sets: 25t3, 25t6, 50t3, 50t6, 100t3 and 100t6. They differ from one another for the number of events and the number of event types. The number of package types remains constant and is equal to five. The number of event types is equal to three or six and the number of events is equal to 25, 50 or 100 as displayed in Table 2.3.

Table 2.3 Standard instances

Instance	25t3	25t6	50t3	50t6	100t3	100t6
Package types	5	5	5	5	5	5
Events	25	25	50	50	100	100
Event types	3	6	3	6	3	6

In order to construct the instances we generated the distribution model parameters (event value, event demand, event capacity, package composition, and event types) at random with a uniform distribution of probability within the ranges reported in Table 2.4.

Table 2.4 Standard Instances: parameters dimensions

Instance	Minimum	Maximum
Event value	10	50
Event demand	50	100
Event capacity	40	200

The u_{pt} values are generated at random with a uniform distribution of probability within the ranges reported in Table 2.5.

Table 2.5 Standard Instances: u_{pt} generation

$\|\mathcal{T}\|$	Minimum	Maximum	$\sum_{p \in \mathcal{P}} u_{pt}$
3	0	3	3 or 4
6	0	2	6

Overall actual capacity value. The overall actual capacity value (OACV) is defined as the value of all events that belong to the event set multiplied by their actual capacity, that is the minimum among c_e and d_e for each event $e \in \mathcal{E}$:

$$OACV = \sum_{e \in \mathcal{E}} v_e \min\{d_e, c_e\}.$$

The value OACV indicates the maximum potential revenue that can be obtained in an ideal case. Therefore it is an upper bound for all feasible solutions of our models. The OAVC values for our data-set are displayed in Table 2.6.

Table 2.6 Standard instances: overall actual capacity values.

Instance	25t3 25t6	50t3 50t6	100t3 100t6
OACV	46142	100440	205708

2.4.2 Computational results

In this subsection we describe and compare the results obtained with the standard instance. To assess the quality of the solutions obtained we compare them against upper and lower bounds. In particular, the solutions obtained from column generation provide lower bounds and they are compared with the upper bound given by the OACV value. The solutions produced by the heuristic technique based on the HRMP models provide upper bounds to the optimum value and they are compared with the lower bound given by column generation. When analyzing the results obtained from our models, besides the value of the objective function and the computing time, we also consider another relevant indicator, that is package homogeneity.

If a model provides a solution that uses only one or few package types, that solution is considered a homogeneous solution. If a model provides a solution that uses several package tour types, that solution is considered a heterogeneous solution.

The best alternative between a homogeneous and a heterogeneous solution depends on the application of the model. A homogeneous solution selects and concentrates tourists in the more profitable package tour types. Instead a heterogeneous solution privileges the offer of a larger number of package tour types.

In the tables reported in the remainder of this section we use the following terminology:

- 'LRMP o.f.' is the optimal value of the LRMP.
- 'HRMP o.f.' is the optimal value of the heuristic model.
- 'LRMP gap' is the percentage gap between the LRMP o.f. and OACV.
- 'HRMP gap' is the percentage gap between the HRMP o.f. and LRMP o.f..
- 'LRMP time' is the computing time (in seconds) to solve the LRMP model.
- 'HRMP time' is the computing time (in seconds) to compute a heuristic solution. This is the sum of two terms: the 'LRMP time' (because the LRMP model generates the columns used by the HRMP model) and the time taken to solve the heuristic model.

Table 2.7 Standard instances results: optimal values and execution times.

Instances	25t3	25t6	50t3	50t6	100t3	100t6
Instances						
package types	5	5	5	5	5	5
events	25	25	50	50	100	100
event types	3	6	3	6	3	6
OACV	46142	46142	100440	100440	205708	205708
Results						
LRMP o.f.	46142	39036	100440	65078	205708	175246
LRMP gap	0%	15%	0%	35%	0%	15%
IMP						
HRMP o.f.	46142	39034	100440	65077	205611	175165
HRMP gap	0%	0.01%	0%	0%	0.05%	0.05%
LRMP time	1.4	1.4	2.1	2.0	3.8	5.5
HRMP time	1.5	1.4	2.2	2.1	3.9	7.3
BMP						
HRMP o.f.	31489	33369	95805	61757	139737	116298
HRMP gap	31.76%	14.52%	4.61%	5.10%	32.07%	33.64%
LRMP time (sec)	1.2	1.7	2.4	3.1	5.1	7.3
HRMP time (sec)	1.2	1.7	2.4	3.1	5.1	7.3

We compare the results obtained when solving the standard instances with IMP and BMP models. Afterwards we also analyze the characteristics of heuristic solutions obtained in both cases.

Table 2.7 reports the results obtained with IMP and BMP models.

LRMP solution quality. In general the optimal solution of the master problem with both models is a convex combination of columns where the variables g_q and h_q assume fractional values. As we note in Table 2.8 in the columns that belongs to the optimal LRMP solution the amount of non-integer number of tourists in each columns are substantial.

Table 2.8 Standard instances results: integer columns in the LRMP solution

IMP LRMP solution						
Dataset	total tourists	Total tourists on integer columns	% tourists on integer columns	total columns	integer columns	% integer columns
25t3	504,68	-	0%	25	-	-
25t6	204,00	19	9%	17	3	18%
50t3	980,23	287	29%	50	13	26%
50t6	279,00	-	0%	24	-	-
100t3	2002,27	194	10%	100	11	11%
100t6	851,67	-	0%	74	-	-
BMP LRMP solution						
Dataset	total tourists	Total tourists on integer columns	% tourists on integer columns	total columns	integer columns	% integer columns
25t3	2,45	-	-	25	-	-
25t6	2,43	-	-	17	-	-
50t3	2,19	-	-	49	-	-
50t6	2,18	-	-	24	-	-
100t3	2,57	-	-	90	-	-
100t6	2,34	-	-	74	-	-

Analyzing the LRMP solution results, shown in Figure 2.2, we can note that the gap between the LRMP objective function and the OACVs are null for the instances characterized by three event types, while instances with six event types show a larger gap. The reason for this is further investigated in the following subsubsection.

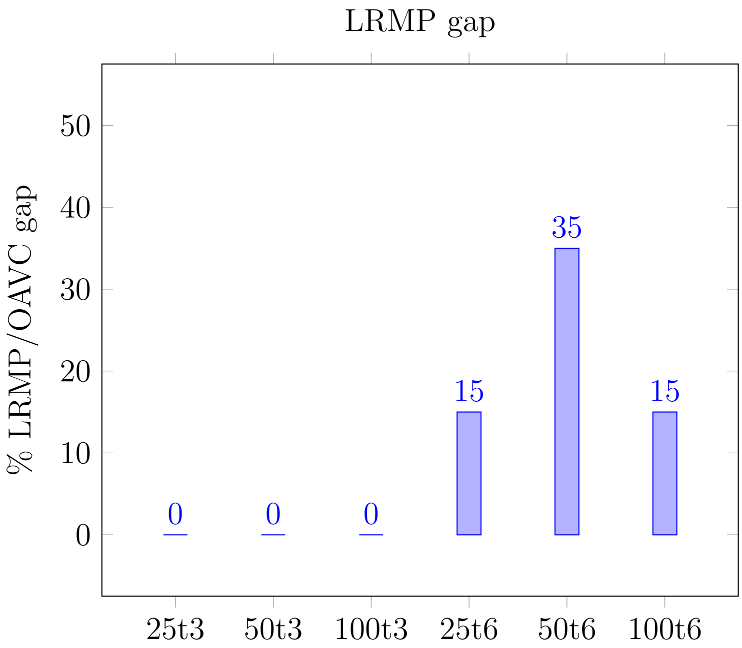

Figure 2.2: Standard isntances: LRMP gap.

Heuristic solution quality. Figures 2.3 and 2.4 display the gap between the heuristic value and the lower bound given by column generation. The HRIMP heuristic model shows an excellent performance and the gap varies from 0 to 0.05%. The HRBMP heuristic model instead shows a gap ranging from 4% to about 35% with no apparent regularity.

LRMP and HRMP performance. Figure 2.5 shows a comparison between the execution time of LRMP and HRMP models. We note a rising trend in the execution time of the LRMP models when the number of events grows; this is intuitive because larger capacity values allow for the generation of more columns. For most instances the time needed to generate a heuristic solution from the set of columns produced by column generation turns out to be negligible.

Homogeneity of the solutions. In this paragraph we analyze the homogeneity of the solutions. In particular we consider

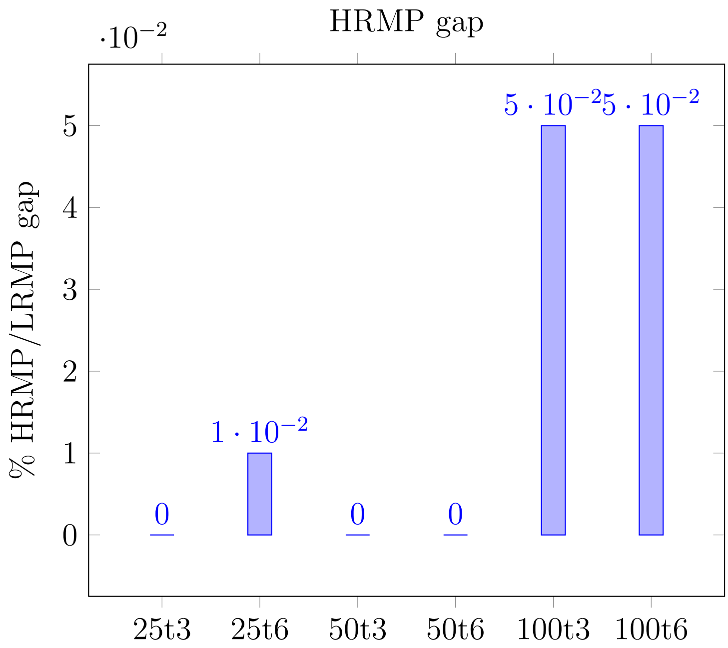

Figure 2.3: Standard instances: IMP HRMP gap

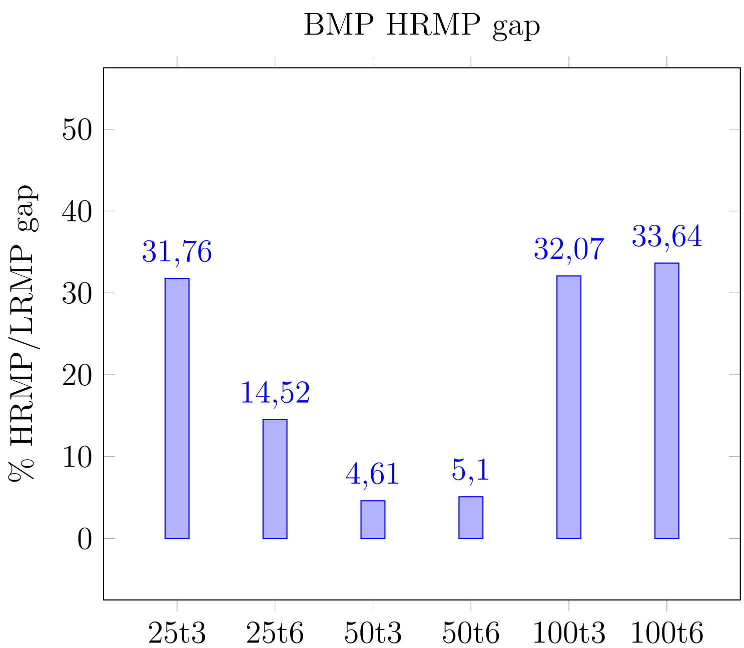

Figure 2.4: Standard instances: BMP HRMP gap.

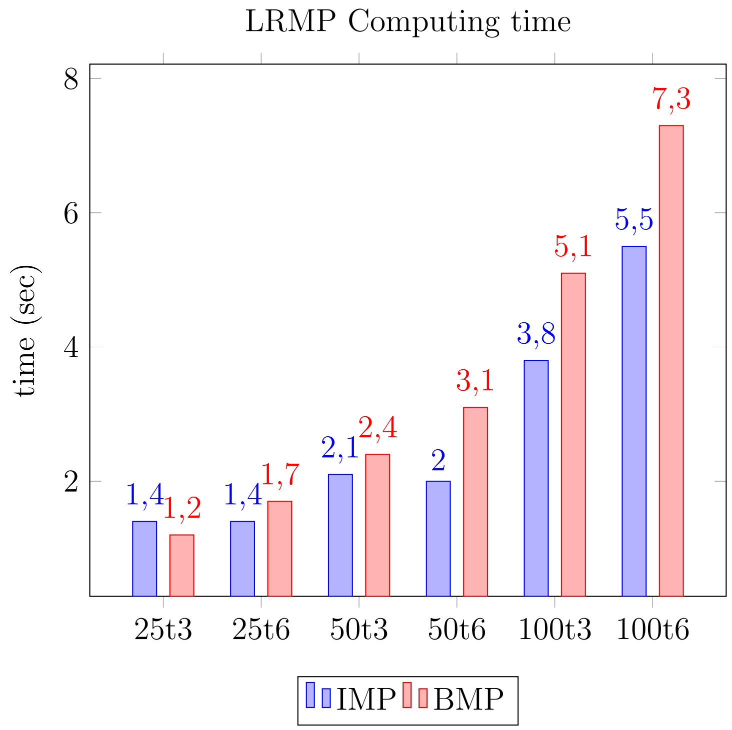

Figure 2.5: Standard instances: Computing time.

- the number of columns generated by the column generation algorithm, 'GENCOL'.
- the number of basic columns in the LRMP solution, 'LRMPCOL'.
- the number of columns selected in the solution of the HRMP model, 'HRMPCOL'.

Table 2.9 displays the results for the IMP and the BMP formulations.

The number of columns generated during the execution of the column generation algorithm shows a rising trend as the number of events grows. This trend is similar with both IMP and BMP models.

The number of columns in the heuristic solution increases with the number of events and decreases with the number of event types.

However the two models produce quite different results in terms of solutions homogeneity. With the BMP heuristic model the number of columns in the HRMP solution is limited to one or two. In other terms the BMP heuristic model generates discrete solutions using very few different types of packages, i.e. it yields very homogeneous solutions. On the other hand the IMP heuristic model generates discrete solutions using a richer mix of packages. This also explains the already mentioned better quality of HRIMP solutions compared to HRBMP solutions.

Table 2.9 Standard instances results: homogeneity.

Instances	25t3	25t6	50t3	50t6	100t3	100t6
Package types	5	5	5	5	5	5
Events	25	25	50	50	100	100
Event types	3	6	3	6	3	6
IMP						
GENCOL	50	54	103	75	160	245
LRMPCOL	25	17	50	24	100	74
HRMPCOL	27	21	58	28	116	93
BMP						
GENCOL	50	67	95	105	185	242
LRMPCOL	25	17	49	25	90	74
HRMPCOL	2	1	2	1	1	1

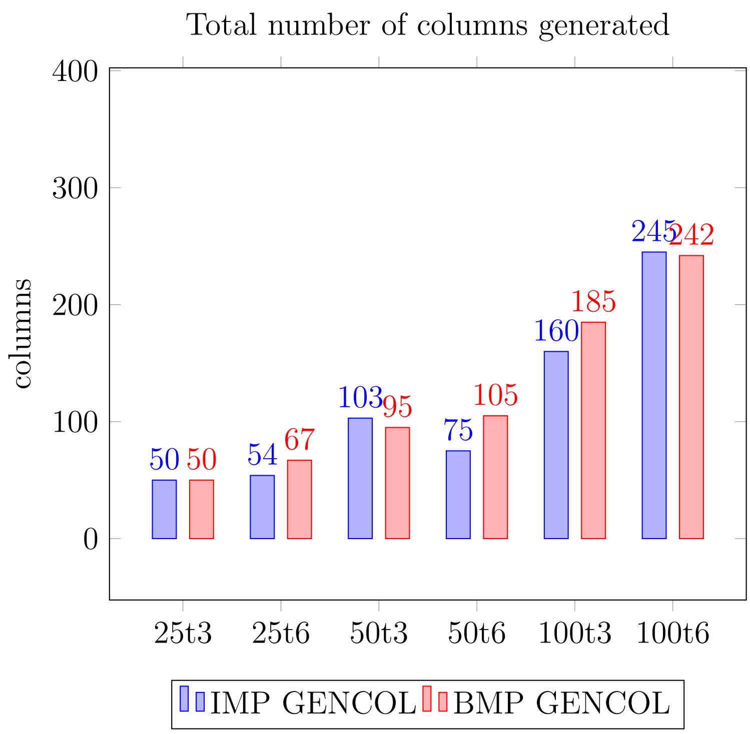

Figure 2.6: Standard instances: Generated columns

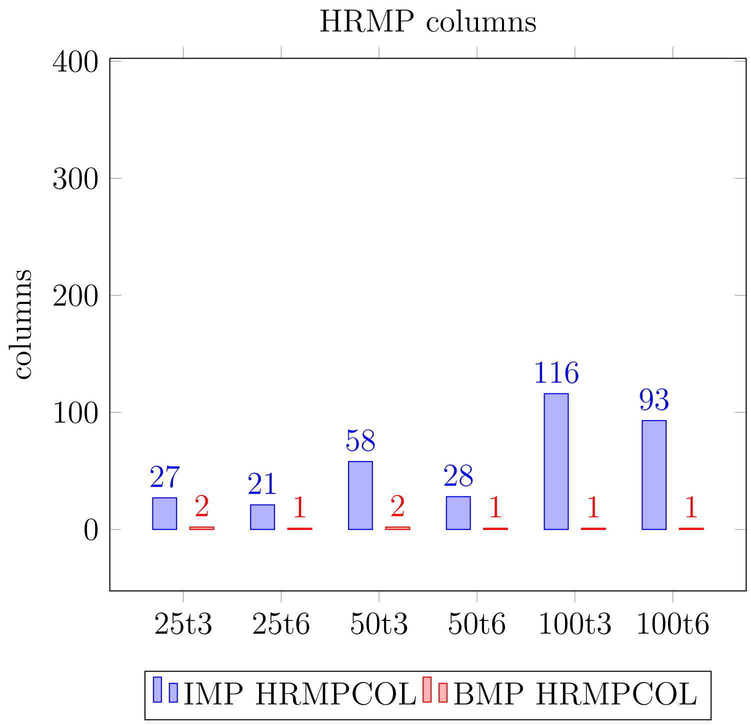

Figure 2.7: Standard instances: HRMP columns

2.4.3 Final Comments

The IMP model provides solutions of high LRMP quality in instances characterized by three events types and LRMP quality decrease for instances characterized by six event types. Therefore this suggests that an increase in event types can have a negative effect in LRMP quality, this hypothesis is further investigated in the event type variation analysis. The IMP model provides solutions of high HRMP quality.

The execution time of the IMP model sees a rising trend with a number of events increase. The capacity growth allows the model to generate more columns. When the generated columns grow also the execution time grows.

The solution provided by the IMP HRMP model has a similar package tour pattern as the solution provided by the IMP LRMP model. Both solutions use several package/columns in the optimal solution. The composition of the IMP HRMP package tour used in the optimal solution in terms of product type is further investigated in the product type variation subsection.

The BMP model provides a LRMP solution that follows the same trend as IMP LRMP solution in terms of LRMP quality. Nevertheless the BMP model provides solutions of poor HRMP quality. The HRMP quality does not seem to follow a particular trend as the number events and the number of event types grows.

The solution provided by the BMP HRMP model is characterized by a high homogeneity because the columns that belong to the optimal solution are just one or two and therefore belong to a very few number of package types.

2.5 Parametric Analysis

This section is divided into several parts: in subsection 2.5.1 we analyze some variations on the data-set and we perform a parametric anaysis to assess the robustness of our approach; in subsection 2.5.2 we present the results that are obtained by changing some parameters of the models: the number of events, the number of event types and the number of package types. Finally we draw some conclusions in subsection 2.5.3.

2.5.1 A parametric formulation

The motivation for the tests described hereafter is the search for less homogeneous solutions with the HRBMP model. Less homogeneous solutions are also expected to be better in terms of the objective function, resembling those obtained with the HRIMP model.

We recall that in the standard BMP formulation each variable y_e in the pricing subproblem is limited by the actual event capacity, that is $\min\{d_e, c_e\}$. This leads to the generation of very dense columns. Consequently these columns exhaust some event capacities and therefore it is difficult to combine more than two packages together; in turn this leads to poor quality of the solutions, as underlined in the previous subsubsection.

In order to force column generation based on the BMP model to produce less heterogeneous solutions, we impose an additional constraint limiting the value of the y variables. Then we perform a parametric analysis on the right hand side of such constraints. We call this formulation PBMP, where P stands for Parametric. The right hand side, indicated by l, ranges from 5 to the largest actual capacity among all events, i.e. $\max_{e\in\mathcal{E}} \min\{d_e, c_e\}$. Obviously there is no point in testing larger values, because the problem would be unfeasible. The PBMP model is as follows.

Master problem:

$$\text{maximize } z = \sum_{p\in\mathcal{P}} \sum_{q\in\bar{\mathcal{Q}}_p} w_q h_q \tag{2.52}$$

$$\text{s.t. } \sum_{p\in\mathcal{P}} \sum_{q\in\bar{\mathcal{Q}}_p} y_{eq} h_q \leq \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{2.53}$$

$$0 \leq h_q \leq 1 \qquad \forall p \in \mathcal{P}\ \forall q \in \bar{\mathcal{Q}}_p. \tag{2.54}$$

Pricing problem:

The pricing sub-problem for each package type $p \in \mathcal{P}$ is the following.

$$\text{maximize } \overline{w}_p = \sum_{e \in \mathcal{E}} (v_e - \mu_e) y_e \tag{2.55}$$

$$\text{s.t. } \sum_{e \in \mathcal{E}_t} y_e = u_{pt} n \qquad \forall t \in \mathcal{T} \tag{2.56}$$

$$y_e \leq l \qquad \forall e \in \mathcal{E} \tag{2.57}$$

$$y_e \leq n \qquad \forall e \in \mathcal{E} \tag{2.58}$$

$$y_e \text{ integer} \qquad \forall e \in \mathcal{E} \tag{2.59}$$

$$n \text{ integer} \tag{2.60}$$

Parametric analysis of instance 25t6 was done and five u_{pt} variations were tested; average values are reported in Table 2.10. The results of parametric analysis on all instances are displayed in Appendix 3.

Table 2.10 Parametric formulation: results.

l	1	5	8	10	25	50	BMP
LRMP o.f.	42265.5	42265.5	42265.5	42265.5	42265.5	42265.5	42265.5
LRMP gap	8%	8%	8%	8%	8%	8%	8%
HRMP o.f.	41876	39354	38071	37543	32051	19340	31832
HRMP gap	1%	7%	10%	11%	24%	54%	25%
time opt	9.7	2.3	2.7	2.2	1.6	1.6	1.5
time rnd	9.8	2.3	2.7	2.3	1.7	1.6	1.5

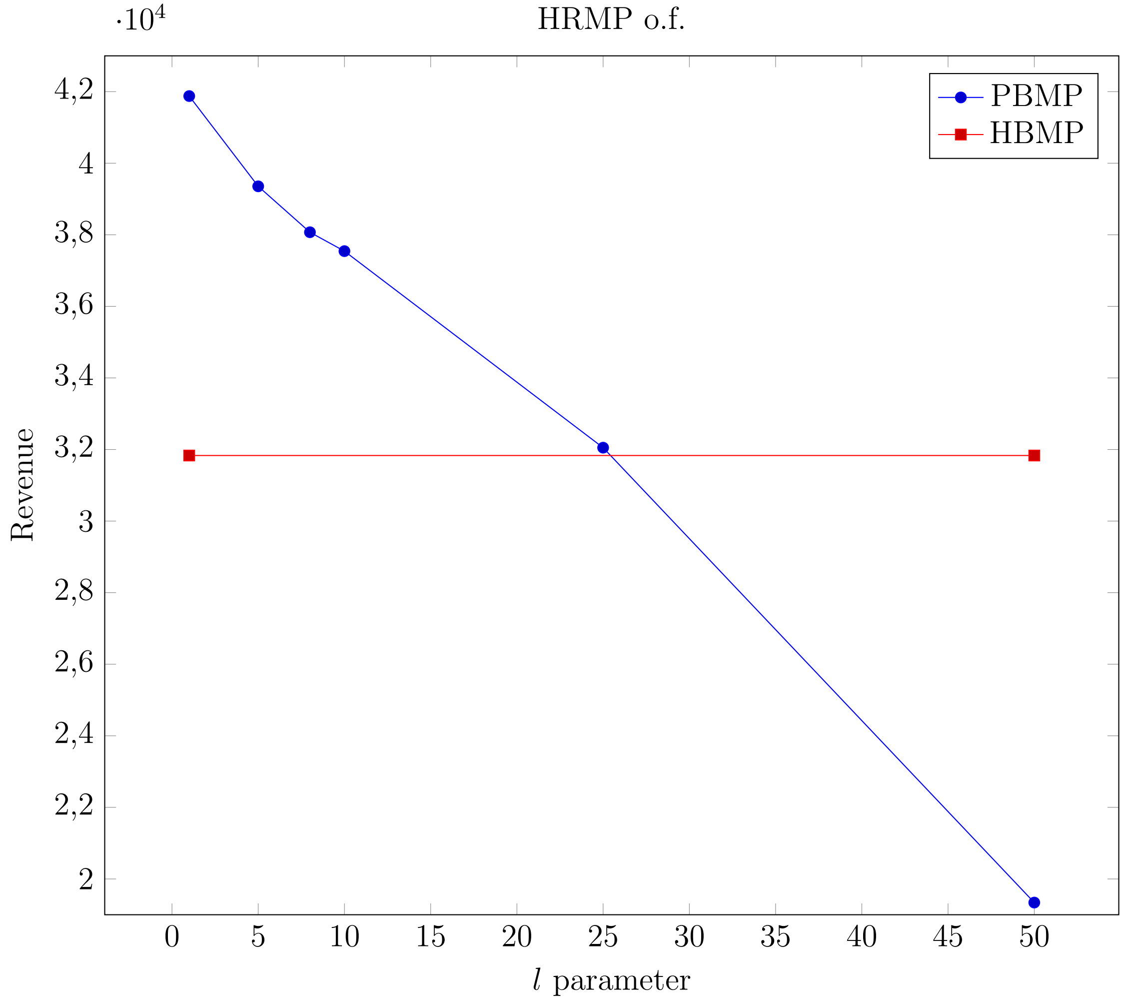

Figure 2.8: Heuristics revenue of parametric BMP compare with HBMP revenue

Using the standard BMP formulation (rightmost column) the HRBMP solution objective value is 31832. As we note in in Figure 2.8 the Parametric formulation performs better than the BMP solution for values lower than 25 that numerically is the half the maximum the event value can assume (see Table 2.4). This indicates that it is suitable to investigate a formulation that eliminates the actual capacity limiting constraint (2.45) from the pricing problem.

Comments. An extension related to this parametric analysis concerns the generation of packages for groups instead of single tourists and it will be described in the next chapter.

2.5.2 Model robustness

The experiments presented hereafter are motivated by the need of assessing the robustness of the solutions provided by our models. For this purpose we select in turn a parameter of the model and we study the relationship between its value and the outcome of the model. We repeat this analysis for three fundamental parameters of our instances: the number of package types, the number of event types and the number of events.

Number of package types In Table 2.11 we present the results obtained with instance 50t6 when the number of package types is changed. Every time a new package type is added, a new row in the composition parameter u_{pt} is also added. We call these instances $VP3, \ldots, VP9$, where the last number corresponds to the number of package types.

We underline that changes in the parameter u_{pt} imply significant changes in the results because if a package type with a more convenient u_{pt} is added, the optimal value increases; on the other hand if a package with a less convenient composition vector u_{pt} is added, the optimal value remains stable and the new package remains unused.

As shown in Figure 2.9, when the number of package types grows the gap between the LRMP solution and the OACV decreases. This trend is not linear; higher lower bounds are achieved due to the use of more convenient package types.

As shown in Figure 2.10, the value of the heuristic solutions is stable for instances $VP4$, $VP5$ and $VP6$ and then it increases. For instances $VP07$, $VP08$ and $VP09$ the primal-dual gap is larger because the increase in the upper bound (dual bound) is larger than the increase in the lower bound (primal bound).

As shown in Table 2.11 the computing time for column generation and for the heuristic model grows with the number of package types, because more

Table 2.11 Variation of $\mathcal{P}$: optimal values and computing time (sec.).

