Approximate Linear Programming in Network Revenue Management with Multiple Modes

David Sayah[a]

[a] *Chair of Logistics Management, Johannes Gutenberg University Mainz, Jakob-Welder-Weg 9, D-55128 Mainz, Germany.*

Abstract

Approximate linear programming has been applied to network revenue management problems under the fundamental modeling assumption that products define combinations of one resource bundle and a fare class. We consider products that can have multiple operational modes allowing companies to select the way they want to serve the purchaser of a multi-mode product. We show that the presence of multi-mode products implies a weaker relation between an affine approximate linear program (ALP) and a compact reformulation, known as reduction. Consequently, the upper bound on the maximum expected revenue obtained via the reduction is not necessarily as tight as the upper bound produced via the ALP. We further demonstrate that the gap between these two formulations is bounded in general and zero in a particular class of instances, when multi-mode products are flexible products. For general instances, we exploit a set-packing structure within the reduction in order to improve the upper bound, i.e., we introduce a cutting plane method that strengthens the reduction by separating valid inequalities. Our computational tests indicate that it is possible to halve the gap in not more than 4% of the time needed to solve the ALP via column generation.

Key words: affine approximation, network revenue management, multiple operational modes, transportation networks

1. Introduction and Literature Review

Revenue management problems arise in various industries like passenger and cargo transportation. These companies typically sell products in different fare classes to heterogeneous customer segments with random demands subject to fixed inventory capacities. They face the problem to decide dynamically when to sell which product so that the expected revenue over the entire planning horizon is maximized. If a product consumes capacity of multiple resources, this problem is known as network revenue management problem. The underlying optimization task can be formulated as a dynamic program which is intractable due to the number of possible states growing exponentially with the number of resources. A standard heuristic approach to network revenue management is to solve simpler problems which approximate the value function of the dynamic program. Solutions to these simpler problems are typically used in order to derive bid price controls, see Talluri and van Ryzin (1998) and for other network controls Talluri and van Ryzin (2004).

Email address: sayah@uni-mainz.de (David Sayah)

Deterministic linear programming is a classical approach to network revenue management (Williamson, 1992). Though computationally appealing, the upper bound on the maximum expected revenue produced by the deterministic linear program (DLP) is relatively weak as it neglects the dynamics of the revenue management problem. It is broadly agreed that the bound obtained via a given approximation method is a measure of its accuracy and that it indicates the performance of the policy derived from that approximation. Upper bounds are also useful in order to assess the quality of any other heuristic approach to network revenue management. For these reasons, recent research in this field has a focus on the development of methods that produce as tight upper bounds as possible.

In this respect, most promising approaches are commonly labelled as approximate dynamic programming. The linear programming approach to approximate dynamic programming transforms the original high-dimensional dynamic program into a linear program of manageable size (e.g., Schweitzer and Seidmann, 1985; Powell, 2011; Bertsekas, 2012). This transformation is achieved using functional approximations. The resulting linear program is usually referred to as approximate linear program (ALP). In network revenue management, Adelman (2007) initially proposed a functional approximation that is linear and separable over the resources. This is called affine approximation and produces dynamic bid prices which are superior to the static bid prices obtained via a DLP. Farias and van Roy (2007) and Meissner and Strauss (2012) refined Adelman's idea using functional forms that are separable over the resources and inventory levels. This is known as separable piecewise-linear approximation and delivers dynamic and capacity-dependent bid prices. Both affine and separable piecewise-linear approximations have been applied to choice-based network revenue management (Zhang and Adelman, 2009; Meissner and Strauss, 2012).

The solvability of ALPs is an important issue. For instance, the size of a separable piecewise-linear ALP is considerably larger than the size of an affine ALP, i.e., the improved bid prices are computationally much more expensive than the dynamic bid prices. Lagrangean relaxation (Topaloglu, 2009; Kunnumkal and Topaloglu, 2010) is a competing method to generate the improved bid prices. Kunnumkal and Talluri (2014) have recently established that the upper bound obtained by Lagrangean-based decomposition equals that of a separable piecewise-linear approximation in network revenue management. Consequently, one can resort to subgradient optimization if tackling large-scale ALPs is too hard in practice.

Recent findings, however, have drawn the attention back to approximate linear programming. New compact reformulations of ALPs, known as reductions, have been developed for the network revenue management settings with and without choice (Vossen and Zhang, 2013; Tong and Topaloglu, 2014). A reduction is compact in the sense that both the number of constraints and the number of variables grow only linearly with certain problem parameters. A second important feature of reductions in these settings is that they achieve the same upper bound as the ALPs while being typically compact enough for an off-the-shelf linear programming solver.