Variation of $\mathcal{P}$							
Instance	$VP3$	$VP4$	$VP5$	$VP6$	$VP7$	$VP8$	$VP9$
LRMP o.f.	53.473	65.078	65.078	65.078	76.512	91.646	91.646
LRMP gap	47%	35%	35%	35%	24%	9%	9%
IMP							
HRMP o.f.	53.467	65.077	65.077	65.078	76.475	91.607	91.607
HRMP gap	0.01%	0.00%	0.00%	0.00%	0.05%	0.04%	0.04%
LRMP time	1.2	2.1	2.0	3.2	3.1	4.4	6.0
HRMP time	1.3	2.1	2.1	3.3	23.2	4.5	6.4
BMP							
HRMP o.f.	52.262	61.757	61.757	61.757	61.757	64.286	70.871
HRMP gap	2.26%	5.10%	5.10%	5.10%	19.28%	29.85%	22.67%
LRMP time	2.0	3.0	3.1	4.1	4.1	4.2	4.4
HRMP time	2.0	3.1	3.1	4.1	4.1	4.3	4.4

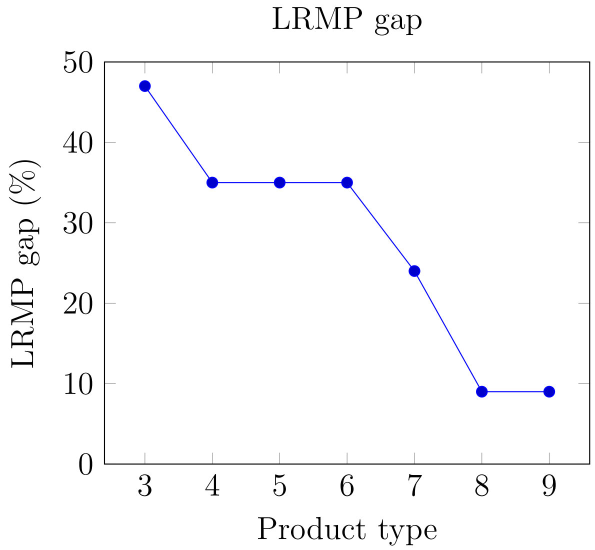

Figure 2.9: Variation of $\mathcal{P}$: LRMP solution quality

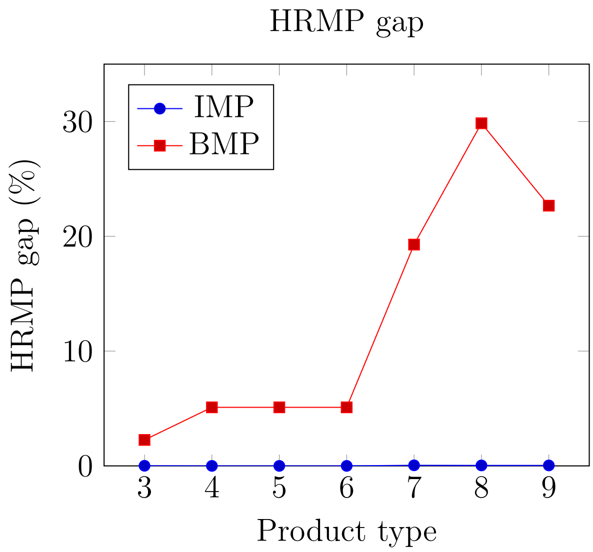

Figure 2.10: Variation of $\mathcal{P}$: HRMP gap

Table 2.12 Variation of $\mathcal{P}$: generated columns and solutions homogeneity.

Variation of $\mathcal{P}$							
Instances	$VP3$	$VP4$	$VP5$	$VP6$	$VP7$	$VP8$	$VP9$
IMP							
GENCOL	36	66	75	94	97	150	171
LRMPCOL	18	24	24	24	31	40	41
RMPCOL	20	27	28	26	39	49	59
BMP							
GENCOL	57	95	105	117	103	146	153
LRMPCOL	18	24	25	24	31	41	41
RMPCOL	2	1	1	1	1	1	2

columns are generated.

Table 2.12 displays information on homogeneity of the IMP and BMP solutions.

The number of columns generated by the IMP and BMP models follows a similar rising trend as the number of package types grows. This also explains the rising trend in the computing time.

This non systematic and unpredictable behavior suggests the opportunity of a deeper analysis in order to understand what makes a certain package type attractive. Table 2.13 displays the number of basic columns of each package type p in the optimal LRIMP solution.

It is possible to note that the most often used package types are 2, 4 and 8. In order to explain this observation we define two event type properties: the cross event type value and the cross event type actual capacity.

The **cross event type value** V_t for each event type $t \in \mathcal{T}$ is the overall

Table 2.13 Variation of $\mathcal{P}$: number of basic columns at optimality.

p	$VP3$	$VP4$	$VP5$	$VP6$	$VP7$	$VP8$	$VP9$	tot.
1	0	0	0	0	0	6	2	8
2	9	13	13	12	10	11	16	84
3	9	0	0	0	0	0	0	9
4	-	11	11	12	12	6	8	60
5	-	-	0	0	0	0	0	0
6	-	-	-	0	0	0	0	0
7	-	-	-	-	9	0	0	9
8	-	-	-	-	-	17	14	31
9	-	-	-	-	-	-	1	1

Table 2.14 Variation of $\mathcal{P}$: cross event type value and actual capacity.

t	1	2	3	4	5	6
C_t	714	297	910	223	279	924
V_t	339	144	445	78	126	382

value of all events in $\mathcal{E}_t$.

$$V_t = \sum_{e \in \mathcal{E}_t} v_e \ \forall t \in \mathcal{T}.$$

The **cross event type actual capacity** C_t for each event type $t \in \mathcal{T}$ is the overall actual capacity of events in $\mathcal{E}_t$.

$$C_t = \sum_{e \in \mathcal{E}_t} \min\{c_e, d_e\} \ \forall t \in \mathcal{T}.$$

As we can see from Table 2.14, the lowest values of C_t and V_t are those of event types 2, 4 and 5. These event types act like bottlenecks, because as soon as their capacity is exhausted no more packages can be produced. Therefore packages of a package type characterized by a composition pattern u_{pt} that includes few units of event types 2, 4 and 5 are easier to combine with other packages and hence they are likely to be part of the optimal LRMP solution. We refer to them as bottleneck event types and we indicated their subset with B.

Table 2.15 displays the composition pattern parameter of each package type $p \in \mathcal{P}$ (rows) and each event type $t \in \mathcal{T}$ (columns). For each package type $p \in \mathcal{P}$ we also indicate the overall number of required events from set B that we refer as β^B and the standard deviation of the u_{pt} values in B that we refer as ρ^B.

$$B = \{2, 4, 5\}$$

$$\beta^B = \sum_{t \in B} u_{pt}$$

Table 2.15 Variation of $\mathcal{P}$: composition parameters.

u_{pt}	1	**2**	3	**4**	**5**	6	β^B	ρ^B
1	0	**1**	1	**2**	**1**	1	4	0.5
2	1	**0**	2	**1**	**1**	1	2	0.5
3	2	**1**	0	**1**	**1**	1	3	0.0
4	2	**1**	1	**0**	**1**	1	2	0.5
5	1	**1**	0	**2**	**1**	1	4	0.5
6	0	**1**	1	**2**	**2**	0	5	0.5
7	1	**2**	2	**1**	**0**	0	3	0.8
8	2	**1**	2	**0**	**0**	1	1	0.5
9	2	**2**	1	**0**	**0**	1	2	0.9

$$\rho^B = \sigma_{t \in B} u_{pt}$$

Bottleneck event types correspond to bolded columns. These two indicators show very strong correlation with the presence of the corresponding package types in the optimal solution. From Table 2.13 we note that package types 2, 4 and 8 occur several times in the optimal LRMP solution and from Table 2.15 we also note that the same package types have the lowest values of β^B and ρ^B.

We can conclude that every time a more convenient package type is added to the instance, we observe an increase in the upper bound and the lower bound, in the number of basic columns in the LRMP optimal solution and in the computing time. The convenience of a package type can be reliably estimated by two indicators, intuitively meaning that a package type is convenient when it requires few events of bottleneck event types.

Number of event types Table 2.16 presents the results obtained from the variation of the number of event types in instance 50t6, which in origin is characterized by 6 events types. Every time a new event type is added or subtracted, a new column in the composition parameter u_{pt} is also added or subtracted. Furthermore the event set partition $\mathcal{E}_t$ changes as a new event type is added or subtracted, because the set $\mathcal{E}$ of events remains the same. The event set $\mathcal{E}$ is so partitioned in a different number of subsets. We call the modified instances $VT3, \ldots, VT9$, where the last figure represents the number of event types. Further instances with other event type variation are tested and their results are presented in Appendix 3.

The effect of changing the number of event types is even more significant than that of changing the number of package types. The addition of an event type implies a change in every product type composition parameter u_{pt} for all $p \in \mathcal{P}$. Moreover the addition of an event type provokes a decrease in the cross event type capacities.

From Table 2.11 we note that the addition of more event types initially implies a significant loss in the value of the upper bound, due to the cross

Table 2.16 Variation of $\mathcal{T}$: optimal values and computing time (sec.).

Variation of $\mathcal{T}$							
Instance	$VT3$	$VT4$	$VT5$	$VT6$	$VT7$	$VT8$	$VT9$
LRMP o.f.	100440	100440	88361	65078	51669	56697	69468
LRMP gap	0.00%	0.00%	12.03%	35.21%	48.56%	43.55%	30.84%
IMP							
HRMP o.f.	100440	100410	88354	65077	51669	56695	69468
HRMP gap	0.00%	0.03%	0.01%	0.00%	0.00%	0.00%	0.00%
LRMP time	2.2	2.3	2.6	2.0	1.5	2.1	1.6
HRMP time	2.2	2.5	2.6	2.1	1.5	2.1	1.7
BMP							
HRMP o.f.	72265	76274	64952	61757	50196	55210	61760
HRMP gap	28.05%	24.06%	26.49%	5.10%	2.85%	2.62%	11.10%
LRMP time	2.3	2.2	3.1	3.1	3.1	2.8	3.5
HRMP time	2.4	2.3	3.1	3.1	3.1	2.8	3.6

event capacity reduction. Instances $VT8$ and $VT9$ are characterized by a limited improvement of the upper bound, because partitioning the events in many subsets tends to produce more diverse and therefore easier to combine package types.

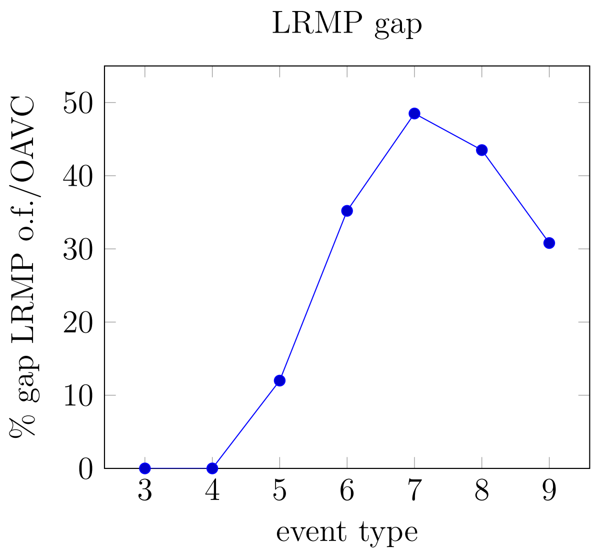

Figure 2.11: Variation of $\mathcal{T}$: LRMP gap

As with the standard instances the HRIMP model yields a very small gap with respect to the upper bound. With the HRBMP model the gap is definitely larger, it decreases as the number of event types grows (see Figure 2.12) because the heuristic solution improves. However this trend is not monotonic: instance $VP9$ has a worse heuristic solution than $VP8$.

The computing time of both BMP and IMP models does not seem to be correlated with the number of event types (see Figure 2.13). Since the

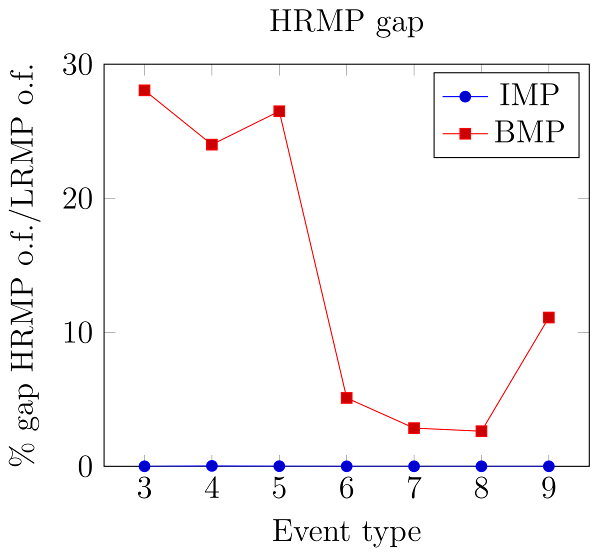

Figure 2.12: Variation of $\mathcal{T}$: HRMP gap

Table 2.17 Variation of $\mathcal{T}$: solutions homogeneity.

	$\mathcal{T}$ Variation						
Instance	$VT3$	$VT4$	$VT5$	$VT6$	$VT7$	$VT8$	$VT9$
				IMP			
GENCOL	93	100	109	75	59	74	68
LRMPCOL	50	50	39	24	19	21	25
RMPCOL	58	56	50	28	23	24	28
				BMP			
GENCOL	95	90	120	105	91	94	134
LRMPCOL	50	50	39	25	19	21	24
RMPCOL	1	1	1	1	1	1	1

number of calls to the pricing algorithm depends on the number of package types but not on the number of event types, the change of $|\mathcal{T}|$ does not have any significant impact on the computing time.

Table 2.17 described the homogeneity of the solutions provided by the IMP and BMP models.

From Figure 2.13 we note that in the IMP model the number of columns generated follows the same trend as the IMP computing time. In the BMP model this correlation is not so clear.

The number of columns in the LRIMP, LRBMP and HRIMP optimal solutions follows a decreasing trend even if for instance $VT9$ it present a rising trend. This is probably due to a different configuration of event set partition: a particularly convenient partition of event type can produce a capacity increase and therefore more columns are generated (see Figure 2.14 and Figure 2.15).

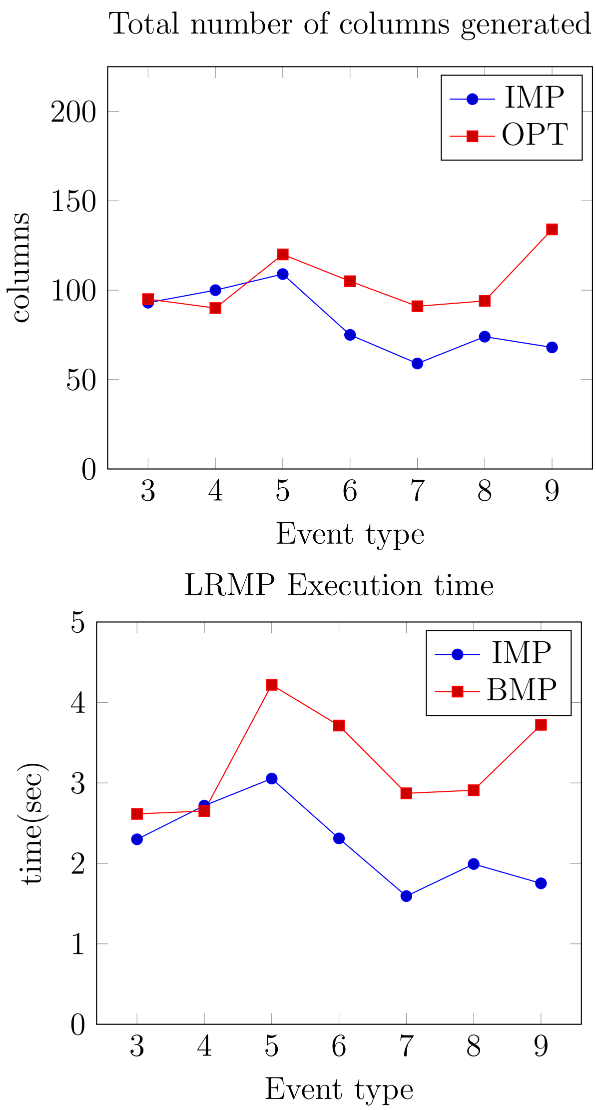

Figure 2.13: Variation of $\mathcal{T}$: Columns generated and computing time.

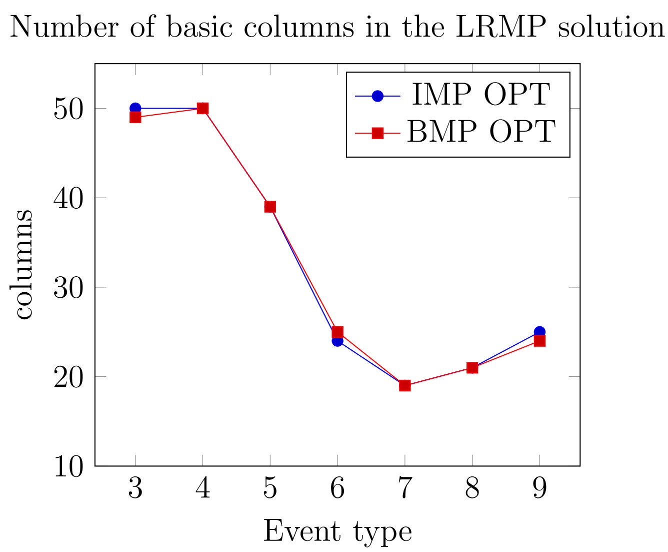

Figure 2.14: Variation of $\mathcal{T}$: LRMP optimal columns

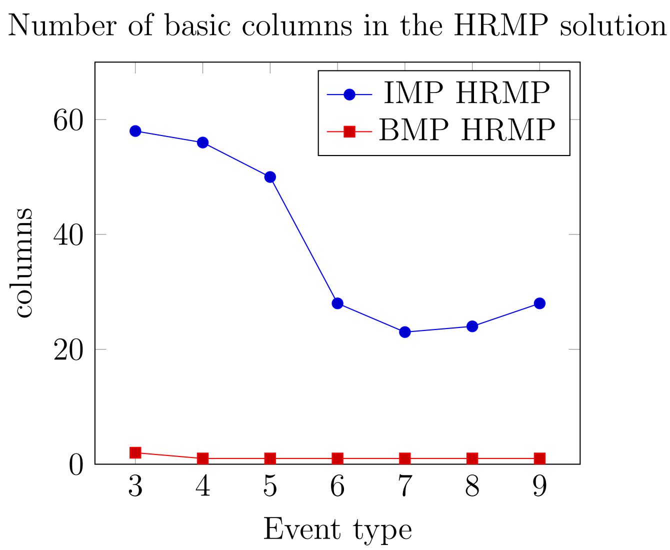

Figure 2.15: Variation of $\mathcal{T}$: HRMP basic columns

Table 2.18 Variation of $\mathcal{T}$: package types distribution IMP solution

$\mathcal{T}$ Variation							
Instance	$VT3$	$VT4$	$VT5$	$VT6$	$VT7$	$VT8$	$VT9$
p=1	46%	24%	6%	0%	0%	0%	22%
p=2	15%	51%	54%	54%	49%	49%	26%
p=3	15%	3%	0%	0%	46%	48%	2%
p=4	2%	6%	40%	46%	5%	3%	10%
p=5	21%	17%	0%	0%	0%	0%	40%

Table 2.19 Variation of $\mathcal{E}$: optimal values and computing time (sec.).

Variation of $\mathcal{E}$					
Instance	$VE40$	$VE50$	$VE60$	$VE70$	$VE80$
OACV	81780	100440	118966	141443	166722
LRMP o.f.	92769	65078	79322	102095	134151
LRMP gap	23%	35%	33%	28%	20%
IMP					
HRMP o.f.	62769	65077	79322	102093	134143
HRMP gap	0.00%	0.00%	0.00%	0.00%	0.01%
LRMP time	2.7	2.0	2.9	3.9	4.9
HRMP time	2.8	2.1	2.9	4.5	5.1
BMP					
HRMP o.f.	57921	61757	73234	91757	112826
HRMP gap	7.72%	5.10%	7.68%	10.13%	15.90%
LRMP time	3.1	3.1	4.4	7.4	8.6
HRMP time	3.1	3.1	4.4	7.4	8.7

This results from the variation in the composition pattern:when more convenient package types are generated, owing to the change in the partition of the events, and the cross event capacity decreases, owing to the smaller size of the subsets $\mathcal{E}_t$, the optimal solutions tend to be made by columns of few package types. (see Table 2.18).

Number of events In Table 2.19 we present the results obtained with instance 50t6 when the number of events is changed. Every change is made by inserting ten additional events in $\mathcal{E}$, starting with 40 up to 80, as indicated in the name of the instances. Additional events are randomly assigned to event types. The composition parameter u_{pt} does not change. Since the OACV increases with the number of events, the values of the upper and lower bounds also do.

In Figure 2.16 we can note that the optimal value of the LRMP shows an evident growth with the number of events, because the increase in capacity yields larger cross event capacity. The heuristic lower bounds provided by the HRIMP model are consistently higher than those provided by the HRBMP model.

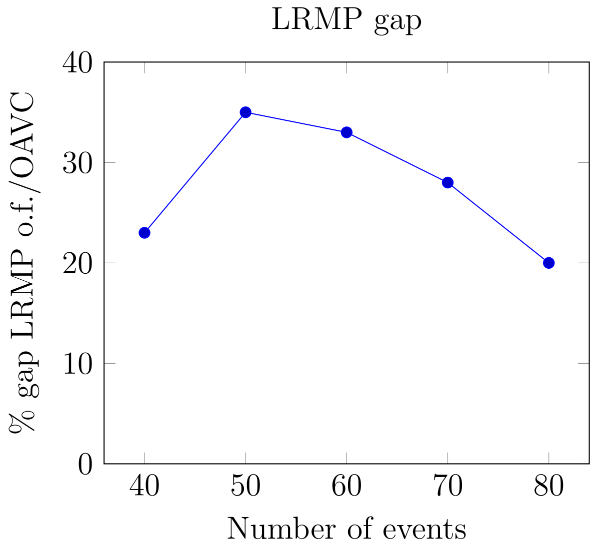

Figure 2.16: Variation of $\mathcal{E}$: LRMP gap.

Figure 2.17 shows that the solution of the HRMP models improve, as expected, as the number of events grows, but the primal-dual gap also grows larger: larger capacity implies that the columns generated by the HRBMP solution can accommodate more events, but the improvement in the objective function is not as large as the improvement in the OACV.

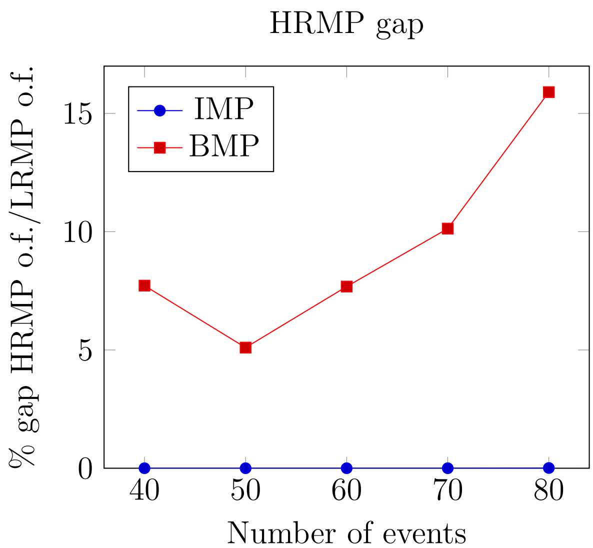

Figure 2.17: Variation of $\mathcal{E}$: HRMP gap

The computing time of column generation, with both models, shows an increase with the number of events, as expected, because more and more columns are generated due to the larger and larger cross event capacity.

Finally we analyze the relationship between the number of events and the

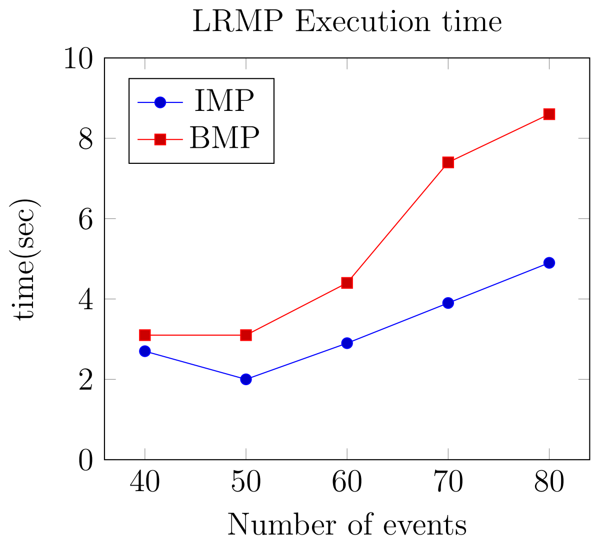

Figure 2.18: Variation of $\mathcal{E}$: computing time to solve LRMP.

Table 2.20 Variation of $\mathcal{E}$: optimal values and computing time (sec.).

Variation $\mathcal{E}$					
Instances	$VE40$	$VE50$	$VE60$	$VE70$	$VE80$
IMP					
GENCOL	83	75	118	142	205
LRMPCOL	24	24	29	40	52
RMPCOL	28	28	29	45	57
BMP					
GENCOL	103	105	145	218	279
LRMPCOL	24	25	28	39	52
RMPCOL	1	1	1	1	2

homogeneity of the solutions. Table 2.20 shows the results obtained with the IMP and the BMP model.

More columns are generated when more events can be combined together and this in turn explains the increase in computing time.

Figure 2.20 shows the number of columns in the optimal LRMP solution. Table 2.21 shows that these columns are very homogeneous, that is they are made of packages of few different types.

2.5.3 Final Comments

The tests reported above show that when significant parameters of the instances are changed the HRIMP model keeps providing better solutions than the HRBMP model. The optimal solution of the LRBMP model is consistently more homogeneous.There is an evident correlation between the number

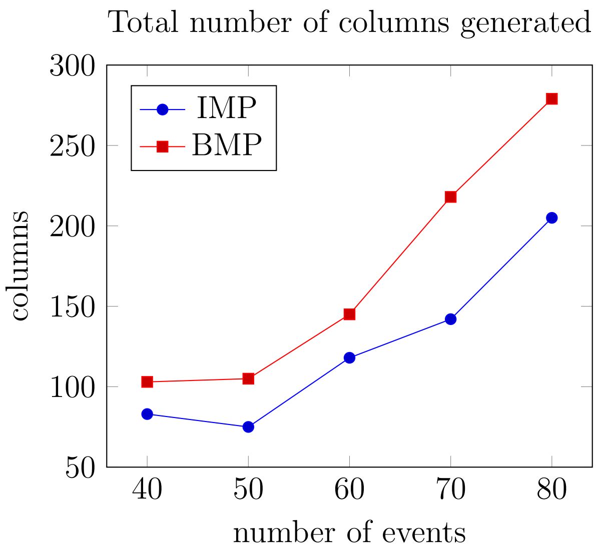

Figure 2.19: Variation of $\mathcal{E}$: Generated Package tours

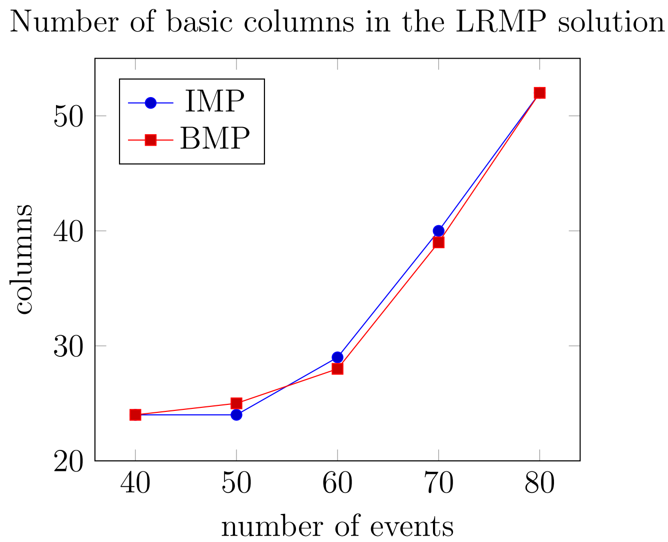

Figure 2.20: Variation of $\mathcal{E}$: LRMP performance

Table 2.21 Variation of $\mathcal{E}$: package types distribution IMP solution

Variation$\mathcal{E}$						
Instance	E40	E50	E60	E70	E80	E90
p=1	-	-	46%	45%	55%	54%
p=2	64%	54%	-	-	-	-
p=3	5%	-	-	-	-	13%
p=4	31%	46%	54%	55%	45%	32%
p=5	-	-	-	-	-	1%

of columns generated and the computing time.

The parametric analysis shows that the composition parameter u_{pt} and the partitions $\mathcal{E}_t$ have a high impact on the solutions. We have identified cross event type capacity as an indicator that allows identifying event types acting as bottlenecks.

The BMP model is less robust to capacity variations, because it takes into account capacity constraints not only in the master problem but also in the pricing problem.In the parametric BMP model the capacity constraint is eliminated from the pricing problem; as a result, the HRBMP solutions achieve a smaller HRMP gap for small paramertic values.

Chapter 3

Extensions

The proposed extensions approach several aspects of Tour Package Composition Problem: demand segmentation, inconvenience cost considerations, groups of multiple sizes and stochastic demand. In each extension we will use the most suitable formulation: integer master (IMP) or binary master (BMP). In the first section the group model that gives solutions with tour packages characterized by a certain dimension. In the second section we present the IMP version and the BMP version of the Inconvenience cost problem. In the third section we present the Demand Segmentation Model in which segmented demand expressed preferences towards the whole offer of packages tour. In the fourth section we present a model that solves a stochastic demand problem. In the last section we compare the models and results presented in the Chapter. Each section is divided into the following sub-sections: motivation, problem formulation, column generation formulation, data set description, results and comments.

3.1 A model for tourist groups

3.1.1 Motivation

The classical Network Revenue Management problem[9, 49] and the Single Capacity Problem [50, 51]. In these models for each individual booking request a decision is taken whether to accept it or not. On the contrary in our setting it is very important to consider group requests, because package tours can be sold to tourist groups willing to stay together. The size of the groups can also be determined by other factors such as the capacity of the vehicles carrying the tourists around.

From the results presented so far it appears that the HRIMP model yields better solutions than the HRBMP model. The primal dual gap is lower than 0.05%, while it ranges from 4% to 33% with the HRBMP model. However, since HRIMP solutions are less homogeneous, they show a high diversity in the number of tourists assigned to the same packages as Table 3.1 shows.

Table 3.1 HRIMP: Distribution of tourists into different packages.

Instance	25t3	25t6	50t3	50t6	100t3	100t6
Avg. n. of tourists per package	18.56	9.71	16.90	9.96	17.14	9.15
Standard deviation	10.76	8.46	13.23	7.13	14.20	8.93

In this section we propose a model that generates packages of a dimension multiple of a given integer parameter. We consider this parameter to be the capacity of the vehicle used by the groups of tourists. For this purpose we elaborate a BMP model and a IMP model. In the BMP formulation the constraints that limit the package size to be an integer multiple of the group size are in the pricing problem. We indicate this model with BMPMUL. In the IMP formulation the constraints that limit the package size to be an integer multiple of the group size are in the master problem of the HRMP formulation . We indicate this model with IMPMUL.

3.1.2 Problem formulation

We indicate with r the size of the groups. We impose that the number of tourists assigned to package q and visiting event e is an integer multiple of r. An integer variable a_{eq} indicates the number of vehicles needed to serve tourists visiting event e within a package q.

The model is non-linear with integer and binary variables.

$$
\begin{aligned}
\text{maximize } z = & \sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{Q}_p}\sum_{e\in\mathcal{E}} v_e x_{eq} g_q && && (3.1)\\
\text{s.t. } & \sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{Q}_p} x_{eq} g_q \le \min\{d_e, c_e\} && \forall e\in\mathcal{E} && (3.2)\\
& \sum_{e\in\mathcal{E}_t} x_{eq} = u_{pt} && \forall p\in\mathcal{P}\ \forall q\in\mathcal{Q}_p\ \forall t\in\mathcal{T} && (3.3)\\
& x_{eq} g_q = a_{eq} r && \forall p\in\mathcal{P}\ \forall q\in\mathcal{Q}_p\ \forall e\in\mathcal{E} && (3.4)\\
& g_q \text{ integer} && \forall p\in\mathcal{P}\ \forall q\in\mathcal{Q}_p && (3.5)\\
& x_{eq} \text{ binary} && \forall e\in\mathcal{E}\ \forall q\in\mathcal{Q}_p && (3.6)
\end{aligned}
$$

3.1.3 Column generation

Binary master group model (BMPMUL) In the column generation model we need to introduce an integer variable n_q representing the number of tourists with each package tour q and a non-repetition constraint that forbids any event to be assigned more than once to any package. We dynamically generate the elements of $\mathcal{Q}_p$ for each package type $p \in \mathcal{P}$.

Master problem:

$$
\begin{aligned}
\text{maximize } z = & \sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{Q}_p} w_q h_q && && (3.7)\\
\text{s.t. } & \sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{Q}_p} y_{eq} h_q \le \min\{d_e, c_e\} && \forall e\in\mathcal{E} && (3.8)\\
& 0 \le h_q \le 1 && \forall p\in\mathcal{P}\ \forall q\in\mathcal{Q}_p && (3.9)\\
& && && (3.10)
\end{aligned}
$$

where w_q is the value of all events that belong to package $q \in \mathcal{Q}$, that is $w_q = \sum_{e\in\mathcal{E}} v_e y_{eq}$. The master problem is linear because y_{eq} are data, not variables. We indicate with λ_e the dual variable of each constraint (3.8).

Pricing problem:

The pricing sub-problem is decomposed into a problem for each package type

$p \in \mathcal{P}$.

$$\text{maximize } \overline{w}_p = \sum_{e \in E} (v_e - \lambda_e) y_e. \tag{3.11}$$

$$\text{s.t.} \sum_{e \in \mathcal{E}_t} y_e = u_{pt} n_p \qquad \forall t \in \mathcal{T} \tag{3.12}$$

$$y_e \leq \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{3.13}$$

$$y_e = r a_e \qquad \forall e \in \mathcal{E} \tag{3.14}$$

$$y_e <= n_p \qquad \forall e \in \mathcal{E} \tag{3.15}$$

$$y_e \geq 0 \text{ integer} \qquad \forall e \in \mathcal{E} \tag{3.16}$$

$$n_p \geq 0 \text{ integer} \qquad \forall p \in \mathcal{P}. \tag{3.17}$$

Integer master group model (IMPMUL) In Chapter 2 section 2.2.1 the IMP formulation that corresponds to the IMPMUL LRMP formulation is presented. The HRMP formulation uses the columns generated by the LRMP formulation.

We consider an integer linear programming problem whose columns are those that have been generated while solving model (2.12)-(2.14) with column generation. We denote as $\tilde{\mathcal{Q}}_p$ this set of columns for each package type $p \in \mathcal{P}$. The model we solve is the following.

$$\text{maximize } z = \sum_{p \in \mathcal{P}} \sum_{q \in \tilde{\mathcal{Q}}_p} w_q g_q \tag{3.18}$$

$$\text{s.t.} \sum_{p \in \mathcal{P}} \sum_{q \in \tilde{\mathcal{Q}}_p} x_{eq} g_q \leq \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{3.19}$$

$$g_q = r\ a_q \qquad \forall p \in \mathcal{P}\ q \in \tilde{\mathcal{Q}}_p \tag{3.20}$$

$$g_q \text{ integer} \qquad \forall p \in \mathcal{P}\ \forall q \in \tilde{\mathcal{Q}}_p. \tag{3.21}$$

$$a_q \text{ integer} \qquad \forall p \in \mathcal{P}\ \forall q \in \tilde{\mathcal{Q}}_p. \tag{3.22}$$

3.1.4 Data-set and results

The model was tested with five data-sets generated from instance 25t6 and 5 u_pt variations and for the following group sizes: 5, 10, 15, 20, 25, 35, 50. The actual capacity event average is 66.94, its standard deviation is 13.57, its minimum value is 40 and its maximum value is 98.

In Table (3.2) and (3.3) we present the computational results.

Table 3.2 BMPMUL computational results.

Binary master group model (BMPMUL)								
r	1	5	10	15	20	25	35	50
LRMP o.f.	43308.7	43308.7	43308.7	43308.7	43308.7	43308.7	41971.3	35111.4
LRMP gap	6%	6%	6%	6%	6%	6%	9%	24%
HRMP o.f.	32400	31062	29358	29607	26930	26128	24327	24740
HRMP gap	25%	28%	32%	31%	38%	40%	42%	29%
LRMP time (sec.)	2.3	2.3	2.3	2.2	2.2	2.0	2.4	1.9
HRMP time (sec.)	2.4	2.3	2.3	2.2	2.2	2.1	2.3	1.9

Table 3.3 IMPMUL computational results.

Integer master group model (IMPMUL)								
r	1	5	10	15	20	25	35	50
LRMP o.f.	42265.5	42265.5	42265.5	42265.5	42265.5	42265.5	42265.5	42265.5
LRMP gap	8%6	8%	8%	8%	8%	8%	8%	8%
HRMP o.f.	42045	41246	38464	36531	34444	32165	24969	13860
HRMP gap	1%	2%	9%	14%	18%	24%	41%	67%
LRMP time (sec.)	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
HRMP time (sec.)	1.2	1.2	1.3	1.3	1.3	1.3	1.2	1.2

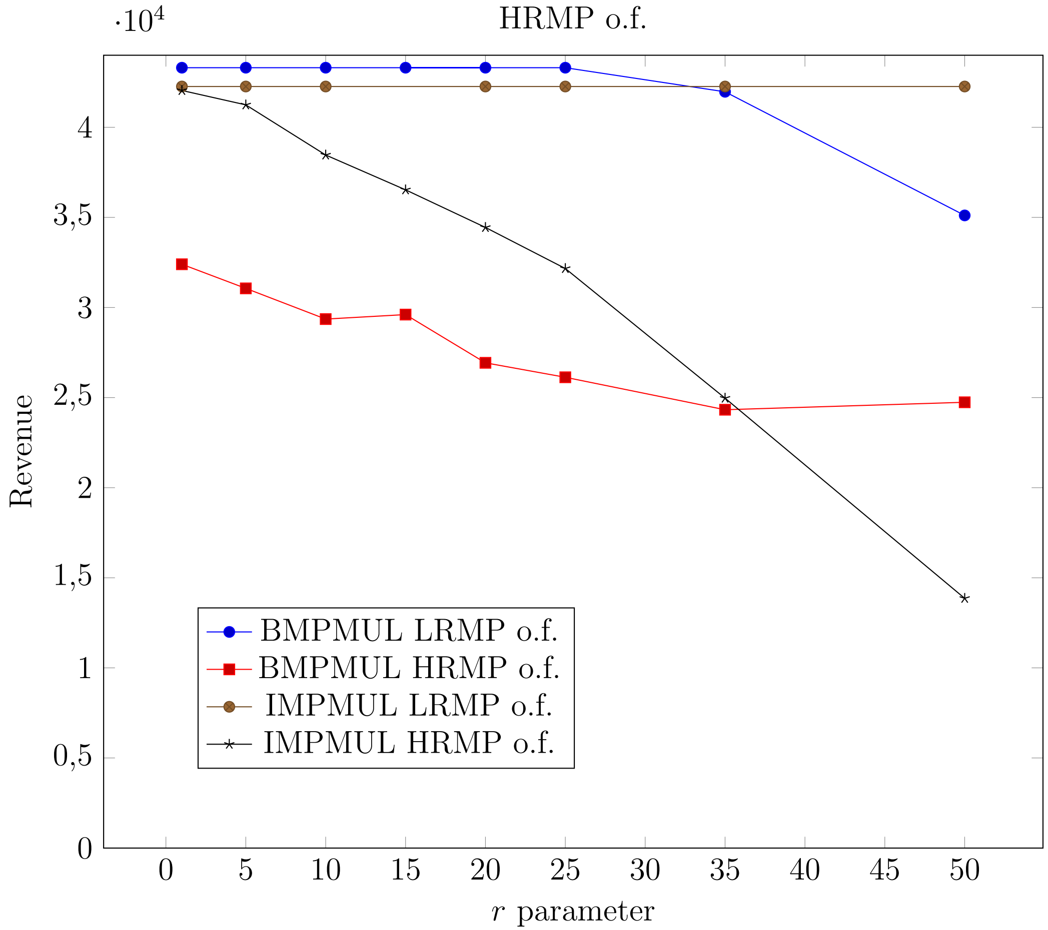

Figure 3.1: IMPMUL and BMPMUL revenue

3.1.5 Comments

For group size values that are smaller than minimum event capacity IMPMUL provides higher HRMP o.f. values than BMPMUL. For larger values of group size BMPMUL achieves higher HRMP o.f. values. In addition IMP model provides solutions that are easier to be applied in practice. The BMPMUL solution provides a *bunch* of packages combined together; therefore the group dimension r does not characterize a group of tourists that visit the same subset of events. The IMPMUL solution can guarantee that.

3.2 A model with demand segmentation

3.2.1 Motivation

In Revenue Management problems and especially in the Single Resource Capacity Problem and Network RM problem, demand is segmented according to consumers' buying capabilities. To consider this feature in our problem, we define segments of demand characterized by the number of customers willing to buy a package tour at a certain maximum price.In this model we still have demand and price linked to resources (i.e. events). Nevertheless we introduce segments characterized by a total demand and a correspondent maximum reservation price. This is to say that packages can be sold only to customers in a segment whose reservation price is not less than the value of the package. Hence we generate packages that comply not only with the constraints considered so far but also with maximum price constraints depending on the segment. The IMP model will be used for column generation. We indicate this model with IMPSEG. It is not possible to propose a BMP model because as it generates columns that represents a "bunch" of packages combined together it is not possible to separate them in order to impose the constraint associated with the package price limitations. The model respect a not-nested capacity rule discussed in paragraph 1.2.1 where each demand class accesses just it correspondent segment capacity.

3.2.2 Problem formulation

The main differences in the formulation are listed below.

Data. Instances are defined with these additional data:

- a set π of segments;
- a demand s_k associated with each segment $k \in \pi$;
- a reservation price p_k associated with each segment $k \in \pi$;

Variables

- the composition of packages is indicated by a binary assignment variable x_{eq}, that takes value 1 if and only if event $e \in \mathcal{E}$ is included in package $q \in \mathcal{Q}_{kp}$, where $\mathcal{Q}_{kp}$ is the set of packages of type $p \in \mathcal{P}$ and value compatible with segments up to $k \in \pi$ and $\mathcal{Q} = \bigcup_{p \in \mathcal{P}, k \in \pi} \mathcal{Q}_{kp}$;
- the number of packages $q \in \mathcal{Q}_{kp}$ of each segment $k \in \pi$ and each type $p \in \mathcal{P}$ is indicated by the integer variable g_q.

Constraints The constraints are formulated as follows.

- Capacity constraint.

$$\sum_{k \in \pi} \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}_{kp}} x_{eq} g_q \leq c_e \quad \forall e \in \mathcal{E}$$

- Demand constraint.

$$\sum_{k \in \pi} \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}_{kp}} x_{eq} g_q \leq d_e \quad \forall e \in \mathcal{E}$$

- Package composition constraint.

$$\sum_{e \in \mathcal{E}_t} x_{eq} = u_{pt} \quad \forall k \in \pi \ \forall p \in \mathcal{P}, \ \forall q \in \mathcal{Q}_{kp}, \ \forall t \in \mathcal{T}$$

- Segment demand constraint: the amount of packages assigned to each segment $k \in \pi$ cannot be higher than segment demand s_k.

$$\sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}_{kp}} g_q \leq s_k \quad \forall k \in \pi$$

- Reservation price constraint. The value of each package assigned to each segment $k \in \pi$ is constrained to be not larger than p_k and not smaller than p_{k-1}.

$$p_{k-1} \leq \sum_{e \in \mathcal{E}} x_{eq} v_e \leq p_k \quad \forall k \in \pi \ \forall p \in \mathcal{P} \ \forall q \in \mathcal{Q}_{kp}$$

Objective. The objective is the same as in the previous models.

$$z = \sum_{k \in \pi} \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}_{kp}} \sum_{e \in \mathcal{E}} v_e x_{eq} g_q.$$

3.2.3 Column generation

Restricted linear master problem (LRIMP). We insert the demand segmentation in the IMP formulation. The resulting model is indicated by IMPSEG.

Master problem:

$$\text{maximize } z = \sum_{k\in\pi}\sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{Q}_{kp}} w_q g_q \tag{3.23}$$

$$\text{s.t.} \sum_{k\in\pi}\sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{Q}_{kp}} x_{eq} g_q \leq \min\{d_e, c_e\} \quad \forall e \in \mathcal{E} \tag{3.24}$$

$$\sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{Q}_{kp}} g_q \leq s_k \quad \forall k \in \pi \tag{3.25}$$

$$g_q \geq 0 \quad \forall k \in \pi\ \forall p \in \mathcal{P}\ \forall q \in \mathcal{Q}_{kp} \tag{3.26}$$

where w_q is the value of each package $q \in \mathcal{Q}_{kp}$, that is $w_q = \sum_{e\in\mathcal{E}} v_e x_{eq}$. We indicate with λ_e and μ_k the two non-negative dual vectors of constraints (3.24) and (3.25) of the LRIMP, where the integrality conditions of variables g_q have been relaxed into non-negativity conditions $g_q \geq 0$.

Pricing problem:

The pricing sub-problem is decomposed for every package $p \in \mathcal{P}$ and for every segment $k \in \pi$.

$$\text{maximize } \overline{w}_{kp} = \sum_{e\in\mathcal{E}} v_e x_e - \sum_{e\in\mathcal{E}} \lambda_e x_e - \mu_k \tag{3.27}$$

$$\text{s.t.} \sum_{e\in\mathcal{E}_t} x_e = u_{pt} \quad \forall t \in \mathcal{T} \tag{3.28}$$

$$p_{k-1} \leq \sum_{e\in\mathcal{E}} v_e x_e \leq p_k \tag{3.29}$$

$$x_e \text{ binary} \quad \forall e \in \mathcal{E}. \tag{3.30}$$

The objective function can be rewritten as:

$$\text{maximize } \overline{w}_{kp} = \sum_{e\in\mathcal{E}} (v_e - \lambda_e) x_e - \mu_k$$

.

Heuristic Restricted Master Problem (HRIMP). A heuristic solution is computed in the same way as shown in the previous chapter. We solve a restricted integer master problem without relaxing the integrality constraints and considering only the columns generated by the column generation algorithm.

Table 3.4 Model with demand segmentation: computational results.

Instance	25t3	25t6	50t3	50t6	100t3	100t6
Package types	3	6	3	6	3	6
Events	25	25	50	50	100	100
Event types	5	5	5	5	5	5
OACV	46142	46142	100440	100440	205708	205708
LRMP o.f.	46142	32964	100440	59859	205708	154889
LRMP gap	0%	28.56%	0%	40.40%	0%	24.70%
HRMP o.f.	46142	32964	100440	59859	205697	154808
HRMP gap	0%	0%	0%	0%	0.01%	0.05%
time LRMP	1.9	1.3	2.9	2.6	4.7	4.5
time HRMP	1.9	1.4	2.9	2.6	8.5	4.6

3.2.4 Data-set and results

The data-set consists of instances 25t3, 25t6, 50t3, 50t6, 100t3 and 100t6. The only difference is that segment parameters s_k and p_k were added. In particular we have considered three demand segments, corresponding to low, medium and high purchasing power.

In order to generate feasible instances we estimated segment reservation prices p and segment demands s that are consistent with event demands d, event values v and package type composition patterns u.

We generated segment demand s as follows. We first estimated a total demand

$$D = \frac{\sum_{e\in\mathcal{E}} d_e}{\sum_{p\in\mathcal{P}} \sum_{t\in\mathcal{T}} u_{pt}/|\mathcal{P}|}.$$

Then we assigned this total demand D to each segment using the following percentages: 50% to the low segment, 30% to the medium segment and 20% to the high segment.

We generated segment reservation price values p as follows. For each event type $t \in \mathcal{T}$ we sort the values v_e for all $e \in \mathcal{E}_t$. Then for each $p \in \mathcal{P}$ we sum the u_{pt} largest and smallest values in order to compose a maximum and a minimum virtual value for package type p. We define $MAXW$ and $MINW$ as the maximum among the maximum virtual values and the minimum among the minimum virtual values. Finally we define the p values for the three segments as 40%, 60% and 85% of $MAXW$ and not lower than $MINW$.

In Table 3.4 we display the computational results.

Looking at Figure 3.2 we can note that the gap between the LRMP optimal value and the overall actual capacity value (OACV) is null for the instances with 3 package types and this occurs for both IMPSEG and IMP. However for instances with 6 package types we observe a gap: the results of

IMP are better than those of IMPSEG. This was expected because IMPSEG is a more constrained problem due to the introduction of the additional constraints. In addition, observing Table 3.4 we note that the gap between the LRMP and HRMP values, HRMP gap, are almost null.

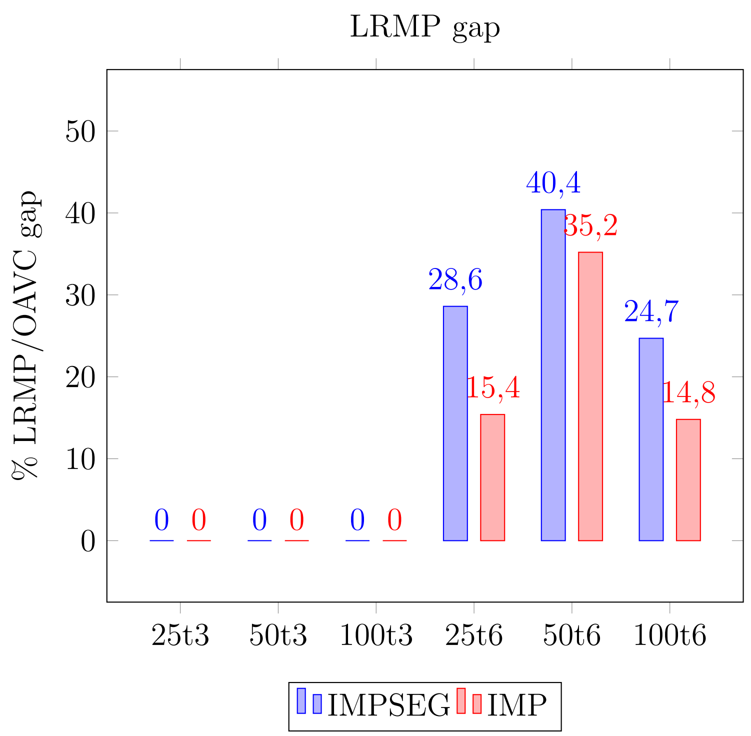

Figure 3.2: Model with demand segmentation: LRMP solution quality

In Figure 3.3 we display the computing time of IMPSEG and IMP. The two models show a similar trend.

The maximum demand s that can be allocated to each demand segment - high, medium and low - is a problem parameter. It is interesting to compare the distribution of the offered packages in the solutions with the distribution of the demand in the segments. This comparison is displayed in Table 3.5.

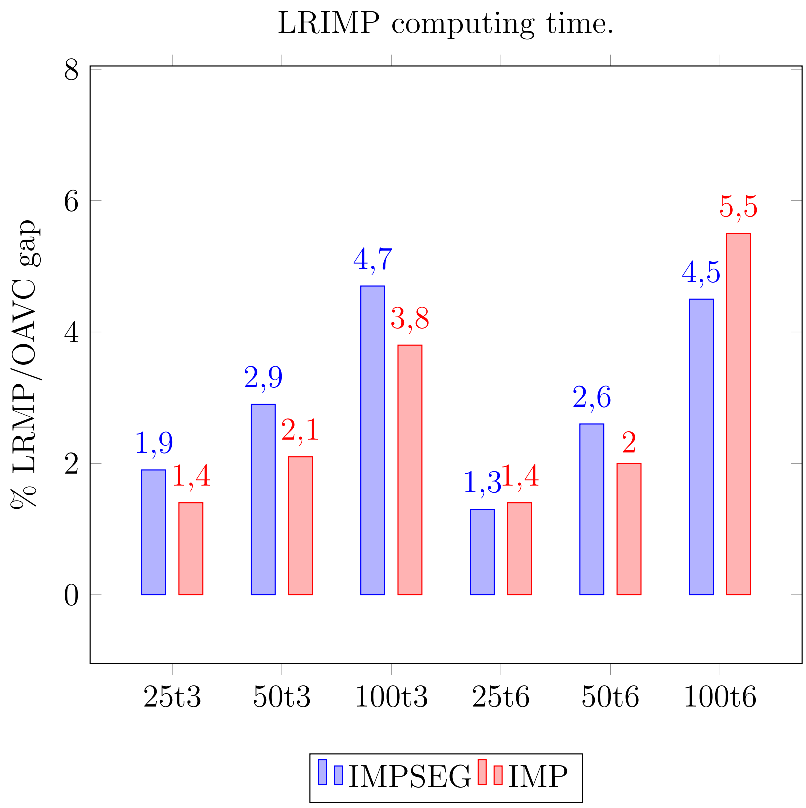

Figure 3.3: Model with demand segmentation: computing time.

Table 3.5 Model with demand segmentation: parameters and results

Demand parameters and results								
Instance	low parameter	low result	medium parameter	medium result	high parameter	high result	total parameter	total result
25t3	239	239	143	143	96	96	478	478
25t6	136	69	81	81	54	54	271	204
50t3	710	439	426	338	284	186	1420	964
50t6	402	-	241	118	161	161	804	279
100t3	985	985	591	591	394	394	1970	1.970
100t6	558	294	335	335	223	223	1116	852

As we can see the model tends to produce more packages in the medium and high segments, because it prefers the use of packages with higher value. Instance 50t6 has a particular configuration of events, where a shortage in some event types capacity limits the total capacity usage and explains the poor quality of the solution. In instance 50t6 the IMPSEG model solutions computed by column generation and by the heuristic model do not include any package of the low segment.

3.2.5 Comments

The introduction of demand segments does not change the nature of the problem and it is easily managed by the models here proposed and by the column generation algorithm we have used. Very good heuristic solutions can still be obtained from the HRMP model. The gap between the primal and the dual bounds is never larger than 0.05%.

As expected the IMPSEG model tends to allocate tourists to higher classes, since they are more convenient.

3.3 Model with inconvenience cost

3.3.1 Motivation

The main motivation for studying the PTC problem is to optimize the distribution of potential tourism demand on a territory adjacent to classical tourism centers, in order to exploit the available tourism facilities in an optimal way and to maximize revenue. However a primary disadvantage to overcome is the geographical distance that implies an inconvenience for the tourists who decide to visit the considered territory, since they have to spend money and time to travel there. For this reason we extended our models in order to take into account the inconvenience cost in the objective function.

3.3.2 Problem formulation

The only difference from the PTC problem described in the previous chapter is the addition of an inconvenience cost f in the objective function. This parameter represents the cost in which tourists incur by choosing to move to the territory under study. This cost can be related to the transportation cost of the transfer. The meaning of subtracting it from the objective function is to discount the inconvenience cost from the total value of the offered package.

We consider both modeling options for column generation: a model with integer master variables (IMPINC) and a model with binary master variables

(BMPINC). In both cases the inconvenience cost affects the computation of the reduced cost of the columns in the pricing sub-problem.

3.3.3 The IMP model with inconvenience cost

In the IMPINC model each column corresponds to a single package. We dynamically generate the elements of sets $\mathcal{Q}_p$ for every package type $p \in \mathcal{P}$.

Linear Restricted Master problem:

$$\text{maximize } z = \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}_p} w_q g_q \tag{3.31}$$

$$\text{s.t. } \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}_p} x_{eq} g_q \leq \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{3.32}$$

$$g_q \geq 0 \qquad \forall p \in \mathcal{P}\ \forall q \in \mathcal{Q}_p \tag{3.33}$$

where w_q is the value of each package $q \in \mathcal{Q}$ discounted by the inconvenience cost, that is $w_q = \sum_{e \in \mathcal{E}} v_e x_{eq} - f$. We define λ_e as the dual variables of constraints (3.32). Integrality conditions of variables g are relaxed into non-negativity conditions.

Pricing problem: The pricing sub-problem is decomposed for every package type $p \in \mathcal{P}$:

$$\text{maximize } \overline{w}_p = w_q - \sum_{e \in \mathcal{E}} \lambda_e x_e \tag{3.34}$$

$$\text{s.t. } \sum_{e \in \mathcal{E}_t} x_e = u_{pt} \qquad \forall t \in \mathcal{T} \tag{3.35}$$

$$x_e \text{ binary} \qquad \forall e \in \mathcal{E}. \tag{3.36}$$

The objective function can be written as

$$\text{maximize } \overline{w}_p = \sum_{e \in \mathcal{E}} (v_e - \lambda_e) x_e - f.$$

The pricing problem is a discrete optimization problem that can be further decomposed in as many sub-problems as the number of event types $t \in \mathcal{T}$. For each event type, indeed, the sub-problem is optimally solved by selecting the u_{pt} events with maximum reduced value $(v_e - \lambda_e)$. We note that the addition of the parameter f in the objective function of the pricing problem does not change its combinatorial structure and does not affect its computational complexity.

3.3.4 The BMP model with inconvenience cost

In the BMPINC model each column corresponds to a bunch of packages of the same type $p \in \mathcal{P}$. We dynamically generate the elements of sets $\mathcal{Q}_p$ for every package type $p \in \mathcal{P}$.

Linear Restricted Master problem:

$$\text{maximize } z = \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}_p} w_q h_q \tag{3.37}$$

$$\text{s.t. } \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}_p} y_{eq} h_q \leq \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{3.38}$$

$$0 \leq h_q \leq 1 \qquad \forall p \in \mathcal{P}\ \forall q \in \mathcal{Q}_p \tag{3.39}$$

where w_q is the value of each package $q \in \mathcal{Q}_p$, that is $w_q = \sum_{e \in \mathcal{E}} v_e y_{eq} - n_p f$. In this model the inconvenience value is multiplied by the number of packages in the bunch. We define μ_e as the dual variables of constraints (3.38). The binary conditions on variables h are relaxed into $0 \geq h_q \geq 1$.

Pricing problem:

The pricing sub-problem is decomposed for every package type $p \in \mathcal{P}$:

$$\text{maximize } \overline{w}_p = w_p - \sum_{e \in \mathcal{E}} \mu_e y_e \tag{3.40}$$

$$\text{s.t. } \sum_{e \in \mathcal{E}_t} y_e = u_{pt} n_p \qquad \forall t \in \mathcal{T} \tag{3.41}$$

$$y_e \leq \min\{d_e, c_e\} \qquad \forall e \in \mathcal{E} \tag{3.42}$$

$$y_e \geq 0 \text{ integer} \qquad \forall e \in \mathcal{E} \tag{3.43}$$

$$n_p \geq 0 \text{ integer} \tag{3.44}$$

$$\tag{3.45}$$

The objective function can be rewritten as

$$\text{maximize } \overline{w}_p = \sum_{e \in \mathcal{E}} (v_e - \mu_e) y_e - n_p f.$$

3.3.5 Instances and results

The model was tested with a data-set obtained from instance 50t6 for the following inconvenience costs: 10, 25, 50, 75, 100. We can compare these values with the average value of the events and with the average value of the packages. In instance 50t6 the average event value is 30 and the standard

Table 3.6 Models with inconvenience cost: computational results.

Models with inconvenience cost						
f	0	10	25	50	75	100
OACV	100440	100440	100440	100440	100440	100440
LRMP o.f.	65078	62288	58103	51128	44153	37178
LRMP gap	35%	38%	42%	49%	56%	63%
IMPINC						
HRMP o.f.	65077	62284	58101	51123	44151	37178
HRMP gap	0.00%	0.01%	0.00%	0.01%	0.00%	0.00%
time LRMP (sec.)	1.9	2.0	2.1	2.4	2.0	1.9
time HRMP (sec.)	1.9	2.1	2.2	2.4	2.0	2.0
BMPINC						
HRMP o.f.	61757	58967	54782	47807	40832	33857
HRMP gap	5%	5%	6%	6%	8%	9%
time LRMP (sec.)	2.7	2.5	2.8	2.9	2.8	2.8
time HRMP (sec.)	2.7	2.5	2.8	2.9	2.9	2.8

deviation of the event value is 12.26. The minimum event value is 10 and the maximum event value is 50. The average event package value is not a given datum but a problem output. The average package value in IMP model results for instance 50t6 is 233. Therefore the inconvenience cost varies according to event's value range rather than the package value range.

In Table 3.6 we present the results of the IMPINC and the BMPINC models for instance 50t6 with different inconvenience costs.

LRMP solution quality. From Figure 3.4 we can note that the gap between the LRMP objective function and the OACV grows as the inconvenience cost grows. This is intuitive because the inconvenience cost is subtracted from the objective function but it is not taken into account in the estimate of the OACV.

Figure 3.5 refers to the IMPINC model and displays the number of generated columns (GENCOL), the number of columns that belong to the LRMP solution (LMPCOL) and the number of columns that belong to the HRMP solution (RMPCOL). We can see that the general structure of both solutions does not change, in spite of the revenue reduction.

HRMP solution quality By comparing the gap between the HRMP objective function and the LRMP objective function of the IMPINC and BMPINC models. As displayed in Figure 3.6 the HRMP gap in the IMP model is almost null as in other integer master formulation. In the BMPINC model this gap increases as the inconvenience cost grows.

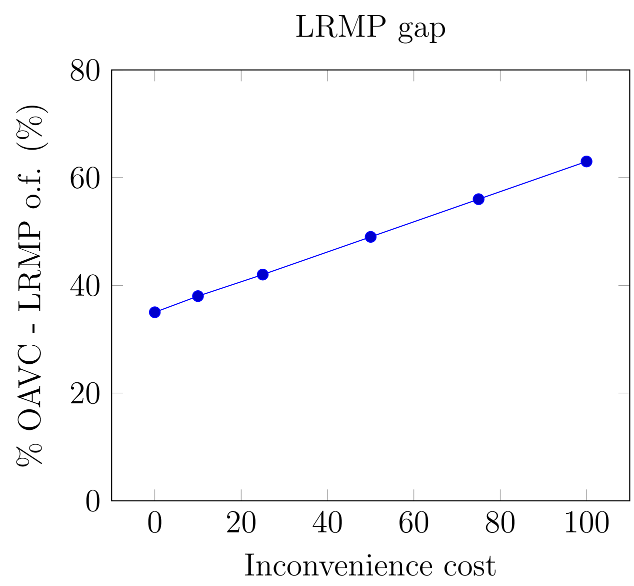

Figure 3.4: Model with Inconvenience cost: LRMP solution quality

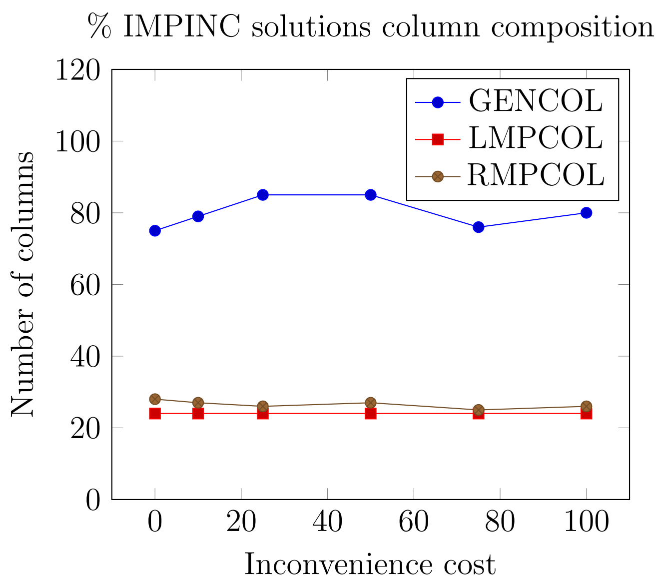

Figure 3.5: Model with inconvenience cost: package tour composition.

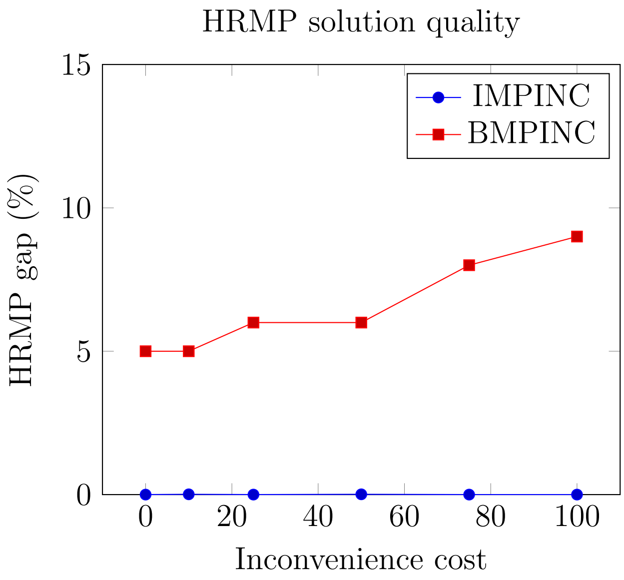

Figure 3.6: Model with inconvenience cost: heuristic solution quality.

Computing time. Concerning the computing time, we can observe from Figure 3.7 that the IMPINC shows a better performance. We remark that in both the IMPINC and the BMPINC models the computing time of the LRMP and of the HRMP formulation do not present a substantial difference.

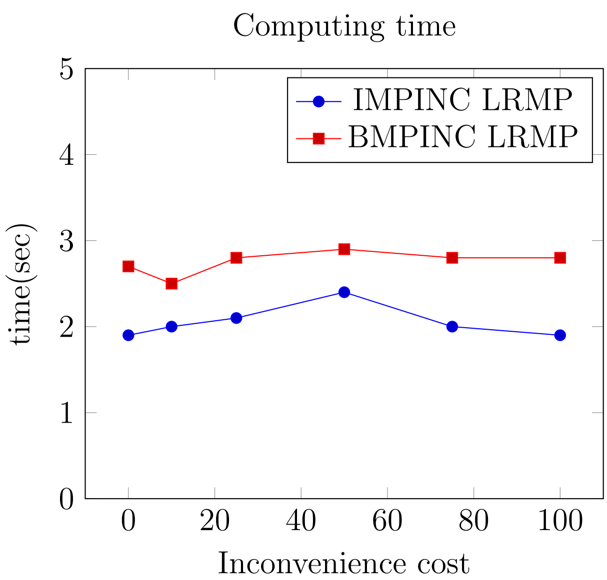

Figure 3.7: Model with inconvenience cost: computing time.

3.3.6 Comments

The main consequence of the introduction of the inconvenience cost is the decrease in the objective function value. The number of columns used in the LRMP and HRMP suggests that the general structure of both solutions does not change. By comparing the binary master inconvenience model and the integer master inconvenience model it is possible to note that the IMPINC model shows, as expected, a better HRMP solution quality and requires shorter computing time.

3.4 Stochastic demand model

3.4.1 Motivation

In this extension we incorporate demand uncertainty into our model and we propose a Stochastic Programming (SP) model. In SP the decision-making process with uncertainty is organized in several subsequent steps. An initial solution is computed without an exact knowledge of the demand, but relying on a known probability distribution. Following a two-stage recourse approach we introduce explicitly corrective actions to be taken when demand becomes known. These corrective actions, called recourse actions in SP, imply penalties in the objective function and the goal is to minimize the total expected cost. The advantage of this method is that risk is considered explicitly. The disadvantage is that the model may be too large to solve and may require a suitable decomposition technique. We give a description of the "two-stage" recourse approach technique in Appendix 2. Hereafter we refer to the stochastic demand model with integer variables as the SIMP model.

3.4.2 Formulation

Our model consists in an innovative approach regarding two-stage stochastic models. In classical approach variables are partitioned in first stage and second stage variables. In our case the first and second stage decisions are discriminated by the columns that belong to each scenario. We generated a set of package tours and their quantity for each scenario and in the objective function these columns are weighted by the probability to be sold.

In addition to the data considered so far we also introduce the following data:

- a set S of scenarios ordered from the worst one to the best one i.e. by increasing demand;
- the probability P_s for each scenario $s \in S$.

- $\hat{P}_s = \sum_{r \in R | r \geq s} P_r$, a cumulated probability of scenarios not worst than s such that $d_{es'} \leq d_{es''} \forall e \in \mathcal{E}, \forall s' \leq s''$.
- a maximum demand d_{es} for each event $e \in \mathcal{E}$ for each scenario $s \in S$ (this replaces the demand vector d with one component for each event);

Variables. Besides the variables already used so far, we also need the following variable:

- an integer non-negative variable g_q^s for each event $q \in \mathcal{Q}_p^s$ and for each scenario $s \in S$.

Constraints. Capacity constraints and package composition constraints.

- Capacity constraint

$$\sum_{p \in \mathcal{P}} \sum_{r \in R | r \leq s} \sum_{q \in \mathcal{Q}_p^r} x_{eq} g_q^r \leq \min(c_e, d_{es}) \quad \forall e \in \mathcal{E} \ \forall s \in S.$$

- Package composition constraint

$$\sum_{e \in \mathcal{E}_t} x_{eq} = u_{pt} \quad \forall p \in \mathcal{P}, \ \forall q \in \mathcal{Q}_p^s, \ \forall t \in \mathcal{T}.$$

Objective The objective is to maximize the overall value obtained from the events.

$$z = \sum_{p \in \mathcal{P}} \sum_{s \in S} \sum_{q \in \mathcal{Q}_p^s} \hat{P}_s w_q^s g_q^s$$

3.4.3 Column generation

Formulation. We dynamically generate the elements of set $\mathcal{Q}_p^s$ for every package type $p \in \mathcal{P}$ and for every scenario $s \in S$.

Master problem:

$$\text{maximize } z = \sum_{p \in \mathcal{P}} \sum_{s \in S} \sum_{q \in \mathcal{Q}_p^s} \hat{P}_s w_q g_q^s \tag{3.46}$$

$$\text{s.t.} \sum_{p \in \mathcal{P}} \sum_{r \in R | r \leq s} \sum_{q \in \mathcal{Q}_p^r} x_{eq} g_q^r \leq \min(c_e d_{es}) \qquad \forall e \in \mathcal{E} \ s \in S. \tag{3.47}$$

$$g_q \text{ integer} \qquad \forall q \in \mathcal{Q}_p^s \ \forall s \in S \tag{3.48}$$

We relax the integrality restrictions (3.48) to obtain the linear relaxation of the master problem. We indicate by λ_{es} the non-negative dual variables of constraints 3.48.

Pricing sub-problem.

The pricing sub-problem is decomposed for every package type $p \in \mathcal{P}$ and $s \in S$.

$$\text{maximize } \overline{w}_{ps} \tag{3.49}$$

$$\text{s.t.} \sum_{e \in \mathcal{E}_t} x_e = u_{pt} \qquad \forall t \in \mathcal{T} \tag{3.50}$$

$$x_e \text{ binary} \qquad \forall e \in \mathcal{E}. \tag{3.51}$$

$$\text{where } \overline{w}_{ps} = \sum_{e \in \mathcal{E}} v_e x_e \hat{P}_s - \sum_{e \in \mathcal{E}} \lambda_e x_e \ \forall p \in \mathcal{P} \ \forall s \in S.$$

3.4.4 Data-sets and results

The stochastic data-sets are similar to the standard ones. The parameters to be added are the probability of each scenario P_s and the scenarios' demand. The medium scenario demand equals to the demand d_e of the standard instances. The low and high scenario demand are respectively 75% and 160% of the medium scenario demand. We present the results for standard data-sets with the addition of two versions of P_s: version A and version B. The values of P_s for both versions are displayed in Table 3.7.

Table 3.7 Stochastic model: P_s

Dataset version	A	B
low	0.1	0.6
medium	0.3	0.3
high	0.6	0.1

Stochastic IMP results. In Table 3.8 and 3.9 we display the results obtained with the stochastic model.

Table 3.8 Stochastic model: results with version A.

Stochastic IMP version A						
Instance	25t3SA	25t6SA	50t3SA	50t6SA	100t3SA	100t6SA
Package types	5	5	5	5	5	5
Events	25	25	50	50	100	100
Event types	3	6	3	6	3	6
OCV	54229	54229	115324	115324	235261	235261
LRMP o.f.	48073	38432	103573	64979	212009	179654
% LRMP o.f. - OCV	11%	29%	10%	44%	10%	24%
LRMP time (sec.)	3.3	4.0	6.3	5.4	10.3	15.2
HRMP o.f.	48069	38432	103569	64971	212006	179644
% OCV - HRMP	11%	29%	10%	44%	10%	24%
HRMP time (sec.)	3.5	4.0	6.4	5.4	10.4	24.2
Low demand HRMP o.f.	3746	32312	78328	55691	164100	135239
Medium demand HRMP o.f.	43180	37347	93671	63746	193568	160311
High demand HRMP o.f.	48073	38432	103573	64979	212009	179654

Table 3.9 Stochastic model: results with version B.

Stochastic IMP version B						
Instance	25t3SB	25t6SB	50t3SB	50t6SB	100t3SB	100t6SB
Package types	5	5	5	5	5	5
Events	25	25	50	50	100	100
Event types	3	6	3	6	3	6
OCV	54229	54229	115324	115324	235261	235261
LRMP o.f.	41868	36175	90847	61023	186853	158247
% LRMP o.f. - OCV	23%	33%	21%	47%	21%	33%
LRMP time (sec.)	4.3	6.1	7.5	7.2	10.5	14.2
HRMP o.f.	41868	36175	90847	61020	186843	158234
% HRMP o.f. - OCV gap	23%	33%	21%	47%	21%	33%
HRMP time (sec.)	4.5	6.2	7.6	7.3	10.6	20.2
Low demand	39329	34982	86191	58884	175804	148916
Medium demand	41687	36175	90669	60850	185971	157790
High demand	41868	36175	90847	61023	186853	158247

Remarkably in all our tests we could find a heuristic value (HRMP) coinciding with the upper bound value (LRMP); this provides an a posteriori optimality guarantee to the solutions of the HRMP model.

In the previous models analysis we compared the objective function with the overall actual capacity value (OACV) that takes into account not only the events capacities but also the events demand. In the stochastic model, demand is subject to scenario variation and therefore OACV is no longer a suitable term of comparison. Hence we use the overall capacity value (OCV) that does not depend on demand and is defined as

$$OCV = \sum_{e \in \mathcal{E}} v_e c_e.$$

We define as the OCV gap as the percentage gap between the OCV and the optimal function value.

We observe in Figures 3.8 and 3.9 that the LRMP objective function is always incremented when a higher scenario is added, particularly with Version A probability configuration, characterized by a probability predominance of the high demand scenario.

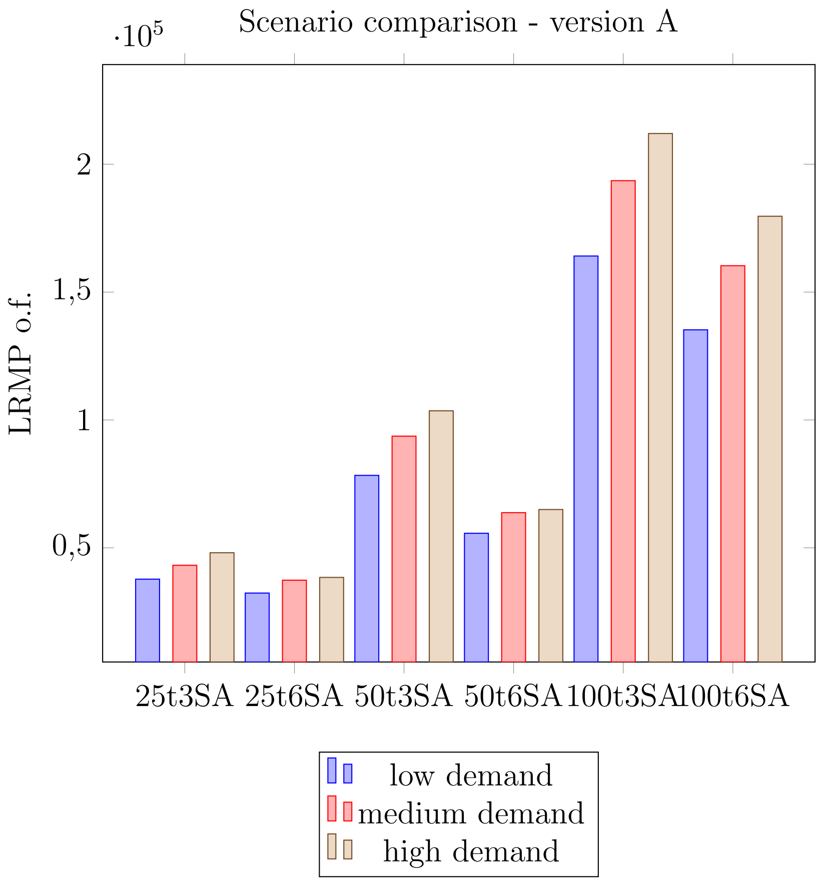

Figure 3.8: Stocastic Model: LRMP- A scenarios

As expected the P_s version A, that is characterized by a superior high

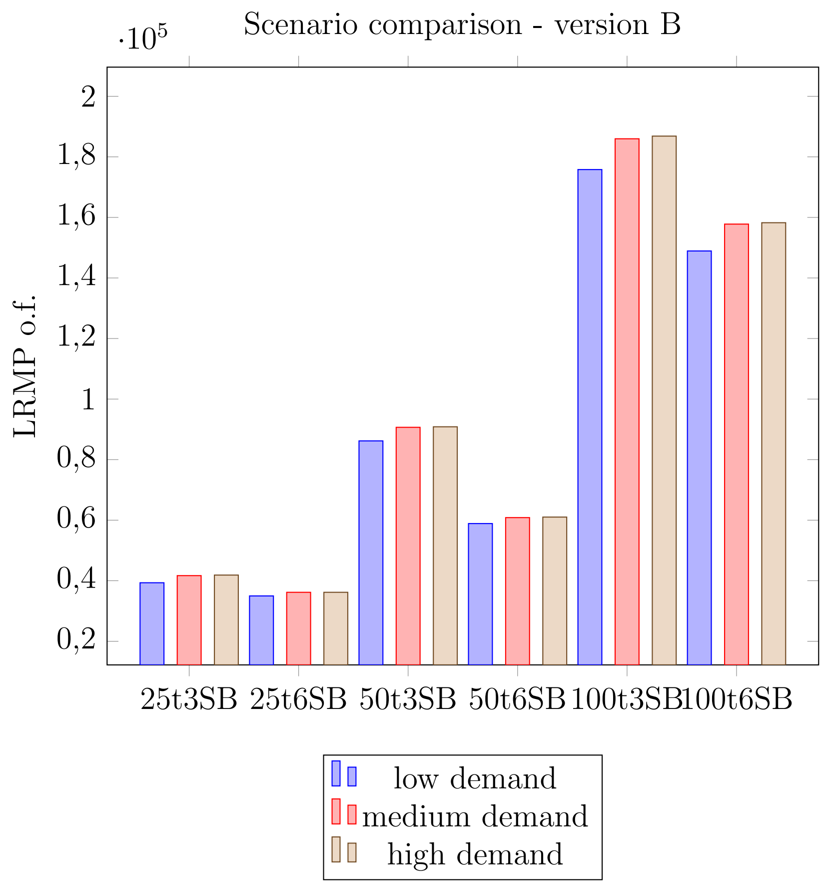

Figure 3.9: Stochastic model: LRMP-B scenarios

scenario demand probability achieves higher LRMP values with respect to version B instances as displayed in Figure 3.10.

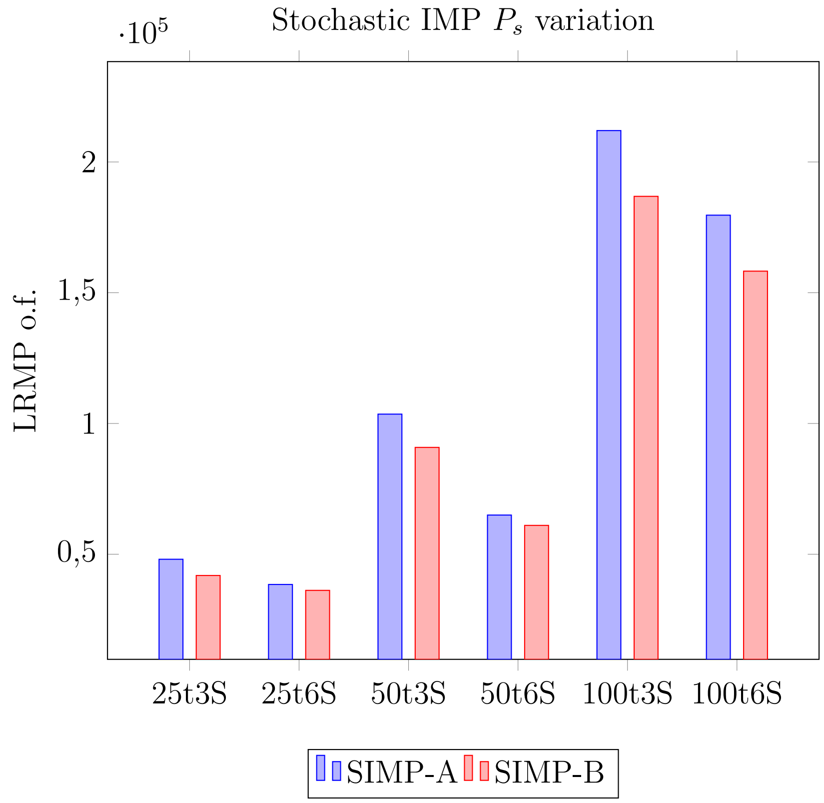

Figure 3.10: Stochastic model: P_s variation

Scenario probability variation. We use instance 25t6S to test further scenario probability dispersion as displayed in Table 3.10. A higher probability variation implies lower LRMP objective function values.

Table 3.10 Stochastic model: probability variation

P_s variation				
versions	X	Y	Z	W
P_{low}	0.3	0.2	0.1	0
P_{medium}	0.4	0.6	0.8	1
P_{high}	0.3	0.2	0.1	0
OCV	54229	54229	54229	54229
LRMP o.f.	37173	37678	38286	39036
% LRMP o.f. - OCV	31%	31%	28%	28%
LRMP time (sec.)	4.1	4.1	3.8	3.8
HRMP o.f.	37168	37676	38285	39034
% HRMP o.f. - OCV gap	31%	31%	28%	28%
HRMP time (sec.)	4.2	4.1	3.9	3.8

Demand variation. We use instance 25t6S to test different ratio between low and high scenarios' demand and medium scenario demand. The results are displayed in Table 3.11. A higher demand variation implies a lower LRMP objective function values.

Table 3.11 Stochastic model: demand variation

Demand variation					
versions	J	H	K	L	M
% medium demand low scenario	0.9	0.8	0.7	0.6	0.5
% medium demand high scenario	1.1	1.2	1.3	1.4	1.5
OCV	54229	54229	54229	54229	54229
LRMP o.f.	38337	37548	38250	35791	34547
% LRMP o.f. - OCV	29%	31%	45%	34%	36%
LRMP time (sec.)	4.1	4.3	4.3	4.7	4.7
HRMP o.f.	38332	37544	38250	35791	34547
% HRMP o.f. - OCV	29%	31%	45%	34%	5%
HRMP time (sec.)	4.2	4.4	4.3	4.8	4.7

3.4.5 Comments

As expected, in SIMP model, as in IMP models, there is a small gap between the HRMP objective function and the LMRP objective function. As expected the variation of scenario probabilities and of demand imply a lower LRMP objective function values.

Chapter 4

Conclusions

In this chapter we underline the original features of our work and we propose future research topics on the PTC problem.

4.1 PTC formulations

In this thesis we have analyzed a new mathematical programming model in the context of destination management for tourism promotion especially suitable to enhance tourism activities in territories not reached by mass tourism and characterized by a large number of small accommodation facilities and points of interest.

In particular we have addressed the problem of determining the optimal composition of package tours. Existing literature deals with package tour composition mainly from a routing point of view, i.e. at a tactical level. On the contrary we addressed the selection problem at a strategic level.

Our model extends the existing scientific literature on products bundling because it considers the classification of products into several types, which allows to solve problem instances with a number of products much larger than those solved so far.

The proposed model of the PTCP differs from classical network revenue management problems mainly because demand is placed on resources (events), not on products (packages).

The problem we considered arises at a strategic level, not at an operational one, since it deals with optimally structuring the offer of products, not with real-time decisions on how to manage incoming customers' requests.

From the viewpoint of the solution method, we employed column generation to linearize a non-linear model with an exponential number of variables, not to decompose a linear model into subproblems as already done in the literature.

Integer and binary master formulations. The Linear Restricted Master Problem (LRMP) of both integer and binary formulations provide an upper bound for the PTC problem. We have studied the correlation between the LRMP objective function and the capacity utilization by a measure called Overall Actual Capacity Value. Furthermore we observed that the capacity utilization depends heavily on the data-set cross events capacities, i.e. the sum of the capacities of events of the same type.

We also defined a math-heuristic method that exploits the columns generated during the column generation algorithm and can provide primal feasible solutions of good quality (often optimal in our tests). The IMP formulation is clearly superior to the BMP as far as the heuristic solutions are concerned. This is easily explained by the observation that the BMP produces homogeneous solutions made of very few columns, while the IMP tends to provide heterogeneous solutions made by a larger number of different columns.

Extensions. The extensions illustrated in Chapter 3 incorporate particular features in our PTC model.

The model with groups is one of the main motivations for using the BMP formulation. Differently from the IMP formulation, capacity constraints must be added also to the pricing problem and the solution generated is a bunch of feasible packages grouped together. This implies that solutions turn out to be very homogenous in terms of package types.

The extension with demand segmentation, a typical feature of revenue management applications. The inclusion of this feature in the IMP formulation does not change the nature of the model; the results are still characterized by a small gap between the LRMP and the HRMP optimal values that is never larger than 0.05%. With this extended model it may be difficult to generate feasible instances; some indications are given to overcome this limitation.

The model with inconvenience costs is important for applications to territories that are likely to imply an additional travel for tourists, like - for example - the cities of Lombardy region different from Milan in the case of Expo2015.

The stochastic model motivation is to incorporate uncertainty in demand. We have analyzed and compared the composition of the optimal solution for several different combinations of scenario probabilities and demand level.

Future research. The work presented in this thesis opens the way for several possible developments. For instance:

- gathering all the proposed extensions in a single model simultaneously;
- running a discrete event simulation model to validate the results obtained from the stochastic model,
- considering more complex problems yielding more complex pricing problems,
- developing models to be applied at an operational decision level.

More complex problems may originate from larger degrees of freedom in the definition of feasible patterns and therefore in the formulation of composition constraints. The composition parameter u_{pt}, that states the number of events of each event type that is supposed to be in each package tour type, can be defined as a decision variable that can vary within a range (e.g. a pattern may require to include a number of historical points of interest not smaller than one and not larger than three). Moreover composition constraints may involve combinations of events of different types (e.g. a feasible pattern may require two churches and one museum or one church and two museums).

These extensions would give more flexibility to the decision-maker and would further enhance to optimize the capacity usage.

Finally it is possible to extend the PTCP with pricing policies, where the price of a package is not given by the sum of the prices of its events, but discounts and other dynamic pricing policies are employed, according to variations in demand forecasts in order to maximize the revenue.

Bibliography

[1] Assolombarda. http://www.expo2015.assolombarda.it/expo/expo - 2015 i numeri di expo 2015 -. 2014.

[2] J. F. Leimkuhler B. C. Smith and R. M. Darrow. Yield management at american airlines. *Interfaces*, 22 (1):8–31, 1992.

[3] P. E. Pfeifer. The airline discount fare allocation problem. *Decision Sciences*, 20(1):149–157, March 1989.

[4] R. C. Cross. *Revenue Management: hard-core tatics for market domination.* Broadway Books, 1540 Broadway, New York, NY 10036, 1998.

[5] L. R. Weatherford and Samule E. Bodily. A taxonomy and research overview of perishable-asset revenue management: yield management, overbooking and pricing. *Operations Research*, 40 (5):831–844, September-October 1992.

[6] J. I. McGill and Garrett J. Van Ryzin. Revenue management: research overview and prospects. *Transportation Science*, 33(2):233–252, May 1999.

[7] S. Brumelle and D.Walczak. Dynamic airline revenue management with multiple semi-markov demand. *Operations Research*, 51(1):137–148, January - February 2003.

[8] M. Modarres and M. Sharifyazdi. Revenue management approach to stochastic capacity allocation problem. *European Journal of Operations Research*, 192:442–459, 2009.

[9] D. Zhang. An improved dynamic programming decomposition approach for network revenue management. *Manufacturing Service Operations Management*, 13 (1):35–52, 2011.

[10] K. Pak and N. Piersma. Airline revenue management: an overview of OR techniques 1982-2001 . Technical report, Erasmus Research Institute of Management, January 2002.

[11] G. Bitran and R. Caldentey. An overview of pricing models for revenue management. *Manufacturing Service Operations Management*, 5 (3):203–229, Summer 2003.

[12] J.C.H. Chen W. Chiang and X. Xu. An overview of research on revenue management: current issues and future research. *Int. J. Revenue Management*, 1(1):97–128, 2007.

[13] K. T. Talluri and G. J. Van Ryzin. *The theory and practice of Revenue Management.* Springer Science+Business Media, Inc., 233 Spring Street, New York, NY 10013, USA, 2005.

[14] R. L. Phillips. *Pricing and Revenue Management Optimization.* Stanford University Press, Stanford California USA, 2005.

[15] G.H. Hardy, J.E. Littlewood, and G. Pòlya. Some theorems concerning monotonic functions. *Inequalities, Cambridge University Press, Cambridge, England*, pages 83–84, 1988.

[16] V.C.P. Chen, D. Gunther, and E.L. Johnson. A markov decision problem based approach to the airline YM problem. 1998.

[17] V.C.P. Che, D. Gunther, and E.L. Johnson. A bid price heuristics. 1999.

[18] Cooper W.L. Asymptotic behavior of an allocation policy for revenue management. *Operations Research*, 50:720–727, 2002.

[19] Ward Hanson and R. Kipp Martin. Optimal bundle pricing. *Management Science*, 36 (2):155–173, February 1990.

[20] B. Ghosh and S. Balachander. Competitive bundling and counterbundling with generalist and specialist firms. *Management Science*, 53 (1):159–168, 2007.

[21] C. Alejandro, J.C. Ferrer, and G. Bitràn. On pricing and composition of multiple bundles. *Congresso Latino-Iberoamericano di Investigacion Operativa*, pages 3052–3061.

[22] B. H. Kobayashi. does economics provide a reliable guide to regulating commodity bundling by firms? a survey of the economic literature. *Journal of Competition Law and Economics*, 1 (4):707–746, 2005.

[23] S. Stremersch and G.J. Tellis. Strategic bundling of products and prices: A new synthesis for marketing. *Journal of Marketing*, 66 (1):55–72, Jan 2002.

[24] T. Olderog and B. Skiera B. The benefits of bundling strategies. *Schmalenbach Business Review*, 52 (2):137–159, 2000.

[25] M. Banciu, E. Gal-Or, and P. Mirchandani. Bundling strategies when products are vertically differentiated and capacities are limited. *Management Science*, 56(12):2207–2223, December 2010.

[26] W.J. Adams and J. L. Yellen. Commodity bundling and the burden of monopoly. *The Quarterly Journal of Economics*, Vol(3):475–498, Aug 1976.

[27] R. Schmalensee. Gaussian demand and commodity bundling. *The Journal of Business,*, 57(1):S211 –S230, 1984.

[28] R. Venkatesh and V. Mahajan. A probabilistic approach to pricing a bundle of products or services. *Journal of Marketing Research,*, 30:494–508, Nov 1993.

[29] R. Venkatesh and W. Kamakura. Optimal bundling and pricing under a monopoly: Contrasting complements and substitutes from independently valued products. *The Journal of Business*, 76(2):211–231, April 2003.

[30] Y. Bakos and E. Brynjolfsson. Bundling information goods: Pricing, profits, and efficiency. *Management Science*, 45(12):1613–1630, December 1999.

[31] L.M. Hitt and P. Chen. Bundling with customer self-selection: A simple approach to bundling low-marginal-cost goods. *Management Science*, 51(10):1481–1493, October 2005.

[32] S. Wu, L.M. Hitt, P. Chen, and G. Anandalingam. Customized bundle pricing for information goods: A nonlinear mixed-integer programming approach. *Management Science*, 54(3):608–622, March 2008.

[33] G.R. Bitran and J.C. Ferrer. On pricing and composition of bundles. *Production and Operations Management*, 16(1):93–108, January-February 2007.

[34] R. Klein S. Mayer and S. Seiermann. A simulation-based approach to price optimisation of the mixed bundling problem with capacity constraints. *Int. J.ProductionEconomics*, 143:584–598, 2013.

[35] J.R.B. Ritchie and G.I. Crouch. The competitive destination: A sustainable perspective. *Tourism Management*, 22 (SI):1–7, 2000.

[36] M. Manente. Destination management and economic background: defining and monitoring local tourist destinations. *International Conference of Tourism Malaga*, 2008.

[37] T. Bieger. Reengineering destination marketing organizations - the case of switzerland. *The Tourist Review*, 3:4–17, 1998.

[38] A. Flagestad and C. A. Hope. Strategic success in winter sports destinations: A sustainable value creation perspective. *Tourism Management*, 22(5):445–461, 200.

[39] V. T. C. Middleton. *Principles of Economics Book. 4*, chapter The marketing and management of tourism destinations: Research directions for the next decade. In Springer [52], 1994.

[40] D.B. Weaver. A broad context model of destination scenarios. *Tourism Management*, 21(3):217–224, 2000.

[41] D. Buhalis. Marketing the competitive destination of the future. *Tourism Management*, 21 (SI):97–116, 2000.

[42] C. P. Cooper Ed., editor. *Advances in Case-Based Reasoning*, chapter Conceptual issues in the meaning of tourism. Progress in tourism, recreation and hospitality management, London: Belhaven Press, 1990.

[43] Marshall A. *Principles of Economics Book. 4*, chapter Industrial Organization, Continued. The Concentration of Specialized Industries in Particular Localities. In Springer [52], 1994.

[44] R. Sainaghi. From contents to processes: Versus a dynamic destination management model (ddmm). *Tourism Management*, 27:1053–1063, 2006.

[45] M. Sharples S. McCabe and C. Fosterc. Stakeholder engagement in the design of scenarios of technology-enhanced tourism services. *Tourism Management Perspectives*, 4:36–44, October 2012.

[46] D. Buhalis and R. Law. Progress in information technology and tourism management: 20 years on and 10 years after the internet the state of etourism research. *Tourism Management*, 29:609–623, 2008.

[47] B. Rodríguez, J. Molina, F. Pérez, and R. Caballero. Interactive design of personalised tourism routes. *Tourism Management*, 33:926–940, 2012.

[48] J.C. Neto, C.A. de Almeida, G. Roveroto, and J.D.H. Tavares. Applications of operations research techniques in optimization to visit tourist points of viña del mar. *Informatica Economica*, 16(2):5–13, 2012.

[49] T.C. Lee and M. Hersh. A model for dynamic airline seat inventory control with multiple seat bookings. *Transportation Science*, 27:252–265, August 1993.

[50] K. Littlewood. Forecasting and control of passenger bookings. *In Proceedings of the Twelfth Annual AGIFORS Symposium, Nathanya, Isreal*, 1972.

[51] S.L. Brumelle and J.I. McGill. Airline seat allocation with multiple nested fare classes. *Operations Research*, 41:127–137, 1993.

[52] Springer, editor. *Principles of Economics Book. 4*. P. Keller & T. Bieger Eds., St. Gallen: Publication of the AIEST., 1994.

[53] G.B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[54] I. Nowak. Relaxation and decomposition methods for mixed integer nonlinear programming. *Ph.D. Thesis Humboldt-Univrsitat zu Berlin*, October 2004.

[55] P.C. Gilmore and R.E. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9:pp. 849–859, 1961.

[56] P.C. Gilmore and R.E. Gomory. A linear programming approach to the cutting stock problem. *Operations Research - Part II*, 11:863–888, 1963.

[57] G. Desaulniers, J. Desrosiers, and M.M. Solomon. Accelerating strategies in column generation methods for vehicle routing and crew scheduling problems. *In Ribeiro, C.C. and Hansen, P., editors, Essays and Surveys in Metaheuristics, Boston. Kluwer*, pages 309–324.

[58] M.E. Lubbecke and J. Desrosiers. Selected topics in column generation. *Les Cahiers du GERAD*, G200264, 2002.

[59] Y. Agarwal, K. Mathur, and H.M. Salkin. A set-partitioning-based exact algorithm for the vehicle routing problem. *Networks*, 19:731–749, 1989.

[60] G. Desaulniers, J. Desrosiers, and M.M. Solomon. Accelerating strategies in column generation methods for vehicle routing and crew scheduling problems. *In C.C. Ribeiro and P. Hansen, editors, Essays and Surveys in Metaheuristics, Boston, 2001b. Kluwer.*

[61] G. Desaulniers, J. Desrosiers, and M.M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations research*, 40(2):342–354, 1992.

[62] Löbel A. Optimal vehicle scheduling in public transit. *PhD thesis, Technische Universit¨at Berlin*, 1997.

[63] Löbel A. Vehicle scheduling in public transit and lagrangean pricing. *Management Science*, 44(12):1637–1649, 1998.

[64] C.C. Ribeiro and F. Soumis. A column generation approach to the multiple-depot vehicle scheduling problem. *Operations Research*, 42(1):41–52, 1994.

[65] R. Borndorfer and A. Löbel. Scheduling duties by adaptive column generation. *ZIB-Report 01-02, Konrad-Zuse-Zentrum f¨ur Informationstechnik, Berlin*, 2001.

[66] M. Desrochers and F. Soumis F. A column generation approach to urban transit crew scheduling. *Transportation Science*, 23:1–13, 1989.

[67] J. Desrosiers, F. Soumis, and M. Desrochers. Routing with time windows by column generation. *Networks*, 14:545–565, 1984.

[68] S.O. Krumke, J. Rambau, and L.M. Torres. Real-time dispatching of guided and unguided automobile service units with soft time windows. *Proc. 10th Annual European Symposium on Algorithms, Rome, In R.H. Möhring and R. Raman, editors.*

[69] M.E. Lubbecke. Engine scheduling by column generation. *PhD thesis, Braunschweig University of Technology, Cuvillier Verlag, Gottingen*, 2001.

[70] M.E. Lubbecke and U.T. Zimmermann. Engine routing and scheduling at industrial inplant railroads. *Transportation Science*, 37 (2):pp. 183–197, 2003.

[71] M. Sol. Column generation techniques for pickup and delivery problems. *PhD thesis, Eindhoven University of Technology*, 1994.

[72] T.G. Crainic and J.M. Rousseau. The column generation principle and the airline crew pairing problem. *Infor*, 25:136–151, 1987.

[73] P.H. Vance, A. Atamtürk, C. Barnhart, E. Gelman, E.L. Johnson, Krishna A., D. Mahidhara, G.L. Nemhauser, and R. Rebello. A heuristic branch-and-price approach for the airline crew pairing problem. *Technical report*, June 1997.

[74] C. Barnhart and R.R. Schneur. Air network design for express shipment service. *Operations Research*, 44(6):852–863, 1996.

[75] A. Erdmann, A. Nolte, A. Noltemeier, and R. Schrader. Modeling and solving an airline schedule generation problem. *Annuals of Operations Research*, 107:117–142, 2001.

[76] C. Barnhart, N.L. Boland, L.W. Clarke, E.L. Johnson, G.L. Nemhauser, and Shenoi R.G. Flight string models for aircraft fleeting and routing. *Transportation Science*, 32(3):pp. 208–220, 1998.

[77] G. Desaulniers, J. Desrosiers, Y. Dumas, M.M. Solomon, and F. Soumis. Daily aircraft routing and scheduling. *Management Science*, 43(6):841–855, 1997.

[78] I. Ioachim, J. Desrosiers, F. Soumis, and N. Belanger. Fleet assignment and routing with schedule synchronization constraints. *European Journal of Operations Research*, 119(1):75–90, 1999.

[79] Y. Crama and A.G. Oerlemans. A column generation approach to job grouping for flexible manufacturing systems. *European Journal of Operations Research*, 78(1):58–80, 1994.

[80] K. Park, S. Kang, and S. Park. An integer programing approach to the bandwidth packing problem. *Management Science*, 42(9):1277–1291, 1996.

[81] C.C. Ribeiro, M. Minoux, and M.C. Penna. An optimal column-generation-with-ranking algorithm for very large scale set partitioning problems in traffic assignment. *European Journal of Operations Research*, 41:232–239, 1989.

[82] J.K. Sankaran. Column generation applied to linear programs in course registration. *European Journal of Operations Research*, 87(2):328–342, 1995.

[83] F. Vanderbeck. Decomposition and column generation for integer programs. *PhD thesis, Universite catholique de Louvain*, 1994.

[84] C.A.J. Hurkens, J.L. De Jong, and Z. Chen. A branch-and-price algorithm for solving the cutting strips problem. *Applied Mathematics Ser. B (Engl. Ed.)*, 12(2):215–224, 1997.

[85] J.M. Valerio de Carvalho. Exact solution of bin-packing problems using column generation and branch-and-bound. *Annual Operations Research*, 86:pp. 629–659, 1999.

[86] J.M. Valerio de Carvalho. Using extra dual cuts to accelerate convergence in column generation. *Technical report, Dept. Producao e Sistemas, Universidade do Minho, Portugal,*, 2000.

[87] P.H. Vance. Branch-and-price algorithms for the one-dimensional cutting stock problem. *Computation Optimization Application*, 9(3):211–228, 1998.

[88] P.H. Vance, C. Barnhart, E.L. Johnson, and G.L. Nemhauser. Solving binary cutting stock problems by column generation and branch-and-bound. *Computation Optimization Application*, 3 (2):pp. 111–130, 1994.

[89] F. Vanderbeck. Computational study of a column generation algorithm for bin packing and cutting stock problems. *Mathematical Programming*, 86(3):565–594, 1999.

[90] C. Barnhart, C.A. Hane, and P.H. Vance. Integer multicommodity flow problems. *Lecture Notes in Economics and Mathematical Systems*, 450:17–31, 1997.

[91] C. Barnhart, C.A. Hane, and P.H. Vance. Using branch-and-price-and-cut to solve origin-destination integer multicommodity network flow problems. *Operations Research*, 48(3):318–326, 2000.

[92] J.M. Bourjolly, G.Laporte, and H. Mercure. A combinatorial column generation algorithm for the maximum stable set problem. *Operations Research Letters*, 20(1):21–29, 1997.

[93] P. Hansen, B. Jaumard, and M. P. de Aragão. Mixed-integer column generation algorithms and the probabilistic maximum satisfiability problem. *European Journal of Operations Research*, 108:671–683, 1998.

[94] E.L. Johnson. Modelling and strong linear programs for mixed integer programming. *In S.W. Wallace, editor, Algorithms and Model Formulations in Mathematical Program- ming*, Berlin, Springer:1–43, 1989.

[95] A. Mehrotra and M.A.Trick. A column generation approach for graph coloring. *INFORMS Journal Computation.*, 8(4):pp. 344–354, 1996.

[96] M.W.P. Savelsbergh. A branch-and-price algorithm for the generalized assignment problem. *Operations Research*, 45(6):831–841, 1997.

[97] W.E. Wilhelm. A technical review of column generation in integer programming. *Optimization and Engineering*, 2:159–200, 2001.

[98] J. P. Arabeyre, J. Fearney, F. C. Steiger, and W. Teather. The airline crew scheduling problem: A survey. *Transportation Science*, 3:140–163, 1969.

[99] K. L. Hoffman and M. Padberg. Solving airline crew scheduling problems by branch-and-cut. *Management Science*, 39 (6):657–682, 1993.

[100] R. E. Marsten. An algorithm for large set partitioning problems. *Management Science*, 20, 1974.

[101] E. Balas and M. Padberg. Set partitioning: A survey. *SIAM Review*, 18:pp. 710–760, 1976.

[102] R.E. Marsten and F. Shepardson. Exact solution of crew scheduling problems using the set partitioning mode: Recent successful applications. *Networks*, 11:165–177, 1981.

[103] I. Gershkoff. Optimizing light crew schedules. *Interfaces*, 19:29–43, 1989.

[104] M. L. Fisher and P. Kedia. Optimal solution of set covering/partitioning problems using dual heuristics. *Management Science*, 36(6):674–688, 1990.

[105] D.M. Ryan. The solution of massive generalized set partitioning problems in air crew rostering. *Journal of the Operations Research Society*, 43:459–467, 1992.

[106] E. El-Darzi and G. Mitra. Graph theoretic relaxations of set covering and set partitioning problems. *European Journal of Operations Research*, 87(1):109–121, 1995.

[107] R. A. Rushmeier, K. L. Hoffman, and M. Padberg. Recent advances in exact optimization of airline scheduling problems. *Dept. of Operations Research and Operations Engineering, George Mason University, Working Paper*, 1995.

[108] D. Wedelin. An algorithm for large scale 0-1 integer programming with application to airline crew scheduling. *Annals of Operations Research*, 57:283–301.

[109] A. Atamturk, Nemhauser G. L., and M. W. P. Savelsbergh. A combined lagrangian, linear programming and implication heuristic for large-scale set partitioning problems. *Journal of Heuristics*, 1:247–259, 1969.

[110] H. D. Sherali and Y. Lee. Tighter representations for set partitioning problems. *Discrete Applied Mathematics*, 68:153–167, 1996.

[111] M. Confort, G. Cornuejols, and M. R. Rao. Decomposition of balanced matrices. *Journal of Combinatorial Theory B*, 2:292–406, 1999.

[112] M. Gamache, F. Soumis, G. Marquis, and J. Desrosiers. A column generation approach for large-scale aircrew rostering problems. *Operations Research*, 47(2):247–263, 1999.

[113] M. Conforti, G. Cornuejols, A. Kapoor, and K. Vuskovic. Balanced 0, 1 matrices, part i: Decomposition theorem. *Journal of Combinatorial Theory Series B*, 81(2):243274, 2001.

[114] M. Conforti, G. Cornuejols, A. Kapoor, and K. Vuskovic. Balanced 0, plus minus 1 matrices, part ii: Recognition algorithm. *Journal of Combinatorial Theory Series B*, 81(2):275–306, 2001.

[115] M. Conforti, G. Cornuejols, A. Kapoor, and K. Vuskovic. Perfect, ideal and balanced matrices. *European Journal of Operations Research*, 133(3):455–461, 2001.

[116] K. Fagerholt. Ship scheduling with soft time windows: An optimisation based approach. *European Journal of Operations Research*, 131(3):559–571, 2001.

[117] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali. Linear programming and network flows, john wiley & sons, new york. pages 320–350, 1990.

[118] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M.W. P. Savelsbergh, and P. H. Vance. Branch and price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[119] L. H. Appelgren. A column generation algorithm for a ship scheduling problem. *Transportation Science*, 3:53–68, 1969.

[120] L. H. Appelgren. Integer programming methods for a vessel scheduling problem. *Transportation Science*, 5:64–78, 1971.

[121] A. M. Geoffrion. Lagrangean relaxation for integer programming. *Mathematical Programming Study*, 2:82–114, 1974.

[122] G. L. Nemhauser and L. A. Wolsey. Integer and combinatorial optimization. *JohnWiley & Sons: New York, NY*, 1988.

[123] M.S. Bazaraa, J. J. Jarvis, and H. D. Sherali. Linear programming and network flows. *John Wiley & Sons, New York*, 320-350, 1990.

[124] J.L. Higle and S. Sen. An introductory tutorial on stochastic linear programming models. *Interfaces*, 29(2):33–61.

[125] A. Prekopa. Stochastic programming. *Kluwer Academic Publishers, The Netherlands*, 1995.

[126] J. Pinter. Stochastic modeling and optimization for environmental management. *Annals of Operations Research*, 31:527–544, 1991.

Appendix I

Column Generation

I.1 Motivation

Decomposition methods solve large scale problems by splitting them into several smaller subproblems that are coupled through a master problem. Usually, the master problem is a simple problem that can be solved in high dimensions, while the subproblems contain the complicated constraints. Decomposition of optimization problems started with the so-called Dantzig-Wolfe decomposition of linear programs with block-angular structure [53]. The method is linked to the dual simplex method, which is still one of the most efficient methods for solving linear programs . [54] This technique was first put to actual use by Gilmore and Gomory [55, 56] as part of an efficient heuristic algorithm for solving the cutting stock problem. Column generation is nowadays a prominent method to cope with a huge number of variables. The embedding of column generation techniques within a linear programming based branch-and-bound framework, introduced by Desrosiers, Soumis, and Desrochers [57] for solving a vehicle routing problem under time window constraints, was the key step in the design of exact algorithms for a large class of integer programs.[58]

The main reason for applying column generation to LP problems is their excessive number of columns (variables). LP problems may have a huge number of variables (for instance when variables have many indices). The computing time spent by the simplex algorithm usually depends more on the number of constraints than on the number of variables: so, it may be convenient to solve the dual of an LP problem with many constraints. LP problems may show a block-diagonal structure, so that they can be decomposed into independent sub-problems when linking constraints are considered separately. LP problems may arise as linear relaxations of reformulations of combinatorial problems with an exponential number of variables.

Table I.1 drawn from Lubbecke-Desrosiers 2002 [58] reports several applications of integer programming column generation.

Table I.1 some applications of integer programming column generetaion

Reference(s)	Application(s)
Agarwal et al. (1989) [59]; Desaulniers et al. (2001b)[60]; Desrochers et al. (1992) [61] Lobel (1997,1998)[62, 63]; Ribeiro and Soumis (1994)[64]	various vehicle routing problems
Borndorfer and Lobel (2001)[65]; Desaulniers et al. (2001b)[60] Desrochers and Soumis (1989)[66]	crew scheduling
Desrosiers et al. (1984) [67]	multiple traveling salesman problem with time windows
Krumke et al. (2002)[68] Lubbecke (2001)[69]; Lubbecke and Zimmermann (2003)[70]; Sol (1994) [71]	real-time dispatching of automobile service units multiple pickup and delivery problem with time windows
Crainic and Rousseau (1987) [72]; Vance et al. (1997) [73]	airline crew pairing
Barnhart and Schneur (1996) [74]	air network design for express shipment service
Erdmann et al. (2001) [75]	airline schedule generation
Barnhart et al. (1998a)[76]; Desaulniers et al. (1997)[77];	fleet assignment and aircraft
Ioachim et al. (1999)[78]	routing and scheduling
Crama and Oerlemans (1994)[79]	job grouping for flexible manufacturing systems
Park et al. (1996)[80]	bandwidth packing in telecommunication networks
Ribeiro et al. (1989) [81]	traffic assignment in satellite communication systems
Sankaran (1995) [82]	course registration at a business school
Vanderbeck (1994) [83]	graph partitioning e.g., in VLSI, compiler design
Vanderbeck (1994) [83]	single-machine multi-item lot-sizing
Hurkens et al. (1997)[84]; Valerio de Carvalho (1999, 2000, 2002b) [85, 86];	bin packing and
Vance (1998)[87]; Vance et al. (1994) [88]; Vanderbeck (1999) [89]	cutting stock problems
Alvelos and Valerio de Carvalho (2000)[86]; Barnhart et al. (1997, 2000) [90, 91]	integer multicommodity flows
Bourjolly et al. (1997)[92]	maximum stable set problem
Hansen et al. (1998)[93]	probabilistic maximum satisfiability problem
Johnson et al. (1993)[94]	minimum cut clustering
Mehrotra and Trick (1996)[95]	graph coloring
Savelsbergh (1997)[96]	generalized assignment problem

I.2 Column Generation Classification

The term 'column generation' has been used in three different but related contexts, all of which apply to problems with a huge number of columns. Rather than enumerating so many columns explicitly, these methods deal with them implicitly, generating a selected set. Each involves a master problem (MP) that is to be optimized. The MP is restricted in the sense that its columns are not all known explicitly. Hence, it is called the restricted MP (RMP).

Type I column generation uses an auxiliary model (AM) to identify an 'attractive' set of columns, defining a RMP that optimizes over these explicitly defined columns. The RMP accepts these columns and does not interact further with the AM.

Type II uses a price-out problem (POP), which interacts with the RMP to identify improving columns. This is the Column Generation type used in this thesis.

Type III, which is based on Dantzig-Wolfe decomposition, employs one or more sub-problems (SPs), which interact with the RMP to identify improving columns.

We report in the following subsections a recap drawn from Wilhem 2001 [97] concerning three model examples from each column generation type.

I.2.1 Type I

Type I column generation employs an AM to generate a large number of feasible columns and a RMP to prescribe the best subset of these columns. This approach was used with noteworthy success in the 1960s and 1970s to schedule airline crews (Arabeyre et al., 1969 [98]; Hoffman and Padberg, 1993 [99]) using the set-partitioning problem as the RMP. The approach can be effective because the set-partitioning problem has a tight linear relaxation but it may result in sub-optimal solutions because the AM may not generate all optimal columns. The following paragraph describes Type I in a historical formulation involving airline crew scheduling.

The airline crew scheduling(ACS) Early approaches to the airline crew-scheduling problem provide classic examples of Type I column generation. The objective is to minimize total crew cost, which represents a significant component in the cost of airline operations.

Data. The following data are given:

- A set I of flight segments $i \in I$, which is defined by type of aircraft and its departure and arrival times for each flight.

- A set J crew rotation $j \in J$, which services a series of segments and may last several days.
- The cost of rotation c_j that includes lodging and other travel expenses, flight time and deadheading in which the crew is repositioned by a flight but does not service it.

Variables. The following variables are used.

- binary variable a_{ij} that assumes value 1 if rotation j serves segment $i \in I$, 0 otherwise,
- binary variable x_j that assumes value 1 if rotation j is prescribed, 0 otherwise $j \in J$,.

Constraints. The following constraints define feasible solutions.

- requirement constraints that states one rotation serve one flight segment.

$$\sum_{j \in J} a_{ij} x_j = 1 \; i \in I \tag{I.1}$$

Objective function. The objective is minimize the total cost of all rotations.

$$\text{Min } Z^{ACS} = \sum_{j \in J} c_j x_j \tag{I.2}$$

This model is well known as the set partitioning problem; rotation j services a subset of the segments $i \in I$ and the problem is to prescribe an optimal subset of the generated rotations to service each segment once.

The first step invokes an auxiliary model (AM) to generate a set of feasible rotations, which is used as a set of explicit columns to define Problem ACS. It is typically not possible to assure that a restricted set of rotations includes a subset that comprises an optimal solution, so the overall approach is a heuristic. Nevertheless, experience has shown that it is possible to prescribe good solutions if the set partitioning problem can be solved effectively for a large number of generated rotations. The second step solves the set partitioning problem with the columns defined explicitly by the AM. A body of research has been directed towards developing algorithms to solve large scale set-partitioning problems (e.g., Marsten, 1974 [100]; Balas and Padberg, 1976 [101]; Marsten and Shepardson, 1981 [102]; Gershkoff, 1989 [103];

Fisher and Kedia, 1990 [104]; Ryan, 1992[105]; Hoffman and Padberg, 1993 [99]; E-Darzi and Mitra, 1995 [106]; Rushmeier et al., 1995 [107];Wedelin, 1995[108]; Atamturk et al., 1996[109]; Sherali and Lee, 1996[110]; Conforti et al., 1999[111]; Gamache et al., 1999[112]; Conforti et al., 2001a, 2001b, 2001c[113, 114, 115]; Fagerholt, 2001[116]). The primary limitation of the Type I method is that the AM generates columns without interacting with the RMP. The RMP can provide information specifically the dual variables, to direct the search for improving columns. Instead of generating a large set of columns with the hope of including the subset that comprises the optimal solution, interaction with the RMP allows columns to be identified as needed. The following subsection describes Type II methods, which do involve interaction between the RMP and SP(s).

I.2.2 Type II

Instead of defining columns explicitly, Type II column generation uses a POP to identify the non basic column with the best Simplex entering criterion. The RMP provides dual variable values to update objective function coefficients in the POP, allowing it to generate improving columns as needed. The following paragraph describes Type II formulation for the cutting stock problem.

Cutting stock (CS) problem. A company stocks rolls (sheets, etc.) of sheet metal (wire, pipe, cloth, lumber, etc.) in standard lengths L_m, $m \in M$. Assume that the company has an unlimited number of each standard length and that its business is to cut these standard lengths into shorter lengths to satisfy customers orders. A cutting pattern may be defined as one combination of shorter lengths that may be cut from one standard length.

Data. The following data are given.

- a set M of lengths m,
- a set I of cut lengths orded,
- a set J of cutting patterns,
- a subset I_j of cut lengths included in the cutting patterns j,
- standard lengths L_m,
- number of pieces ordered b_i,
- lenght ordered by costumers l_i,
- standard lenght cost c_m,
- the cost of the standard roll c_j used in pattern j .

Variables. The following variables are used:

- integer variable a_{ij} denote the number of pieces of length l_i in cutting pattern j,
- integer variable x_j is the number of cutting patterns of type j prescribed $j \in J$.

Constraints. The following constraints define feasible solutions.

- To be feasible, cutting pattern j must observe the limitation imposed by the standard length L_m $m \in M$:

$$\sum_{i \in I} l_i \ a_{ij} \leq \mathrm{L}_m \tag{I.3}$$

- orders for each cut length must be satisfied

$$\sum_{j \in J} a_{ij} \ x_i \leq b_i \ i \in I \tag{I.4}$$

Objective function. The objective is minimize he total cost of filling orders includes the costs of all standard lengths cut to fill the orders

$$\text{Min } Z^{RMP} = \sum_{j \in J} c_j x_j \tag{I.5}$$

The linear relaxation of the cutting stock problem is known to be tight. However, should an integer solution be required, the solution to the linear relaxation could be rounded up. This heuristic is not guaranteed to prescribe the optimal integer solution, however. To guarantee such a solution, the linear relaxation may be used to obtain a lower bound at each node in a branch-and-bound ($B\&B$) search tree.

The primary difficulty is that the number of cutting patterns, $|J|$, may be extremely large, so that solving the LP problem with all $|J|$ columns could take a prohibitive amount of time. Gilmore and Gomory (1961)[55] devised the Type II methodology to circumvent this difficulty. Instead of listing all $|J|$ columns explicitly and pricing each out to identify the column that should enter solution at each simplex iteration, they proposed optimizing a price-out-problem (POP) to determine the entering variable. In fact, an independent POP may be defined for each standard length $m \in M$ with the objective of determining the nonbasic column (i.e., cutting pattern) that minimizes $c_j - z_j$ for each $j \in J$, the LP optimality criterion. It is important to note that the RMP must be optimized over columns that are known explicitly (i.e., considering slack and any other explicit variables) to obtain dual variables that are used to define the objective function in the POP. Defining w_i as the dual variable associated with constraint i of type I.4 for $i \in I$, $\hat{c_m}$ as the cost of standard length L_m for each $m \in M$ and employing decision variables y_i as the number of length l_ji in a cutting pattern j , Problem POP_m may be formulated as:

$$\text{Min } Z_m^{POP} = \hat{c_m} - \sum_{i \in I} w_i^{\star} \; y_i \tag{I.6}$$

$$\sum_{i \in I} l_i \; y_i \leq Ł_m \tag{I.7}$$

$$y_i \leq 0, \; integer, \; i \in I \tag{I.8}$$

Gilmore and Gomory (1961) designed a comprehensive algorithm that forms a prototype for other applications. They proposed a dynamic programming algorithm that solves all $|M|$ POPs simultaneously. They also proposed the use of a knapsack heuristic to solve individual POPs to quickly identify an improving column. If the heuristic cannot generate an improving column, an optimizing procedure must be applied to determine whether or not an improving column exists. Gilmore and Gomory (1963) devised faster knapsack algorithms and problem-specific techniques to deal with a limitation on the number of cutting knives available, the need to balance workloads on multiple cutting machines, and a customer order for a range of amounts instead of a fixed amount. They also explored three problem-specific techniques to facilitate solution. First, they observed that, if a subset of cut lengths have identical dual prices, the shortest length dominates others, which may be eliminated from the knapsack (on this iteration) to speed solution by making the POP smaller. Duplication occurred frequently, enabling POP size to be reduced by about fifty percent. Second, they observed that if a variable entered solution with a large value, it made a large change in the RMP objective function. So, they identified two groups of cut lengths, one with high demand and the other with low demand. Lengths in the former group allow a variable to enter at a large value so they required every other pivot to address only the POPs associated with that group. Tests showed that this device led to significant improvements on both the number of pivots and the run time. Third, they performed tests to compare the use of a heuristic to solve each POP when possible to the use of an optimizing method at all iterations. A heuristic can solve the POP faster but is not guaranteed to prescribe an optimal solution. Their tests showed that optimizing at each iteration reduced run time by significantly reducing the number of RMP pivots.

I.2.3 Type III

Type III column generation applies Dantzig-Wolfe Decomposition (DWD) (Dantzig and Wolfe, 1960 [53]; Bazaraa et al., 1990[117]) to the linear relaxation of an IP. At each iteration, values of dual variables from the RMP update the objective function coefficients in each SP, and one or more SPs are solved to generate an improving column, if possible. The column represents an extreme point (or an extreme ray) of the SP polyhedron. Decision variables in the RMP form a convex combination of the extreme-point based columns (and nonnegative linear combination of extreme rays) generated by each SP. DWD is used to optimize an IP by solving a RMP at each node in the $B\&B$ search tree to obtain a bound on the optimal IP solution value. In this context, column generation is commonly called branch-and-price ($B\&P$) because DWD, which is a price-directed decomposition method (Barnhart et al., 1998 [118]), is applied at each node in the $B\&B$ search tree. The earliest use of DWD to solve an IP appears to be Appelgren (1969 [119]), who addressed a ship-scheduling problem with SPs that were Shortest Path Problems. Due to the structure of his RMP, he was able to obtain integral solutions to the linear relaxation in 98-99% of the instances he tested. A subsequent paper, Appelgren (1971) [120], embedded the DWD within a $B\&B$ search to resolve fractional variables. Appelgren addressed the fundamental issues involved in such an application, including devising a method to select the branching variable and retaining a pool of previously generated columns. In solving relatively large problems that involved scheduling 100 ships over a 6-7 week planning horizon, his approach successfully identified optimal integral solutions requiring very few (5-11) nodes in the $B\&B$ search tree. The following paragraph describes Type III in the formulation involving Generalized Assignment Problem.

Generalized Assignment Problem (GAP) Savelsbergh (1997) formulated GAP as follows:

Data. The following data are given:

- a set J of jobs,
- a set I of agents,
- time capacity b_i for each agent i,
- a_{ij} time required by agent i to process job j,
- profit c_{ij} achieved for each job j processed by agent i.

Variables. The following variables are used:

- binary variable x_{ij} that assumes value 1 if and only if the job j is assigned to agent i and 0 otherwise.

Constraints. The following constraints define feasible solutions:

- assignment constraint, that assure that each job is assigned to one agent

$$\sum_{i \in I} x_{ij} = 1 \; j \in J \tag{I.9}$$

- time constraints which invoke time capacity of each agent i.

$$\sum_{j \in J} a_{ij} x_{ij} \leq b_i \; i \in I \tag{I.10}$$

Objective function.

$$\text{Max } Z^{GAP} = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \tag{I.11}$$

The constraints (I.9) and (I.10) exhibit a block diagonal structure that defines an SP that is associated with each agent $i \in I$. The DWD of the linear relaxation of the GAP leads to RMP:

$$\max Z^{RMP} = \sum_{i \in I} \sum_{j \in J} \left(c_{ij} x_{ij} \right) \lambda_i^k \tag{I.12}$$

$$\text{s.t.} \sum_{i \in I} (x_{ij}^k) \lambda_i^k = 1 \; j \in J \tag{I.13}$$

$$\sum_{k \in K_i} \lambda_i^k = 1 \; i \in I \tag{I.14}$$

$$\lambda_i^k \geq 0, \text{integer } i \in I, \; k \in K_i \tag{I.15}$$

Each row of constraint (I.9) includes a subset of decision variables; in particular x_{ij} for $i \in I$. The corresponding row of the A matrix consists of zeroes, except elements $(j-1)|I| + j, ..., (j-1)|I| + |J|$ that are 1. Thus, we define the product $Ax = \sum_{i \in I} x_{ij}$ as denoted in (I.13). Problem SP_i may be formulated as

$$\max Z_i^{SP} = \sum_{j \in J} (c_{ij} - w_{ij}^\star) \; x_{ij} - \alpha_i^\star \tag{I.16}$$

$$\text{s.t.} \sum_{j \in J} a_{ij} x_{ij} \leq b_i \tag{I.17}$$

$$x_{ij} \in \{0, 1\}, \; j \in J \tag{I.18}$$

Set X includes binary requirements so that SP_i is a binary knapsack problem. For a given set of dual variables (i.e. $w_j^\star$ $(j \in J)$ and $\alpha_i^\star$ $(i \in I)$)

let binary x_{ij}^k give optimum, $Z_i^{SP\star}$. If $Z_i^{SP} > 0, x_{ij}^k$ generates an improving column that enters the RMP. If $Z_i^{SP\star} \leq 0$ for all $i \in I$, the current solution of the RMP is optimal.

The convex hull of feasible integral solutions to SP_i is contained in the polytope associated with its linear relaxation, so the polytope defined by (I.16-I.18) does not have all integral extreme points. Thus, SPs do not exhibit the Integrality Property, and the solution to the RMP is at least as tight as the solution to the linear relaxation of model (I.9) (I.10) (I.11) (Geoffrion, 1974)[121]. In fact, the bound provided by the RMP is equal to that provided by a Lagrangian relaxation that relaxes semi-assignment constraints (I.9) into the objective function. The Lagrangian dual (LD) is
Problem LD:

$$\mathrm{Min}_{\pi}\mathrm{Max}_{x}\ Z^{LD} = \Big\{ \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{j \in J} \pi \Big(1 - x_{ij} \Big) \Big\}$$

constraint (I.10) and integrality constraint.

So, the RMP, an LP, is the dual of the Lagrangian dual that results from relaxing the complicating constraints $Ax \geq b$ (Geoffrion, 1974)[121].

It is possible to obtain bounds that are tighter than those from the linear relaxation by solving a SP over the convex hull of feasible integer points. The SP could be solved easily (e.g., by LP) if all extreme points of its polytope were integral. Unfortunately, as in the related Lagrangian relaxation method, this Integrality Property assures that the decomposition will not give bounds that are tighter than those from the linear relaxation of the original model. If a sub-problem does not exhibit the Integrality Property, it must be solved as an IP, so, in practice, it is only possible to obtain tighter bounds if a sub-problem has a special structure that can be solved effectively, for example, in polynomial or pseudo-polynomial time. For example, algorithms have been devised to solve the knapsack problem and the CSPP in pseudo-polynomial time. Thus, a model must achieve a tradeoff between SP tractability and the tightness of bounds. Nemhauser and Wolsey (1988) [122] and Bazaraa et al. (1990)[123] provide more details on the relationships between Lagrangian relaxation and DWD.

Appendix II

Stochastic Programming

This appendix is tracted from the Introduction Tutorial on Stochastic Linear Programming Models from Higle and Sen[124].

II.1 Motivation

It is often difficult to precisely estimate or forecast certain data of linear programs. In such cases, it is necessary to address the impact of uncertainty during the planning process. In deterministic activity analysis, planning consists of choosing activity levels that satisfy resource constraints while maximizing total profit (or minimizing total cost). All the information necessary for decision making is assumed to be available at the time of planning. Under uncertainty, not all the information is available, and some parameters should be modeled as random variables. Since deterministic methodology has been prevalent in optimization models, it may be tempting to suggest that random variables should be replaced by their means and the resulting optimization problem solved. In general, this approach provides solutions that are structurally different from those provided by stochastic optimization models.

II.2 Impact of uncertainty

The presence of uncertainty affects both feasibility and optimality. In fact, formulating an appropriate objective function itself raises interesting modeling and algorithmic questions.

II.2.1 Feasibility under Uncertainty

To incorporate uncertainty within an LP, one must define feasibility. Two naive approaches have sometimes been adopted in practice. One concerns

SLP with Expected Values and one follows a "wait and see" strategy.

SLP with Expected Values Consider the following four variable deterministic LP:

$$\text{Minimize } x_2 \tag{II.1}$$
$$\text{s.t.} x_1 + x_2 + x_3 = 2 \tag{II.2}$$
$$-x_1 + x_2 + x_4 = 2 \tag{II.3}$$
$$-1 \leq x_1 \leq 1 \tag{II.4}$$
$$x_j > 0, \qquad j = 2, 3, 4. \tag{II.5}$$

Suppose that the coefficients of x_1 and x_2 in II.3 are not known with certainty, and all that is known about these parameters is their joint distribution.

$$(\tilde{a}_{21}, \tilde{a}_{22}) = \begin{cases} (1, \frac{3}{4}\ ) & \text{with probability } \frac{1}{2} \\ (-3, \frac{5}{4}) & \text{with probability} \frac{1}{2}. \end{cases}$$

In this case, $\mathbb{E}[\tilde{a_{21}}] = -1$ and $\mathbb{E}[\tilde{a_{22}}] = 1$ so that the coefficients in (II.3) correspond to the expected values of the random variables. In examining this formulation, we first investigate whether its solution, $(x_1, x_2, x_3, x_4) =$ (0, 2, 0, 0), is feasible under uncertainty. Under uncertainty, the constraint corresponding to (II.3) is equally likely to be either

$$x_1 + \frac{3}{4} + x_2 + x_4 = 2$$

or

$$-3x_1 + \frac{5}{4} + x_2 + x_4 = 2.$$

The vector (0, 2, 0, 0) does not satisfy either of these equations and thus is infeasible under uncertainty! Under uncertainty, the formulation in which random variables are replaced by their expected values may not provide a solution that is feasible with respect to the random variables.

Wait and see. Another approach that practitioners often adopt is based on a wait-and-see analysis (sometimes referred to as scenario analysis or what-if analysis). This approach mimics the process of delaying all decisions until the last possible moment, after all uncertainties have been resolved. As a result, the LPs associated with all possible outcomes of the random quantities are solved. This yields a collection of decision vectors, one for each possible outcome of the random variable(s). In general, none of these solutions may be worthwhile. For example, consider the two possible realizations of the problem in Example 1. The solution associated with $(\tilde{a_{21}}, \tilde{a_{22}}) = (1, \frac{3}{4})$ is $(-1, 3, 0, 0.75)$, while the solution associated with $(\tilde{a_{21}}, \tilde{a_{22}}) = (-3, \frac{5}{4})$ is $(-\frac{2}{17}, \frac{32}{17}, 0, 0.75)$. As with the solution to the expected-value problem, neither of these solutions is feasible with respect to the alternate outcome. That is, if implemented, either solution would have a 50- percent chance of failing to satisfy a constraint.

Modelling response to future infeasibility. As illustrated by Examples 1 and 2, an appropriate decision-making framework under uncertainty should explicitly consider the consequences of future infeasibility within the model. This aspect of modeling responses to future infeasibility sets stochastic programs apart from their deterministic counterparts. In the stochastic programming literature, two approaches are widely studied: one is based on modeling future recourse (response) and another restricts the probability of infeasibility (typically equivalent to system failures) to be no greater than a prespecified threshold. The first approach yields the so-called recourse problems, and the second approach yields problems with probabilistic (or chance) constraints. We focus on recourse problems. The stochastic-programming literature also considers another problem: the distribution problem. Researchers focus on characterizing the distribution of the optimal value or optimal solutions of random LPs. As with wait-and-see problems, the distribution problem does not provide a decision-making framework. Nevertheless,it provides a mathematical common ground between the second-stage random LP in recourse problems and the random LP of the wait-and-see approach. From a computational point of view, this problem remains a major challenge [125].

II.2.2 Optimality under Uncertainty.

A great deal of research revolves around the choice of objectives in decision making under uncertainty. One of the more common objectives is to optimize expected costs (or returns). However, as decision makers, we might be interested in the variability of costs (or returns) associated with a plan. More generally, a decision maker's choices may be guided by a utility function. In decision-making models for an individual, the concept of a utility function has many merits, although its specification can be elusive. The notion of a utility function can become even more elusive in large-scale applications of LP. In the following, we discuss four of the more common objectives for large-scale LPs under uncertainty:

- Minimization of expected costs is by far the most common objective used in large-scale optimization under uncertainty. For such applications as planning power generation, average seasonal cost per day reflects the repetitive cost of supplying electricity. For some applications in telecommunications systems, system performance is often measured in terms of average unserved demand. Finally, in production-and-inventory systems, it is common to use average production and holding costs in evaluating the cost effectiveness of a system. For such systems, the expected cost criterion is easily justified.

- Minimization of expected absolute deviations from goals is a class of objectives that results from extending goal programming techniques to account for uncertainty. In some cases, it may be advantageous to specify goals that depend upon particular scenarios. For example, production goals may depend upon economic factors that are modeled as uncertain quantities. Thus, the goals associated with prosperous and recessionary times may be decidedly different. To meet such managerial objectives under uncertainty, it may be appropriate to minimize expected absolute deviations from set goals.

- Minimization of maximum costs is an alternate class of models. There are various interpretations of the term minimax in stochastic programming models. In one interpretation, no distributional information is available, and all that is known is the set of possible outcomes. In this case, the minimax objective minimizes the maximum loss among all possible outcomes of the random variable. A similar class of problems arises in the case of partial information regarding the probability distributions. For instance, one may have information regarding some characteristics of the distribution (for example, support, mean, and variance), and the set of probability measures of interest may be those that share those characteristics. A worst-case approach under partial information is one in which we choose a decision that minimizes maximum expected loss, regardless of the distribution (from among the class with the specified characteristics). When the class of distributions can be characterized as a polyhedral set, this class of problems can be solved using generalized LP. This minimax approach is known to be conservative and may be appropriate in models that plan to avoid catastrophes. Thus, models associated with environmental planning may appropriately use this objective [126]. In this tutorial, we discuss primarily models with the expected value objective.

II.3 Two-Stage Recourse Models

In two-stage recourse models, we explicitly classify the decision variables according to whether they are implemented before or after an outcome of the random variable is observed. Decisions that are implemented before are known as first-stage decisions while those after are second-stage decisions. The first-stage decision variables can be regarded as proactive and are often associated with planning issues, such as capacity expansion or aggregate production planning. Second-stage decision variables can be regarded as reactive and are often associated with operating decisions. These second-stage decisions allow us to model a response to the observed outcome, which constitutes our recourse. When outcomes are revealed sequentially, decision making involves a multistage planning problem.

In recourse planning, we model a response for each outcome of the random elements that might be observed. In general, this response will also depend upon the first-stage decisions. In practice, this type of planning involves setting up policies that will help the organization adapt to the revealed outcome. For example, in production and inventory systems, the first-stage decision might correspond to production quantities, and demand might be modeled using random variables. When demand exceeds the amount produced, policy may dictate that customer demand be backlogged at some cost. This policy constitutes a recourse response. The exact level of this response (the amount backlogged) depends on the amounts produced and demanded. Under uncertainty, it is essential to adopt initial policies that will accommodate alternative outcomes. Consequently, modeling under uncertainty requires that we incorporate a model of the recourse policy.

In some applications, it is possible to deviate from prescribed limits, although with a penalty cost. For example, in production and inventory management, a backlogging policy leads to shortage costs whenever the demand exceeds the amount in stock.

II.3.1 Simple recourse policy formulation

For a generic two-stage formulation under a simple recourse policy, we use an extension of the notation used in deterministic LP. The rows of a deterministic LP are usually written as $Ax = b$. Under uncertainty, we may think of a submatrix A_1 (of A) and a subvector b_1 (of b) as rows that contain only deterministic parameters. We refer to this portion of the problem as the deterministic part. It corresponds to a first stage of the problem. The remaining rows (containing at least one random element) will be indexed by the set R. We refer to a_i as the ith row vector in A, and use a $\sim$ to reflect the presence of random variables. Let $g_i > 0$ denote the penalty cost for violating the target $\tilde{b}_i$ Then we can state a prototypical model allowing a simple recourse policy as follows:

$$\text{Minimize } cx + \sum_{in\in R} g_i \mathbb{E}[|\tilde{b}_i - \tilde{a}_i x|] \tag{II.6}$$

$$\text{subject to } A_1 x = b_1 \tag{II.7}$$

$$a_{is}x + y_{is}^{+} - y_{is}^{-} = b_{is} \qquad \forall s \in S_i \forall i \in R \tag{II.8}$$

$$L_1 \leq x \leq \mathcal{U}_1. \tag{II.9}$$

This is an SLP with simple recourse. In such an SLP, the first-stage decision variables (x) are the same as the decision variables associated with the 'parent' deterministic LP. Hence, the formulation is not flexible in its response to uncertainty. Whenever the random vectors $\{(\tilde{a}_i, \tilde{b}_i)\}_{i\in R}$ are discrete random variables as in II.2.1, this model can be rewritten as a linear program as shown below. For $i \in R$, let S_i denote an index set of all outcomes of the random vector $\{(\tilde{a}_i, \tilde{b}_i)\}$ and let $p_{is} = P\{(\tilde{a}_i, \tilde{b}_i) = (a_{is}, b_{is})\}$.

$$\text{Minimize } cx + \sum_{i \in R} g_i\Big(\sum_{s \in S_i} p_{is}(y_{is}^+ + y_{is}^-)\Big) \tag{II.10}$$

$$\text{subject to } A_1 x = b_1 \tag{II.11}$$

$$a_{is}x + y_{is}^+ - y_{is}^- = b_{is} \qquad \forall s \in S_i \forall i \in R. \tag{II.12}$$

$$L_1 \leq x \leq \mathcal{U}_1. \tag{II.13}$$

In this formulation, the penalty cost per unit is the same whether $\tilde{a}_i(x) - \tilde{b}_i$ is positive or negative. In some applications, the cost may be nonzero only in one of these two cases. More generally, the per unit cost of $\tilde{b}_i - \tilde{a}_i x$ may be g_i^+ for positive values (of this random variable) and g_i^- for negative values. In this case, the costs used for compensating variables (y_i^+, y_i^-) are g_i^+ and g_i^- and the objective function must be changed to reflect this.

Finally, in stating the SLP with simple recourse, we have assumed that the upper and lower bounds are not subject to uncertainty. In some situations, these bounds may be random. Suppose, for example, that the upper bounds reflect capacity restrictions. When systems fail, such capacity limits may be modeled as random variables. Assuming a simple recourse policy, we can easily extend the statement of the model to include this situation.

While the simple recourse policy offers a notion of feasibility for first-stage plans, the recourse actions themselves are quite limited. For example, in a production-andinventory system that is experiencing shortages, a simple recourse policy is one that simply allows the manufacturer to adopt an out-sourcing option. A more general recourse policy would allow changes in production rates, thus allowing greater flexibility. Under uncertainty, greater flexibility translates into greater responsiveness and greater profitability.

II.3.2 General recourse model

As with the formulation of a simple recourse model, we will present the general recourse formulation as an extension of an LP problem:

$$\text{Minimize } cx \tag{II.14}$$

$$\text{s.t.} Ax = b \tag{II.15}$$

$$L \leq x \leq \mathcal{U}. \tag{II.16}$$

Suppose that the decision maker specifies a subvector of x, say x_1, as the first stage decision variables. These variables cannot be postponed until better information is available. The remaining variables, say x_2, can be postponed. Naturally, with this temporal division of the problem, two types of constraints arise: constraints that involve only the first-stage variables (x_1), and constraints that may involve both sets of variables. Thus, it is convenient to think of a submatrix A_1 (of A) and a subvector b_1 (of b) yielding a subset of the constraints, $A_1x_1 = b_1$. The remaining constraints involve x_1 and x_2, which we write as $Bx_1 + A_2x_2 = b_2$. Finally, the cost vector c is partitioned as (c_1, c_2) so that we may rewrite the formulation as

$$\text{Minimize } c_1x_1 + c_2x_2 \tag{II.17}$$
$$\text{s.t.} A_1x_1 = b_1 \tag{II.18}$$
$$Bx_1 + A_2x_2 = b_2 \tag{II.19}$$
$$L_1 \le x_1 \le \mathcal{U}_1, \tag{II.20}$$
$$L_2 \le x_2 \le \mathcal{U}_2. \tag{II.21}$$

It is convenient to think of this deterministic LP as the 'core' problem from which the stochastic LP will be derived. It models the time-staged dynamics of the interactions among the decision variables.

The constraints $A_1x_1 = b_1$ include the immediate constraints, those that involve only the variables that cannot be delayed. As such, there are no random variables in the immediate data (c_1, A_1, b_1). The random variables appear in the second stage of the problem, which includes the variables x_2 and can be postponed until the uncertainties are realized. Thus, we consider the second-stage data to include random variables, so that we express them as $(\tilde{c_2}, \tilde{B}, \tilde{a_2}, \tilde{b_2})$, (here, we use $\sim$ to indicate a random entity).

To formulate the stochastic LP, let S denote an index set of all possible outcomes of the second-stage quantities $(\tilde{B}, \tilde{a_2}, \tilde{c_2}, \tilde{b_2})$ such that each $s \in S$ corresponds to a unique realization of these quantities $(B_s, A_{2s}, c_{2s}, b_{2s})$. If S is a discrete set, then for each $s \in S$, let $p_s = P\{(\tilde{B}, \tilde{a_2}, \tilde{c_2}, \tilde{b_2}) = (B_s, A_{2s}, c_{2s}, b_{2s})\}$. Also, let x_{2s} denote the recourse response associated with scenario s. The two-stage program with general recourse may now be written as follows:

$$\text{Minimize } c_1x_1 + \sum_{s\in S} p_s c_{2s} x_{2s} \tag{II.22}$$
$$\text{s.t. } A_1x_1 = b_1 \tag{II.23}$$
$$B_s x_1 + A_{2s}x_{2s} = b_{2s} \qquad \forall s \in S \tag{II.24}$$
$$L_1 \le x_1 \le \mathcal{U}_1, L_2 \le x_{2s} \le \mathcal{U}_2 \qquad \forall s \in S. \tag{II.25}$$

This formulation is unlike the simple recourse formulation, in that some (or perhaps all) choices of x_1 that satisfy (II.22) can render (II.25) infeasible for some scenarios. It is possible to include compensating variables (with positive penalty costs) to ensure that the resulting problem is feasible. Furthermore, it can be shown that this extended formulation always has a lower optimal value than a formulation in which the decision maker restricts all decision variables in x (the vector from the deterministic LP) to be first-stage decisions and only a simple recourse policy is allowed in the second stage.

The stochastic program with general recourse is also referred to as a problem with random recourse, since the matrices A_{2s} are allowed to depend on the outcome $s \in S$. However, since the term random recourse might be misconstrued as a case in which the decision maker has no control over the recourse policy, we use the term general recourse. When the matrices A_{2s} are the same for all $s \in S$ (that is, A_2 is not random), the stochastic program is said to have fixed recourse. Even in such cases, the random right-hand-side vector, causes, $\tilde{b}_2$, the recourse decision itself to vary with s, and hence the fixed-recourse formulation retains the variables $x_{2s}, s \in S$. Finally, the special case of fixed recourse, in which $A_{2s} = [I, -I]$ (where I denotes an identity matrix) yields the simple recourse model discussed earlier.

A general recourse problem is said to have complete recourse if for any choice of x_1, a feasible recourse decision is possible for all outcomes $s \in S$. The simple recourse formulation possesses complete recourse. A slightly less restrictive property is that of relatively complete recourse whereby one requires that a feasible recourse decision be possible for all outcomes s provided the first-stage decision (x_1) satisfies the first-stage constraints ($A_1 x_1 = b_1, L_1 \leq x_1 \leq \mathcal{U}_1$). By using penalty costs for deviations from constraint satisfaction, one can ensure complete recourse in any problem.

One of the more important notions incorporated within a stochastic programming formulation is that of implementability (or nonanticipativity). This term reflects the requirement that under uncertainty, the planning decisions (x_1) must be implemented before an outcome of the random variable is observed. That is, the planning decision is made while the random variable is still unknown, and therefore it cannot be based on any particular outcome of the random variable. Thus the wait-and-see approach, which is anticipative, is not an appropriate decision-making framework for planning. On the other hand, the here-and-now approach embodied in the two-stage SLP with general recourse provides planning decisions (x_1) that are not dependent on any outcome of the random variable and hence are nonanticipative. An alternate statement of this requirement is given in the scenario formulation below:

$$\text{Minimize} \sum_{s \in S} p_s[c_1 x_{1s} + c_{2s} x_{2s}] \tag{II.26}$$

$$\text{subject to } A_1 x_{1s} = b_1 \qquad \forall s \in S \tag{II.27}$$

$$B_s x_{1s} + A_{2s} x_{2s} = b_{2s} \qquad \forall s \in S \tag{II.28}$$

$$x_1 - x_{1s} = 0 \qquad \forall s \in S \tag{II.29}$$

$$L_1 \leq x_{1s} \leq \mathcal{U}_1, \tag{II.30}$$

$$L_2 \leq x_{2s} \leq U_2 \;\; s \in S. \tag{II.31}$$

In this formulation, the variables x_{1s} are dependent on the outcome s. However, constraint II.29 explicitly enforces implementability by requiring that all outcomes agree on the same planning decision x_1. We can obtain a slightly more compact representation of this formulation by requiring first $A_1 x_1 = b_1$ and then requiring II.29. By doing so, we avoid replicating the first set of constraints for each outcome. Both of these are equivalent representations of the two-stage SLP with general recourse. The particular representation used typically depends on the algorithm being used to solve the problem. Note that the general recourse problem is a finite dimensional linear program whenever S is a finite set. However, whenever the random variable is continuous these formulations lead to infinite dimensional problems. Under these circumstances, it is more convenient to state the model in the following decomposed form:

$$\text{Minimize } cx_1 + \mathbb{E}[\tilde{h}(x_1)] \tag{II.32}$$

$$\text{subject to } A_1 x_1 \leq b_1 \tag{II.33}$$

$$L_1 \leq x_1 \leq \mathcal{U}_1 \tag{II.34}$$

where each outcome $h_s(x)$ of the random variable $\tilde{h}(x)$ is a function of the LP defined by the outcome $(c_{2s}, A_{2s}, B_s, b_{2s})$ of the random variable $(\tilde{c}_2, \tilde{a}_2, \tilde{B}_s, \tilde{b_2})$, that is,

$$h_s(x_1) = \text{Minimize } c_{2s} x_{2s} \tag{II.35}$$

$$\text{subject to } A_{2s} x_{2s} = b_{2s} - B_s x_1 \tag{II.36}$$

$$L_2 \leq x_{2s} \leq \mathcal{U}_2. \tag{II.37}$$

This decomposed formulation is convenient when the sample space S contains either a large number of atoms (in the case of discrete random variables) or a continuum (in the case of continuous random variables). The function $\mathbb{E}[\tilde{h}(x1)]$ is referred to as the recourse function. This formulation emphasizes the time-staged nature of the decision problem. That is, the selection of x_1 is followed by the selection of x_2, which is undertaken in response to the scenario that unfolds. Thus, the first decision, x_1, represents the immediate commitment made, while the second decision is delayed until additional information is obtained. For this reason, when solving a recourse problem, one typically reports only the first-stage decision vector.

Appendix III

Results

In this chapter we present the results of further instances variations. We consider four numbers of events, that are 25, 50, 75 and 100. For each event $\mathcal{E}$ set we construct six event set partitions $\mathcal{E}_t$ according to the number of event types $|\mathcal{T}|$. that are 3, 4, 5, 6, 7 and 8. Furthermore we create 10 u_{pt} parameter variations. Each subsection present the results for each event set $\mathcal{E}$ corresponding to 25, 50, 75 and 100 events.

III.1 Dataset 25 events

The following tables present the results that regards the 25 events instances.

Table III.1 Instance 25E- LRMP objective function

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	46142.0	46142.0	44269.6	43389.0	40515.0	41681.3	38538.0	41963.0
B	46142.0	42854.0	39346.0	41989.0	44009.5	43074.0	38.549.4	40.570.4
C	46058.0	43040.0	44542.6	40552.5	38913.8	42380.6	37.990.4	38.893.1
D	46142.0	45729.3	43212.0	39355.0	43289.5	41494.8	38.095.5	40.471.0
E	44557.0	42978.0	46142.0	46042.0	39174.2	42720.0	42.881.0	38.999.2
F	46142.0	46036.8	41388.1	44570.7	42674.4	41780.0	45.068.2	41.986.4
G	46142.0	44912.1	37618.0	45472.4	44560.1	42672.4	36.992.0	41.584.2
H	46142.0	46058.0	43728.5	36405.5	44313.8	41641.7	42.279.0	38.324.4
I	46142.0	46142.0	43519.0	40006.3	45200.0	44045.0	42.132.8	41.125.0
J	46142.0	45366.0	43905.1	44041.4	44145.0	42539.8	44985.3	41188.2

Table III.2 Instance 25E - LRMP gap

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	0.1%	0.4%	0.4%	0.4%	0.4%	0.5%	0.4%	0.1%
B	0.1%	0.1%	0.5%	0.5%	0.5%	0.0%	0.5%	0.5%
C	0.3%	0.4%	0.3%	0.4%	0.3%	0.3%	0.8%	0.3%
D	0.3%	0.3%	0.0%	0.5%	0.6%	0.5%	0.5%	0.5%
E	0.3%	0.4%	0.5%	0.5%	0.4%	0.4%	0.8%	0.5%
F	0.1%	0.4%	0.2%	0.3%	0.5%	0.5%	0.5%	0.6%
G	0.2%	0.4%	0.4%	0.5%	0.6%	0.5%	0.8%	0.8%
H	0.0%	0.4%	0.2%	0.2%	0.5%	0.2%	0.8%	0.2%
I	0.2%	0.1%	0.4%	0.7%	0.6%	0.5%	0.7%	0.9%
J	0.1%	0.2%	0.4%	0.4%	0.4%	0.2%	0.7%	0.3%

Table III.3 Instance 25E - IMP HRMP objective function

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	46115	45976	44075	43223	40368	41480	38392	41934
B	46113	42797	39167	41797	43804	43074	38359	40349
C	45924	42888	44414	40390	38788	42244	37687	38770
D	46026	45597	43209	39170	43042	41306	37906	40249
E	44425	42809	45923	45810	39027	42569	42524	38798
F	46118	45864	41294	44448	42450	41554	44837	41736
G	46037	44753	37478	45250	44300	42475	36707	41263
H	46122	45895	43633	36337	44097	41540	41951	38243
I	46053	46115	43336	39732	44915	43830	41851	40761
J	46102	45277	43750	43877	43962	42443	44688	41054

Table III.4 Instance 25E - IMP HRMP gap

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	7.4%	7.1%	23.2%	22.2%	23.9%	39.1%	23.2%	61.0%
B	7.4%	0.0%	0.4%	25.9%	18.7%	26.4%	30.0%	20.4%
C	7.2%	0.4%	27.6%	24.7%	18.3%	25.7%	17.3%	37.0%
D	7.4%	12.0%	12.8%	22.6%	32.7%	26.8%	18.9%	19.5%
E	7.2%	23.7%	30.1%	27.7%	22.5%	19.5%	42.8%	43.0%
F	7.4%	11.8%	26.6%	25.3%	24.5%	32.9%	30.4%	34.2%
G	7.4%	8.7%	13.3%	35.3%	31.8%	35.7%	19.0%	26.7%
H	11.8%	7.0%	25.3%	16.1%	22.8%	29.0%	42.9%	32.1%
I	7.4%	11.2%	20.6%	19.0%	33.0%	36.6%	11.8%	29.5%
J	11.8%	5.5%	25.6%	26.0%	31.7%	34.5%	34.2%	50.6%

Table III.5 Instance 25E - IMP LRMP execution time (sec)

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	1.8	1.6	1.4	1.1	1.1	1.1	1.0	0.7
B	1.8	2.7	1.8	1.2	1.3	1.2	0.9	0.8
C	2.1	1.7	1.5	1.4	1.1	1.1	0.9	0.6
D	1.8	1.6	1.4	1.2	1.2	1.1	0.9	0.7
E	2.1	1.6	1.3	1.0	1.4	1.2	0.9	0.7
F	2.0	1.7	1.4	1.5	1.0	1.0	1.0	0.8
G	1.7	1.8	1.6	1.4	1.1	1.3	0.8	0.9
H	1.8	1.6	1.2	1.1	1.1	1.1	1.0	0.8
I	1.9	1.6	1.5	1.2	1.3	1.0	0.8	0.7
J	1.9	2.3	1.5	1.4	1.5	1.1	0.9	0.9

Table III.6 Instance 25E - IMP HRMP execution time (sec)

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	1.9	1.7	1.4	1.2	1.2	1.1	1.0	0.7
B	1.9	2.8	1.8	1.2	1.3	1.3	0.9	0.8
C	2.2	1.7	1.5	1.4	1.1	1.1	1.0	0.6
D	1.8	1.6	1.4	1.2	1.3	1.1	0.9	0.7
E	2.1	1.6	1.3	1.1	1.4	1.2	1.0	0.7
F	2.1	1.7	1.4	1.5	1.1	1.0	1.0	0.9
G	1.7	1.8	1.6	1.5	1.1	1.3	0.8	0.9
H	1.8	1.6	1.2	1.1	1.1	1.2	1.0	0.8
I	1.9	1.7	1.6	1.2	1.3	1.0	0.8	0.7
J	2.0	2.4	1.6	1.4	1.5	1.1	0.9	1.0

Table III.7 Instance 25E - BMP HRMP objective function

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	42724	42854	34012	33777	30833	25369	29587	16350
B	42724	42854	39204	31109	35786	31706	27000	32300
C	42724	42854	32238	30526	31801	31478	31407	24516
D	42724	40224	37680	30453	29117	30371	30895	32592
E	41364	32808	32238	33295	30379	34387	24532	22222
F	42724	40610	30371	33295	32238	28018	31354	27610
G	42724	40990	32610	29415	30411	27448	29955	30474
H	40677	42854	32685	30539	34217	29583	24140	26008
I	42724	40990	34573	32394	30267	27943	37168	28985
J	40677	42854	32685	325876	30165	27881	29584	20362

Table III.8 Instance 25E - BMP HRMP gap

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	7.4%	7.1%	23.2%	22.2%	23.9%	39.1%	23.2%	61.0%
B	7.4%	0.0%	0.4%	25.9%	18.7%	26.4%	30.0%	20.4%
C	7.2%	0.4%	27.6%	24.7%	18.3%	25.7%	17.3%	37.0%
D	7.4%	12.0%	12.8%	22.6%	32.7%	26.8%	18.9%	19.5%
E	7.2%	23.7%	30.1%	27.7%	22.5%	19.5%	42.8%	43.0%
F	7.4%	11.8%	26,6%	25,3%	24,5%	32,9%	30,4%	34,2%
G	7.4%	8.7%	13,3%	35,3%	31,8%	35,7%	19,0%	26,7%
H	11.8%	7.0%	25,3%	16,1%	22,8%	29,0%	42,9%	32,1%
I	7.4%	11.2%	20,6%	19,0%	33,0%	36,6%	11,8%	29,5%
J	11.8%	5.5%	25,6%	26,0%	31,7%	34,5%	34,2%	50,6%

Table III.9 Instance 25E - BMP LRMP execution time (sec)

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	1.6	1.9	1.7	1.7	1.5	1.4	1.2	0.8
B	1.7	3.4	2.6	1.4	1.7	1.3	1.0	1.0
C	2.2	2.8	1.7	1.7	1.6	1.4	1.0	0.7
D	1.6	1.7	1.8	1.5	1.5	1.4	0.9	1.0
E	2.5	1.7	1.2	1.1	1.8	1.7	1.0	0.8
F	1.8	1.9	1.8	1.8	1.6	1.2	0.9	0.9
G	1.6	2.4	2.2	1.2	1.3	1.6	1.4	0.9
H	1.5	1.7	1.4	1.2	1.2	1.6	0.8	1.0
I	1.6	1.7	2.0	1.9	1.3	1.3	1.0	1.0
J	1.9	2.3	1.6	1.5	1.2	1.5	0.8	0.9

Table III.10 Instance 25E - BMP HRMP execution time (sec)

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	1.6	1.9	1.7	1.7	1.5	1.4	1.2	0.8
B	1.7	3.4	2.6	1.5	1.7	1.3	1.0	1.0
C	2.2	2.9	1.7	1.7	1.6	1.4	1.0	0.7
D	1.6	1.7	1.8	1.5	1.5	1.4	1.0	1.0
E	2.5	1.7	1.2	1.1	1.8	1.7	1.0	0.8
F	1.8	1.9	1.8	1.8	1.6	1.2	0.9	0.9
G	1.6	2.4	2.2	1.2	1.3	1.7	1.4	0.9
H	1.5	1.7	1.4	1.3	1.2	1.7	0.8	1.0
I	1.6	1.7	2.0	1.9	1.4	1.3	1.0	1.0
J	1.9	2.3	1.6	1.6	1.2	1.5	0.8	0.9

III.2 Dataset 50 events

The following tables present the results that regards the 50 events instances.

Table III.11 Instance 50E- LRMP objective function

u_{pt}	number of event types $\lvert\mathcal{T}\rvert$							
variation	3	4	5	6	7	8	9	10
A	100440.0	100440.0	96951.0	96262.0	91314.3	84195.0	84482.6	88272.6
B	100440.0	98443.0	84518.0	97146.0	99304.1	85072.5	75298.0	86432.3
C	100440.0	99850.0	96826.9	95406.0	90021.0	84694.0	83054.4	93953.8
D	100440.0	100440.0	96960.4	90081.7	98652.0	82416.6	87494.0	91265.0
E	98478.0	98469.0	99051.0	100214.0	94362.0	86128.0	82820.2	93563.4
F	100440.0	97913.5	96074.7	98229.3	94127.0	77089.4	88845.0	93737.9
G	100440.0	98160.0	88455.0	100240.0	96416.3	82436.1	94374.3	94353.3
H	100440.0	100440.0	98177.5	87936.5	98385.8	83817.8	82234.7	87109.9
I	100440.0	100440.0	82154.8	93197.3	98983.3	85033.9	91929.0	89570.6
J	100440.0	99850.0	99313.3	96864.3	97602.0	85270.0	95415.5	95422.0

Table III.12 Instance 50E- LRMP gap

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	0.0%	0.0%	3.5%	4.2%	9.1%	16.2%	15.9%	12.1%
B	0.0%	2.0%	15.9%	3.3%	1.1%	15.3%	25.0%	13.9%
C	0.0%	0.6%	3.6%	5.0%	10.4%	15.7%	17.3%	6.5%
D	0.0%	0.0%	3.5%	10.3%	1.8%	17.9%	12.9%	9.1%
E	2.0%	2.0%	1.4%	0.2%	6.1%	14.2%	17.5%	6.8%
F	0.0%	2.5%	4.3%	2.2%	6.3%	23.2%	11.5%	6.7%
G	0.0%	2.3%	11.9%	0.2%	4.0%	17.9%	6.0%	6.1%
H	0.0%	0.0%	2.3%	12.4%	2.0%	16.5%	18.1%	13.3%
I	0.0%	0.0%	18.2%	7.2%	1.5%	15.3%	8.5%	10.8%
J	0.0%	0.6%	1.1%	3.6%	2.8%	15.1%	5.0%	5.0%

Table III.13 Instance 50E- IMP HRMP objective function

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	100209	100119	96634	95808	90716	83750	83828	87763
B	100288	98071	84120	96782	98730	84798	74648	86128
C	100288	98524	96371	94906	89516	84560	82747	93066
D	100278	100075	96607	89587	98184	81897	87223	90731
E	98257	98249	98597	99783	93821	85810	82469	92785
F	100143	97671	95633	97720	93582	76647	88389	93257
G	100332	97873	88172	99668	96046	81944	93949	93870
H	100308	100047	97737	87509	97919	83426	81776	86745
I	100164	100271	81769	92934	98248	84585	91210	88695
J	100173	99456	98826	96475	97054	84743	94843	94829

Table III.14 Instance 50E- IMP HRMP gap

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	0.2%	0.3%	0.3%	0.5%	0.7%	0.5%	0.8%	0.6%
B	0.2%	0.4%	0.5%	0.4%	0.6%	0.3%	0.9%	0.4%
C	0.2%	1.3%	0.5%	0.5%	0.6%	0.2%	0.4%	0.9%
D	0.2%	0.4%	0.4%	0.5%	0.5%	0.6%	0.3%	0.6%
E	0.2%	0.2%	0.5%	0.4%	0.6%	0.4%	0.4%	0.8%
F	0.3%	0.2%	0.5%	0.5%	0.6%	0.6%	0.5%	0.5%
G	0.1%	0.3%	0.3%	0.6%	0.4%	0.6%	0.5%	0.5%
H	0.1%	0.4%	0.4%	0.5%	0.5%	0.5%	0.6%	0.4%
I	0.3%	0.2%	0.5%	0.3%	0.7%	0.5%	0.8%	1.0%
J	0.3%	0.4%	0.5%	0.4%	0.6%	0.6%	0.6%	0.6%

Table III.15 Instance 50E- IMP LRMP execution time (sec)

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	3.9	2.9	3.3	3.0	2.2	1.7	1.7	1.6
B	4.4	5.1	3.3	2.6	2.4	2.0	1.7	1.7
C	6.3	3.7	3.3	2.4	1.8	1.8	1.8	1.8
D	4.6	3.3	3.2	2.2	2.4	1.6	2.2	1.8
E	4.3	3.2	3.0	2.3	2.7	2.1	1.5	1.9
F	3.4	4.2	3.4	2.6	2.4	2.0	2.2	1.7
G	3.2	3.9	3.7	2.6	2.5	2.2	2.3	1.9
H	3.7	3.7	3.4	2.7	2.8	1.8	1.8	1.9
I	4.3	3.5	3.0	2.7	2.6	1.9	1.8	1.6
J	3.9	5.5	3.6	2.7	2.3	2.1	2.3	1.8

Table III.16 Instance 50E- IMP HRMP execution time (sec)

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	4.0	2.9	3.3	3.0	2.2	1.7	1.7	1.7
B	4.6	5.8	3.4	2.6	2.4	2.0	1.7	1.8
C	6.4	3.9	3.4	2.4	1.9	1.9	1.8	1.8
D	4.8	3.3	3.3	2.2	2.4	1.6	2.2	1.9
E	4.5	3.3	3.1	2.3	2.7	2.1	1.5	1.9
F	3.5	4.3	3.5	2.6	2.5	2.0	2.2	1.8
G	3.2	4.1	3.8	2.6	2.5	2.2	2.4	1.9
H	3.8	3.8	3.4	2.8	2.9	1.9	1.9	1.9
I	4.3	3.6	3.1	2.7	2.6	1.9	1.9	1.7
J	4.0	5.5	3.6	2.7	2.4	2.2	2.3	1.9

Table III.17 Instance 50E- BMP HRMP objective function

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	95926	98038	79879	73476	66262	52839	59612	44171
B	95926	98038	78444	70815	65162	74686	48746	57838
C	95926	98850	71166	75109	62668	62721	62052	61787
D	95926	83824	83298	67933	63672	53730	65851	76488
E	90186	77622	71166	71034	67319	64925	47135	62509
F	95926	84396	78331	73880	65458	57108	53181	50922
G	95926	84396	82686	72016	63076	66274	60361	73652
H	89879	98038	65874	71880	72281	54980	57581	68101
I	95926	81355	75297	80228	64135	56151	70560	64568
J	89879	98038	67264	71415	70048	57268	65865	50752

Table III.18 Instance 50E- BMP HRMP gap

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	4.5%	2.4%	17.6%	23.7%	27.4%	37.2%	29.4%	50.0%
B	4.5%	0.4%	7.2%	27.1%	34.4%	12.2%	35.3%	33.1%
C	4.5%	1.0%	26.5%	21.3%	30.4%	25.9%	25.3%	34.2%
D	4.5%	16.5%	14.1%	24.6%	35.5%	34.8%	24.7%	16.2%
E	8.4%	21.2%	28.2%	29.1%	28.7%	24.6%	43.1%	33.2%
F	4.5%	13.8%	18.5%	24.8%	30.5%	25.9%	40.1%	45.7%
G	4.5%	14.0%	6.5%	28.2%	34.6%	19.6%	36.0%	21.9%
H	10.5%	2.4%	32.9%	18.3%	26.5%	34.4%	30.0%	21.8%
I	4.5%	19.0%	8.3%	13.9%	35.2%	34.0%	23.2%	27.9%
J	10.5%	1.8%	32.3%	26.3%	28.2%	32.8%	31.0%	46.8%

Table III.19 Instance 50E- BMP LRMP execution time (sec)

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	4.5	4.1	4.1	3.8	3.2	2.3	2.5	2.0
B	4.7	6.7	4.8	3.8	2.5	1.9	2.6	2.2
C	4.7	3.7	4.2	3.6	2.8	2.7	2.7	2.5
D	4.8	4.1	4.0	3.1	2.3	2.5	2.4	2.7
E	6.1	5.0	3.5	2.4	3.6	2.2	2.3	2.4
F	4.8	5.4	4.2	3.3	3.2	2.4	2.5	2.0
G	4.1	4.9	6.2	2.6	2.6	2.4	2.7	2.4
H	3.9	4.9	4.4	3.7	2.5	2.5	3.3	2.3
I	4.8	3.9	4.6	4.3	2.6	1.9	3.5	2.6
J	4.7	6.2	4.5	3.5	3.1	2.6	2.2	2.3

Table III.20 Instance 50E- BMP HRMP execution time (sec)

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	4.5	4.1	4.2	3.8	3.3	2.3	2.5	2.0
B	4.7	6.7	4.9	3.8	2.5	1.9	2.6	2.2
C	4.7	3.9	4.2	3.6	2.8	2.7	2.8	2.6
D	4.8	4.1	4.0	3.1	2.3	2.5	2.4	2.7
E	6.2	5.1	3.5	2.4	3.7	2.3	2.4	2.5
F	4.8	5.4	4.3	3.3	3.2	2.4	2.5	2.1
G	4.1	4.9	6.3	2.7	2.6	2.4	2.7	2.4
H	3.9	5.0	4.4	3.8	2.5	2.5	3.3	2.3
I	4.8	3.9	4.6	4.4	2.6	1.9	3.5	2.6
J	4.7	6.2	4.6	3.5	3.1	2.6	2.2	2.4

III.3 Dataset 75 events

The following tables present the results that regards the 75 events instances.

Table III.21 Instance 75E - LRMP objective function

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	153550.0	152510.0	151536.0	152886.0	137667.0	144404.0	127268.0	133786.0
B	153550.0	150358.0	119880.0	149267.0	146593.0	146132.0	124758.0	133841.0
C	153550.0	153550.0	150796.0	148933.0	142939.0	141908.0	129656.0	142727.0
D	153550.0	153550.0	150517.0	146758.0	147744.0	144681.0	134407.0	138398.0
E	152272.0	151052.0	151519.0	150741.0	139041.0	150242.0	145902.0	140386.0
F	153550.0	148183.0	139512.0	149237.0	137965.0	144942.0	150860.0	145997.0
G	153550.0	148912.0	132082.0	149185.0	146268.0	141186.0	138221.0	133820.0
H	153550.0	153550.0	149234.0	130231.0	150929.0	151083.0	138685.0	140635.0
I	153550.0	153550.0	134752.0	140131.0	151796.0	151083.0	145755.0	145621.0
J	153550.0	153550.0	150134.0	148187.0	151210.0	139250.0	150404.0	141889.0

Table III.22 Instance 75E - LRMP gap

u_{pt}	number of event types $\lvert\mathcal{T}\rvert$							
variation	3	4	5	6	7	8	9	10
A	0.0%	0.7%	1.3%	0.4%	10.3%	6.0%	17.1%	12.9%
B	0.0%	2.1%	21.9%	2.8%	4.5%	4.8%	18.8%	12.8%
C	0.0%	0.0%	1.8%	3.0%	6.9%	7.6%	15.6%	7.0%
D	0.0%	0.0%	2.0%	4.4%	3.8%	5.8%	12.5%	9.9%
E	0.8%	1.6%	1.3%	1.8%	9.4%	2.2%	5.0%	8.6%
F	0.0%	3.5%	9.1%	2.8%	10.1%	5.6%	1.8%	4.9%
G	0.0%	3.0%	14.0%	2.8%	4.7%	8.1%	10.0%	12.8%
H	0.0%	0.0%	2.8%	15.2%	1.7%	1.6%	9.7%	8.4%
I	0.0%	0.0%	12.2%	8.7%	1.1%	1.6%	5.1%	5.2%
J	0.0%	0.0%	2.2%	3.5%	1.5%	9.3%	2.0%	7.6%

Table III.23 Instance 75E - IMP HRMP objective function

u_{pt}	number of event types $\lvert\mathcal{T}\rvert$							
variation	3	4	5	6	7	8	9	10
A	153222	151926	150929	152257	136912	143595	126194	133026
B	153204	149787	119336	148420	145994	145242	123658	133016
C	153342	152943	150149	148053	142045	140926	128949	141326
D	153185	153076	149806	146145	146848	143895	133197	137105
E	151875	150687	150843	150089	138189	149550	144876	139127
F	153289	147631	138873	148451	137097	143847	150073	145145
G	153122	148368	131486	148674	145278	140345	137422	132547
H	153163	152990	148662	129678	150059	150112	137574	139871
I	153137	153217	134339	139645	150820	150112	144758	144687
J	153092	152976	149430	147321	150337	138565	149512	140635

Table III.24 Instance 75E - IMP HRMP gap

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	0.2%	0.4%	0.4%	0.4%	0.5%	0.6%	0.8%	0.6%
B	0.2%	0.4%	0.5%	0.6%	0.4%	0.6%	0.9%	0.6%
C	0.1%	0.4%	0.4%	0.6%	0.6%	0.7%	0.5%	1.0%
D	0.2%	0.3%	0.5%	0.4%	0.6%	0.5%	0.9%	0.9%
E	0.3%	0.2%	0.4%	0.4%	0.6%	0.5%	0.7%	0.9%
F	0.2%	0.4%	0.5%	0.5%	0.6%	0.8%	0.5%	0.6%
G	0.3%	0.4%	0.5%	0.3%	0.7%	0.6%	0.6%	1.0%
H	0.3%	0.4%	0.4%	0.4%	0.6%	0.6%	0.8%	0.5%
I	0.3%	0.2%	0.3%	0.3%	0.6%	0.6%	0.7%	0.6%
J	0.3%	0.4%	0.5%	0.6%	0.6%	0.5%	0.6%	0.9%

Table III.25 Instance 75E - IMP LRMP execution time (sec)

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	4.1	4.6	4.2	4.6	4.0	3.6	3.2	2.2
B	4.3	6.0	5.7	4.2	4.4	4.3	2.9	2.8
C	5.0	4.6	4.2	3.4	3.7	3.8	3.2	2.9
D	4.8	4.3	4.2	3.8	4.1	3.6	3.3	2.9
E	6.1	5.0	4.0	3.8	4.2	4.3	3.3	2.8
F	6.0	4.7	3.7	3.6	4.7	3.6	3.0	2.8
G	4.4	4.7	4.1	3.8	4.0	4.3	3.3	2.5
H	3.9	5.1	4.0	4.2	4.1	3.9	3.3	2.7
I	4.7	4.9	4.1	3.6	4.7	3.9	3.6	2.9
J	4.3	5.2	4.0	4.4	3.8	3.5	3.2	3.2

Table III.26 Instance 75E - IMP HRMP execution time (sec)

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	4.3	4.6	4.3	4.7	4.4	3.7	3.2	2.3
B	4.5	6.1	5.8	4.2	4.5	4.3	2.9	2.9
C	5.1	4.7	4.4	3.5	3.8	3.9	3.2	2.9
D	4.8	4.4	4.3	3.9	4.1	3.6	3.3	2.9
E	37.1	5.1	4.1	3.8	4.3	4.3	3.3	2.9
F	7.1	4.8	3.8	3.6	4.8	3.6	3.1	2.8
G	4.6	4.9	4.2	3.9	4.0	4.3	3.4	2.6
H	4.0	5.4	4.1	4.2	4.2	3.9	3.3	2.8
I	4.7	5.1	4.2	3.6	4.8	3.9	3.6	2.9
J	4.9	5.4	4.0	4.4	3.9	3.5	3.2	3.3

Table III.27 Instance 75E - BMP HRMP objective function

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	146415	146557	115644	127835	107811	86423	76411	69122
B	146415	146557	119480	107831	98661	90605	82372	96718
C	146415	146557	101341	109348	96249	87337	95892	96389
D	146415	124024	136001	109409	93927	95607	107471	117074
E	141396	122948	104617	109536	104801	110091	83541	83058
F	146415	127675	97755	109536	101854	87603	86177	82244
G	146415	127675	118051	107327	104474	83690	97126	104701
H	141516	146557	103551	113025	101640	94264	85911	123379
I	146415	127010	124235	113758	103410	94715	112242	91334
J	141516	146557	100327	113025	106654	91012	85378	72722

Table III.28 Instance 75E - BMP HRMP gap

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	4.6%	3.9%	23.7%	16.4%	21.7%	40.2%	40.0%	48.3%
B	4.6%	2.5%	0.3%	27.8%	32.7%	38.0%	34.0%	27.7%
C	4.6%	4.6%	32.8%	26.6%	32.7%	38.5%	26.0%	32.5%
D	4.6%	19.2%	9.6%	25.4%	36.4%	33.9%	20.0%	15.4%
E	7.1%	18.6%	31.0%	27.3%	24.6%	26.7%	42.7%	40.8%
F	4.6%	13.8%	29.9%	26.6%	26.2%	39.6%	42.9%	43.7%
G	4.6%	14.3%	10.6%	28.1%	28.6%	40.7%	29.7%	21.8%
H	7.8%	4.6%	30.6%	13.2%	32.7%	37.6%	38.1%	12.3%
I	4.6%	17.3%	7.8%	18.8%	31.9%	37.3%	23.0%	37.3%
J	7.8%	4.6%	33.2%	23.7%	29.5%	34.6%	43.2%	48.7%

Table III.29 Instance 75E - BMP LRMP execution time (sec)

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	5.9	6.4	5.6	5.0	6.3	4.4	4.2	2.2
B	9.0	19.2	9.0	6.0	4.9	5.3	4.6	4.0
C	7.3	6.5	5.5	5.6	6.0	4.8	5.3	3.7
D	7.0	5.8	5.8	5.5	6.5	4.5	3.8	3.4
E	10.3	6.6	5.5	5.1	9.2	5.2	3.7	4.0
F	9.3	13.8	5.3	6.2	6.0	4.7	3.5	3.1
G	6.6	10.7	7.6	8.6	6.0	5.3	5.0	3.3
H	7.4	7.2	5.9	6.0	5.6	5.0	5.1	3.8
I	7.5	5.4	7.3	7.0	6.0	4.0	4.6	3.2
J	7.3	6.5	5.4	6.9	6.1	4.7	3.7	3.8

Table III.30 Instance 75E - BMP HRMP execution time (sec)

u_{pt}	number of event types $\lvert\mathcal{T}\rvert$							
variation	3	4	5	6	7	8	9	10
A	6.1	6.5	5.6	5.1	6.4	4.4	4.3	2.3
B	9.0	19.3	9.0	6.1	4.9	5.3	4.6	4.1
C	7.4	6.6	5.6	5.6	6.0	4.8	5.3	3.7
D	7.0	5.8	5.9	5.5	6.5	4.6	3.8	3.4
E	10.4	6.6	5.5	5.1	9.3	5.3	3.8	4.2
F	9.4	13.9	5.4	6.2	6.1	4.8	3.5	3.2
G	6.7	10.8	7.6	8.7	6.0	5.4	5.0	3.4
H	7.4	7.2	5.9	6.1	5.7	5.2	5.2	3.8
I	7.6	5.4	7.4	7.1	6.1	4.1	4.6	3.2
J	7.4	6.6	5.4	7.0	6.2	4.7	3.8	3.8

III.4 Dataset 100 events

The following tables present the results that regards the 100 events instances.

Table III.31 Instance 100E - LRMP objective function

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	205708.00	203144.00	198559.00	198693.00	189210.00	200686.00	174796.00	178253.00
B	205708.00	205688.00	181143.00	202355.00	197191.00	199857.00	171633.00	174956.00
C	205708.00	205708.00	202612.00	193362.00	191174.00	199187.00	177435.00	181647.00
D	205708.00	205708.00	202206.00	186509.00	203551.00	193827.00	171766.00	181977.00
E	203514.00	205708.00	203870.00	204948.00	195069.00	200805.00	192982.00	175347.00
F	205708.00	194463.00	195763.00	200224.00	195921.00	192704.00	199401.00	174307.00
G	205708.00	197598.00	176557.00	198281.00	198466.00	198471.00	198564.00	174219.00
H	205708.00	203206.00	203931.00	182637.00	202158.00	201502.00	187340.00	171927.00
I	205708.00	205708.00	181764.00	196714.00	202291.00	199741.00	193957.00	175566.00
J	205708.00	205568.00	205299.00	196416.00	198745.00	194245.00	200892.00	181489.00

Table III.32 Instance 100E - LRMP gap

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	0.0%	1.2%	3.5%	3.4%	8.0%	2.4%	15.0%	13.3%
B	0.0%	0.0%	11.9%	1.6%	4.1%	2.8%	16.6%	14.9%
C	0.0%	0.0%	1.5%	6.0%	7.1%	3.2%	13.7%	11.7%
D	0.0%	0.0%	1.7%	9.3%	1.0%	5.8%	16.5%	11.5%
E	1.1%	0.0%	0.9%	0.4%	5.2%	2.4%	6.2%	14.8%
F	0.0%	5.5%	4.8%	2.7%	4.8%	6.3%	3.1%	15.3%
G	0.0%	3.9%	14.2%	3.6%	3.5%	3.5%	3.5%	15.3%
H	0.0%	1.2%	0.9%	11.2%	1.7%	2.0%	8.9%	16.4%
I	0.0%	0.0%	11.6%	4.4%	1.7%	2.9%	5.7%	14.7%
J	0.0%	0.1%	0.2%	4.5%	3.4%	5.6%	2.3%	11.8%

Table III.33 Instance 100E - IMP HRMP objective function

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	205157	202371	197635	197925	188122	199276	173819	177464
B	205127	204809	180267	201435	196201	198255	170516	173982
C	205173	205222	201649	192422	190373	198014	175989	180532
D	205304	205001	201390	185488	202449	192517	170413	180251
E	202931	204920	202909	204066	194385	199716	191771	174017
F	205195	193825	194834	199305	194968	191542	198185	173492
G	205108	196888	175814	197361	197327	197268	196908	173248
H	205132	202613	203297	181650	200839	200269	185967	170381
I	205436	205084	181078	195652	201474	198748	192613	174889
J	205256	204865	204508	195281	197794	192869	199815	180623

Table III.34 Instance 100E - IMP HRMP gap

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	0.3%	0.4%	0.5%	0.4%	0.6%	0.7%	0.6%	0.4%
B	0.3%	0.4%	0.5%	0.5%	0.5%	0.8%	0.7%	0.6%
C	0.3%	0.2%	0.5%	0.5%	0.4%	0.6%	0.8%	0.6%
D	0.2%	0.3%	0.4%	0.5%	0.5%	0.7%	0.8%	0.9%
E	0.3%	0.4%	0.5%	0.4%	0.4%	0.5%	0.6%	0.8%
F	0.2%	0.3%	0.5%	0.5%	0.5%	0.6%	0.6%	0.5%
G	0.3%	0.4%	0.4%	0.5%	0.6%	0.6%	0.8%	0.6%
H	0.3%	0.3%	0.3%	0.5%	0.7%	0.6%	0.7%	0.9%
I	0.1%	0.3%	0.4%	0.5%	0.4%	0.5%	0.7%	0.4%99
J	0.2%	0.3%	0.4%	0.6%	0.5%	0.7%	0.5%	0.5%

Table III.35 Instance 100E - IMP LRMP execution time (sec)

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	6.3	7.9	5.5	8.8	5.6	4.0	4.5	3.5
B	8.7	10.0	7.1	10.4	5.7	5.0	4.6	3.0
C	8.2	7.8	6.6	8.1	5.5	5.4	5.4	3.4
D	6.7	7.2	6.1	7.9	7.8	4.8	4.7	4.2
E	9.0	7.0	6.8	7.8	8.5	5.9	4.8	6.2
F	8.6	8.3	5.9	8.9	7.9	5.1	5.0	5.6
G	8.5	7.1	6.6	9.3	7.4	6.1	5.5	6.3
H	6.9	10.1	6.7	7.9	6.4	5.1	5.1	5.3
I	8.0	7.2	7.0	8.7	6.6	5.6	6.1	5.0
J	8.4	10.9	12.5	7.9	8.8	4.9	5.1	5.2

Table III.36 Instance 100E - IMP HRMP execution time (sec)

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	6.6	8.0	5.6	9.0	5.6	4.0	4.6	3.6
B	10.7	13.2	7.6	10.5	5.7	5.0	4.7	3.0
C	9.0	8.2	6.9	8.3	5.5	5.4	5.4	3.4
D	6.8	7.8	6.4	8.0	7.8	4.8	4.8	4.3
E	16.6	7.3	6.9	8.0	8.5	6.0	4.8	6.3
F	8.8	8.6	6.4	9.5	7.9	5.1	5.1	5.6
G	8.9	7.4	6.8	9.5	7.4	6.1	5.6	6.3
H	7.4	10.9	6.9	8.0	6.5	5.2	5.1	5.3
I	8.3	7.7	7.1	8.8	6.7	5.6	6.2	5.0
J	8.7	14.0	12.9	7.9	8.8	4.9	5.2	5.3

Table III.37 Instance 100E - BMP HRMP objective function

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	195304	203032	161827	152980	132022	123111	137836	98472
B	195304	195337	175708	145803	127806	141168	102727	106825
C	195304	195337	134267	124591	110638	115845	130942	140609
D	195304	195337	171273	143218	138432	122407	105974	140547
E	190912	195337	144191	147514	132534	139670	106416	111955
F	195304	195337	138668	150744	140728	119982	108991	88652
G	195304	195337	155901	151026	128640	118808	101829	167806
H	186837	195337	139892	145863	138125	130812	109836	153976
I	195304	195337	155159	173083	140502	118839	135170	142207
J	186837	195337	137367	146920	147124	140436	108237	95119

Table III.38 Instance 100E - BMP HRMP gap

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	5,1%	0,1%	18,5%	23,0%	30,2%	38,7%	21,1%	44,8%
B	5,1%	5,0%	3,0%	27,9%	35,2%	29,4%	40,1%	38,9%
C	5,1%	5,0%	33,7%	35,6%	42,1%	41,8%	26,2%	22,6%
D	5,1%	5,0%	15,3%	23,2%	32,0%	36,8%	38,3%	22,8%
E	6,2%	5,0%	29,3%	28,0%	32,1%	30,4%	44,9%	36,2%
F	5,1%	-0,4%	29,2%	24,7%	28,2%	37,7%	45,3%	49,1%
G	5,1%	1,1%	11,7%	23,8%	35,2%	40,1%	48,7%	3,7%
H	9,2%	3,9%	31,4%	20,1%	31,7%	35,1%	41,4%	10,4%
I	5,1%	5,0%	14,6%	12,0%	30,5%	40,5%	30,3%	19,0%
J	9,2%	5,0%	33,1%	25,2%	26,0%	27,7%	46,1%	47,6%

Table III.39 Instance 100E - BMP LRMP execution time (sec)

u_{pt}	number of event types $\|\mathcal{T}\|$							
variation	3	4	5	6	7	8	9	10
A	9.6	16.6	8.8	12.2	8.7	5.6	6.0	5.0
B	11.3	18.7	11.9	10.4	7.8	6.9	6.6	4.8
C	11.4	20.1	8.5	11.7	6.8	6.3	8.1	4.5
D	10.3	21.1	8.5	10.2	7.1	7.4	5.3	4.4
E	12.5	22.3	9.2	7.5	9.9	8.7	9.2	5.1
F	9.9	23.5	9.5	10.5	9.7	8.2	9.1	4.6
G	7.6	24.6	11.6	11.5	8.8	7.9	6.4	5.5
H	8.4	25.7	9.6	10.5	6.0	8.4	7.8	3.9
I	9.6	26.8	9.5	12.1	7.3	8.8	6.1	5.3
J	10.4	27.9	8.3	9.6	10.2	8.6	4.5	4.6

Table III.40 Instance 100E - BMP HRMP execution time (sec)

u_{pt}	number of event types $\vert\mathcal{T}\vert$							
variation	3	4	5	6	7	8	9	10
A	10.0	16.8	8.8	12.4	8.8	5.6	6.1	5.1
B	11.7	19.7	12.1	10.4	7.9	7.0	6.7	5.0
C	11.8	20.8	8.6	11.8	6.9	6.4	8.2	4.6
D	10.7	22.0	8.6	10.3	7.1	7.5	5.3	4.5
E	12.6	23.1	9.2	7.6	10.0	8.7	9.3	5.1
F	10.3	24.3	9.6	10.6	9.8	8.3	9.1	4.7
G	7.7	25.4	11.8	11.6	8.8	8.0	6.4	5.5
H	8.5	26.4	9.7	10.8	6.1	8.6	7.9	3.9
I	10.0	27.6	9.6	12.7	7.4	8.8	6.2	5.3
J	10.5	28.7	8.3	9.8	10.4	8.7	4.5	4.6

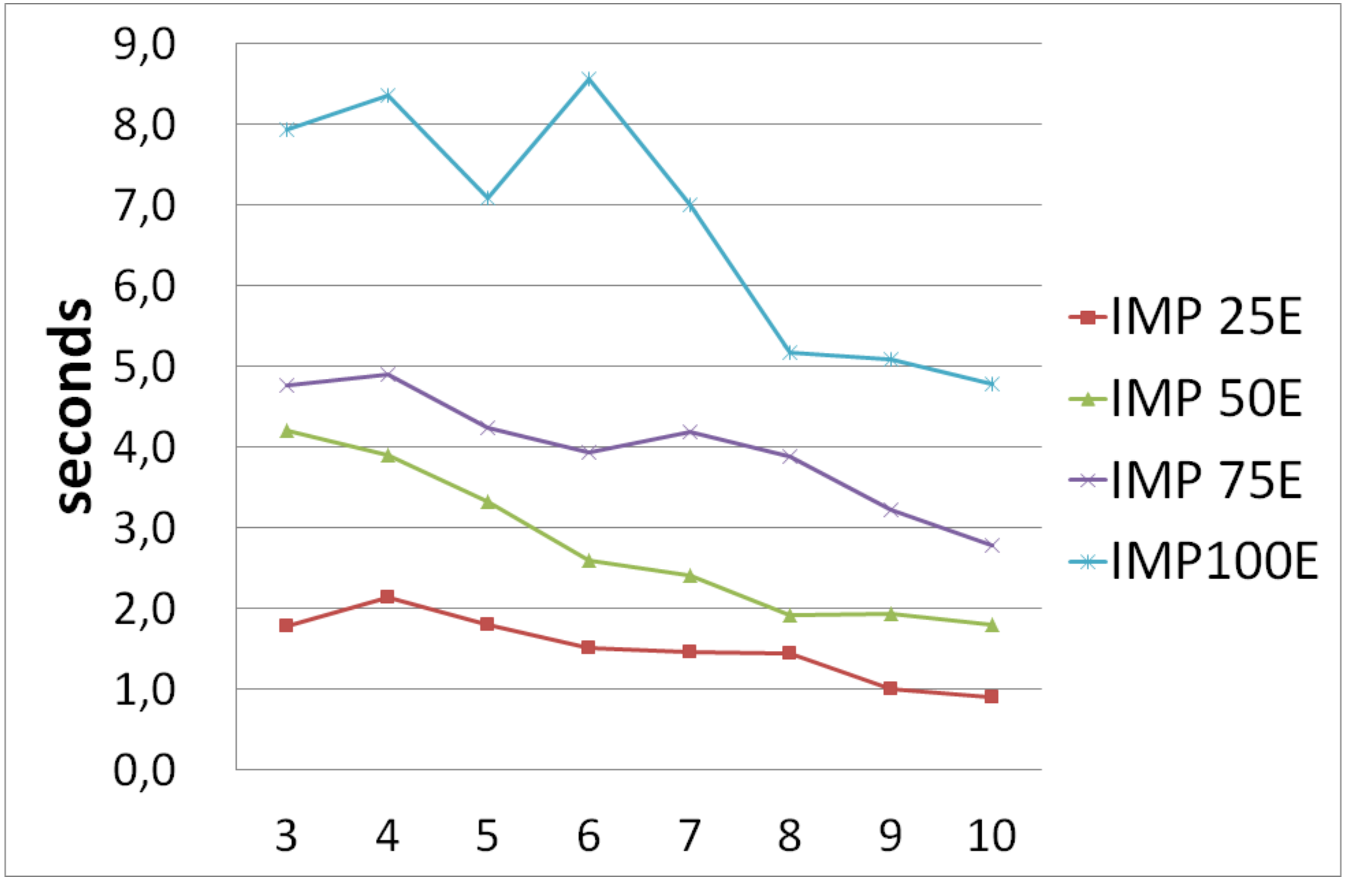

Figure III.1: Computing time datasets 25E, 50E, 75E and 100E

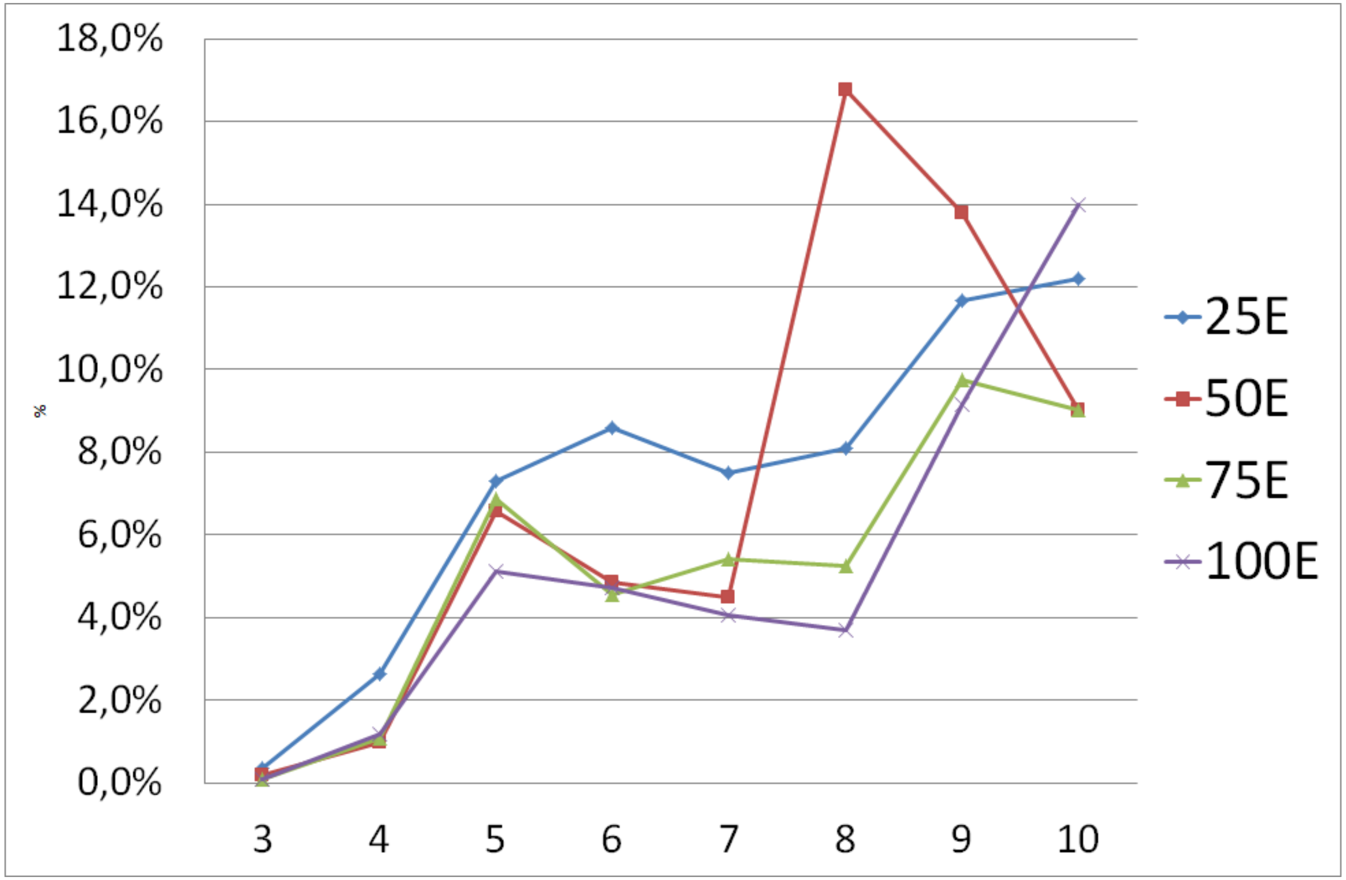

Figure III.2: LRMP gap datasets 25E, 50E, 75E and 100E

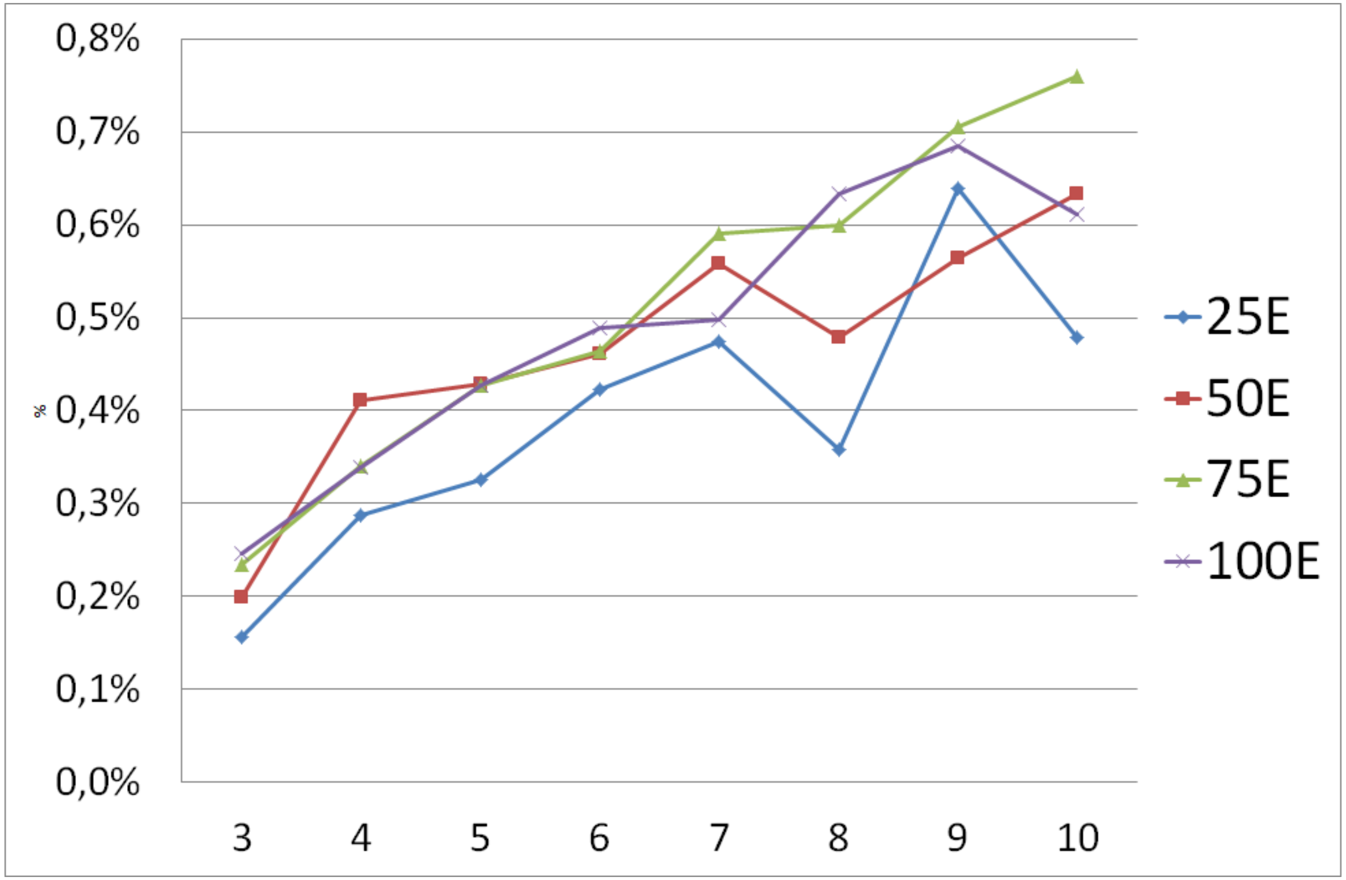

Figure III.3: IMP HRMP gap datasets 25E, 50E, 75E and 100E

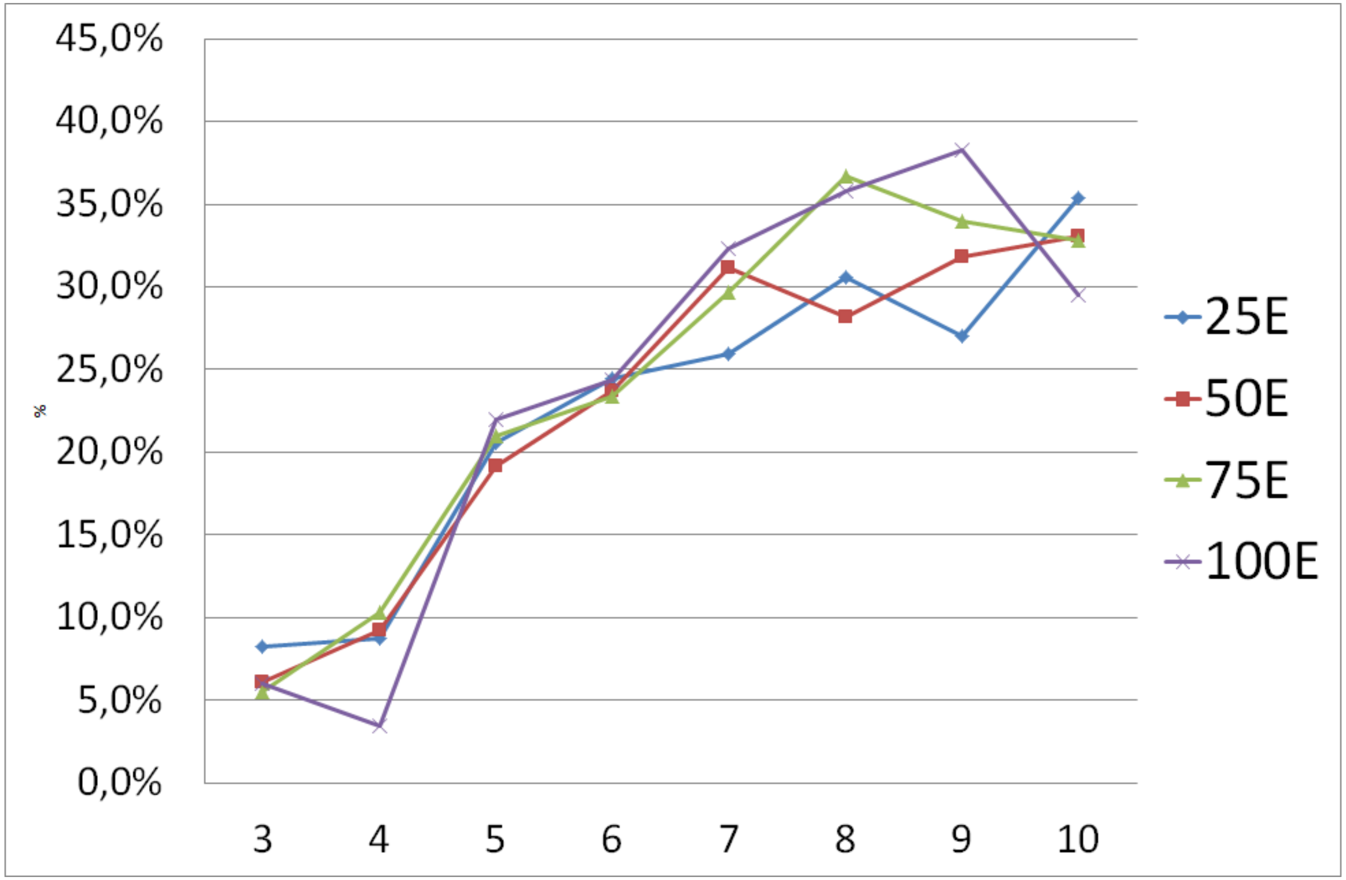

Figure III.4: BMP HRMP gap datasets 25E, 50E, 75E and 100E

III.5 Parametric Formulation

The following tables present the results that regards the Parametric Formulation using dataset 25t6 and its five u_{pt} variations.

Table III.41 Parametric Formulation 25t6 - LRMP objective function

u_{pt}	parameter l					
variation	1	5	8	10	25	50
A	43389.0	43389.0	43389.0	43389.0	43.389.0	43.389.0
B	41989.0	41989.0	41989.0	41989.0	41989.0	41.989.0
C	40552.5	40552.5	40552.5	40552.5	40552.5	40.552.5
D	39355.0	39355.0	39355.0	39355.0	39355.0	39.355.0
E	46042.0	46042.0	46042.0	46042.0	46042.0	46.042.0

Table III.42 Parametric Formulation 25t6 - LRMP gap

u_{pt}	parameter l					
variation	1	5	8	10	25	50
A	6,0%	6,0%	6,0%	6,0%	6,0%	6,0%
B	9,0%	9,0%	9,0%	9,0%	9,0%	9,0%
C	12,1%	12,1%	12,1%	12,1%	12,1%	12,1%
D	14,7%	14,7%	14,7%	14,7%	14,7%	14,7%
E	0,2%	0,2%	0,2%	0,2%	0,2%	0,2%

Table III.43 Parametric Formulation 25t6 - HRMP objective function

u_{pt}	parameter l					
variation	1	5	8	10	25	50
A	43087	41057	39548	38840	33825	20150
B	41222	39623	38996	36325	32572	18575
C	40367	37321	35680	36580	30784	22600
D	39176	36150	34836	35510	29741	15600
E	45530	42621	41296	40460	33332	19775

Table III.44 Parametric Formulation 25t6 - HRMP gap

u_{pt}	parameter l					
variation	1	5	8	10	25	50
A	0.7%	5.4%	8.9%	10.5%	22.0%	53.6%
B	1.8%	5.6%	7.1%	13.5%	22.4%	55.8%
C	0.5%	8.0%	12.0%	9.8%	24.1%	44.3%
D	0.5%	8.1%	11.5%	9.8%	24.4%	60.4%
E	1.1%	7.4%	10.3%	12.1%	27.6%	57.1%

Table III.45 Parametric Formulation 25t6 - LRMP execution time (sec)

u_{pt}	parameter l					
variation	1	5	8	10	25	50
A	7 .9	2 .1	2 .3	2 .2	1 .5	1 .6
B	7 .9	2 .2	2 .5	2 .1	1 .9	1 .6
C	12 .3	2 .6	3 .1	2 .5	1 .7	1 .7
D	14 .7	2 .7	3 .1	2 .6	1 .6	1 .8
E	5 .9	1 .8	2 .2	1 .8	1 .2	1 .2

Table III.46 Parametric Formulation 25t6 - HRMP execution time (sec)

u_{pt}	parameter l					
variation	1	5	8	10	25	50
A	7.9	2.1	2.4	2.2	1.6	1.6
B	8.0	2.2	2.6	2.1	2.0	1.6
C	12.3	2.6	3.2	2.5	1.8	1.7
D	14.7	2.7	3.1	2.6	1.7	1.9
E	5.9	1.8	2.3	1.9	1.3	1.2

III.6 BMP model with groups

The following tables present the results that regards the BMP model with groups using dataset 25t6 and its five u_{pt} variations.

Table III.47 BMP model with groups 25t6- LRMP objective function

u_{pt}	group size r							
variation	1	5	10	15	20	25	35	50
A	43389.0	43389.0	43389.0	43389.0	43389.0	43389.0	43389.0	35766.0
B	41989.0	41989.0	41989.0	41989.0	41989.0	41989.0	41989.0	34590.0
C	40552.5	40552.5	40552.5	40552.5	40552.5	40552.5	40552.5	34343.0
D	46042.0	46042.0	46042.0	46042.0	46042.0	46042.0	39355.0	32270.5
E	44570.8	44570.8	44570.8	44570.8	44570.8	44570.8	44570.8	38587.3

Table III.48 BMP model with groups 25t6 - LRMP gap

u_{pt}	group size r							
variation	1	5	10	15	20	25	35	50
A	6.0%	6.0%	6.0%	6.0%	6.0%	6.0%	6.0%	22.5%
B	9.0%	9.0%	9.0%	9.0%	9.0%	9.0%	9.0%	25.0%
C	12.1%	12.1%	12.1%	12.1%	12.1%	12.1%	12.1%	25.6%
D	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	14.7%	30.1%
E	3.4%	3.4%	3.4%	3.4%	3.4%	3.4%	3.4%	16.4%

Table III.49 BMP model with groups 25t6 - HRMP objective function

u_{pt}	group size r							
variation	1	5	10	15	20	25	35	50
A	33777	32955	30450	31035	27900	27450	26705	25400
B	31109	30760	29190	28080	26600	26400	20265	23650
C	30526	29810	28490	31545	28420	24375	23450	24000
D	33295	32455	29330	30555	25865	26550	25865	25300
E	33295	29330	29330	26820	25865	25865	25350	25350

Table III.50 BMP model with groups 25t6 - HRMP gap

u_{pt}	group size r							
variation	1	5	10	15	20	25	35	50
A	22.2%	24.0%	29.8%	28.5%	35.7%	36.7%	38.5%	29.0%
B	25.9%	26.7%	30.5%	33.1%	36.7%	37.1%	51.7%	31.6%
C	24.7%	26.5%	29.7%	22.2%	29.9%	39.9%	42.2%	30.1%
D	27.7%	29.5%	36.3%	33.6%	43.8%	42.3%	34.3%	21.6%
E	25.3%	34.2%	34.2%	39.8%	42.0%	42.0%	43.1%	34.3%

Table III.51 BMP model with groups 25t6 - LRMP execution time (sec)

u_{pt}	group size r							
variation	1	5	10	15	20	25	35	50
A	2.3	2.5	2.4	1.9	2.1	2.1	2.7	1.7
B	2.4	2.2	2.2	2.0	2.4	2.3	2.5	2.2
C	1.8	1.8	1.8	2.1	2.1	1.7	2.2	1.8
D	3.2	2.9	2.9	2.7	2.2	2.3	2.1	2.0
E	1.8	1.8	1.8	2.1	2.1	1.7	2.2	1.8

Table III.52 BMP model with groups 25t6 - HRMP execution time (sec)

u_{pt}	group size r							
variation	1	5	10	15	20	25	35	50
A	2.3	2.6	2.5	2.0	2.1	2.2	2.3	1.7
B	2.4	2.3	2.3	2.1	2.4	2.3	2.5	2.3
C	1.9	1.8	1.9	2.1	2.2	1.8	2.2	1.9
D	3.3	2.9	2.9	2.7	2.2	2.3	2.2	2.0
E	1.9	1.8	1.9	2.1	2.2	1.8	2.2	1.9

III.7 IMP model with groups

The following tables present the results that regards the IMP model with groups using dataset 25t6 and its five u_{pt} variations.

Table III.53 IMP model with groups 25t6 - LRMP objective function

u_{pt}	group size r							
variation	1	5	10	15	20	25	35	50
A	43389.0	43389.0	43389.0	43389.0	43389.0	43389.0	43389.0	43389.0
B	41989.0	41989.0	41989.0	41989.0	41989.0	41989.0	41989.0	41989.0
C	40552.5	40552.5	40552.5	40552.5	40552.5	40552.5	40552.5	40552.5
D	39355.0	39355.0	39355.0	39355.0	39355.0	39355.0	39355.0	39355.0
E	46042.0	46042.0	46042.0	46042.0	46042.0	46042.0	46042.0	46042.0

Table III.54 IMP model with groups 25t6 - LRMP gap

u_{pt}	group size r							
variation	1	5	10	15	20	25	35	50
A	6.0%	6.0%	6.0%	6.0%	6.0%	6.0%	6.0%	6.0%
B	9.0%	9.0%	9.0%	9.0%	9.0%	9.0%	9.0%	9.0%
C	12.1%	12.1%	12.1%	12.1%	12.1%	12.1%	12.1%	12.1%
D	14.7%	14.7%	14.7%	14.7%	14.7%	14.7%	14.7%	14.7%
E	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%

Table III.55 IMP model with groups 25t6 - IMP HRMP objective function

u_{pt}	group size r							
variation	1	5	10	15	20	25	35	50
A	43223	42375	39310	37395	34820	31400	27265	14150
B	41797	40875	38350	36555	34440	32050	25305	12750
C	40390	39785	37450	35445	34140	31050	26915	15250
D	39170	38590	35680	34365	31360	31525	20580	12800
E	45643	44605	41530	38895	37460	34800	24780	14350

Table III.56 IMP model with groups 25t6 - IMP HRMP gap

u_{pt}	group size r							
variation	1	5	10	15	20	25	35	50
A	0.4%	2.3%	9.4%	13.8%	19.7%	27.6%	37.2%	67.4%
B	0.5%	2.7%	8.7%	12.9%	18.0%	23.7%	39.7%	69.6%
C	0.4%	1.9%	7.7%	12.6%	15.8%	23.4%	33.6%	62.4%
D	0.5%	1.9%	9.3%	12.7%	20.3%	19.9%	47.7%	67.5%
E	0.9%	3.1%	9.8%	15.5%	18.6%	24.4%	46.2%	68.8%

Table III.57 IMP model with groups 25t6 - IMP LRMP execution time (sec)

u_{pt}	group size r							
variation	1	5	10	15	20	25	35	50
A	1.2	1.1	1.2	1.2	1.2	1.2	1.2	1.2
B	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
C	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
D	1.2	1.2	1.3	1.3	1.2	1.2	1.2	1.2
E	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1

Table III.58 IMP model with groups 25t6 - IMP HRMP execution time (sec)

u_{pt}	group size r							
variation	1	5	10	15	20	25	35	50
A	1.2	1.2	1.3	1.2	1.2	1.2	1.2	1.2
B	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
C	1.2	1.5	1.4	1.4	1.4	1.5	1.4	1.4
D	1.2	1.2	1.3	1.4	1.3	1.2	1.2	1.2
E	1.1	1.1	1.1	1.1	1.2	1.1	1.1	1.1