In contrast to existing papers, we relax the assumption that a product is defined by a fare class and one particular resource bundle because it can be restrictive in some industries. We consider industries in which products are designed in such a way that the company needs to select the way of serving their customers. Products that allow for additional flexibility are not entirely new in revenue management. The concept of flexible products has been introduced by Gallego and Phillips (2004). The authors report applications across a wide range of industries, e.g., internet advertising, tour operators, multiple property management, and air cargo. Similar concepts are discussed in, e.g., Talluri (2001), Chen *et al.* (2003), Post (2010), Chen *et al.* (2010), and Petrick *et al.* (2010).

We use the term operational mode as a synonym for resource bundle and distinguish between products with exactly one mode and products with two or more modes. Single-mode products are, for instance, the ordinary itinerary-fare class combinations offered by airlines.

To give an example of a multi-mode product, note that the demand for freight transportation and air cargo is typically not itinerary-specific. These customers rather specify an earliest pick-up and a latest delivery date for the transport of their goods from an origin to a destination. Forwarders may then have a choice between different feasible routes through their transportation networks departing/arriving at different times. Consider a scheduled route between a given origin-destination pair as an operational mode. If a customer orders pick-up and delivery within a certain time window, then the collection of routes which guarantees on-time service for this customer can be seen as a multi-mode product.

Therefore, the basic decision problem in network revenue management with multiple modes is to dynamically accept/reject customer requests and if necessary to assign accepted requests to operational modes.

Existing literature does not answer the question how these more general network revenue management terms affect approximate linear programming. Our paper fills this gap by making the following particular contributions: We present an affine ALP and a reduction for the network revenue management problem with multiple modes. Subsequently, the relationship between the two models is analyzed in the light of the Dantzig-Wolfe decomposition principle for integer programs. From this point of view, we explain why the upper bound obtained via the reduction is not necessarily as tight as the one obtained via the ALP. We provide an example showing that multi-mode products can cause an error in terms of accuracy, i.e., a gap between ALP and reduction. This gap is further shown to be bounded in the sense that the upper bound can never be worse than the upper bound associated with the DLP. For instances in which all multi-mode products are flexible products, we give a guarantee that the gap is zero.

Moreover, we delve into the general scenario without additional restrictions regarding the design of multi-mode products. Since column generation may converge too slow in such instances, a method that can efficiently strengthen the compact formulation is an option. To this end, we devise a new cutting plane method that exploits a set-packing structure which we identify within the reduction. This result grants access to a variety of existing valid inequalities that can be used to shrink the gap between ALP and reduction. We describe one particular implementation based on the separation of odd-cycle cuts.

Finally, a computational study provides insights as to the quality and the computation times of the different approximation methods, the gaps between ALP and reduction, and the effectiveness of the odd-cycle cuts. Our experiments are based on a set of test instances that include the data previously used in the literature. Our numerical results indicate that the reduction can be strengthened quite effectively in a reasonable amount of time.

The remainder of this paper is organized as follows: Section 2 introduces the notation and assumptions of our multi-mode network revenue management model. In Section 3, we describe the DLP, the affine ALP, and the reduction. The relationship between ALP and reduction is examined in Section 4. We discuss the impact of a restricted product design on this relationship in Section 5. In Section 6, we develop the cutting-plane method. The numerical results are presented in Section 7. Final conclusions are drawn in Section 8.

2. Problem Formulation

We consider a company that operates a given set of resources $\mathcal{I} = \{1, \ldots, m\}$ with initial capacities $c = \{c_i \in \mathbb{Z}_+ : i \in \mathcal{I}\}$. Let $\mathcal{M} = \{1, \ldots, n^{\text{o}}\}$ be the set of all operational modes that are available to the company. Each mode $o \in \mathcal{M}$ specifies a subset of resources $\mathcal{I}_o \subseteq \mathcal{I}$ that can be used to satisfy demand. A binary $m \times n^{\text{o}}$ incidence matrix $(a_{io})_{i \in \mathcal{I}, o \in \mathcal{M}}$ is used to represent the resource requirements of the modes. Each entry a_{io} takes value one if mode o consumes one unit of resource i and zero otherwise.

The company sells a set of products $\mathcal{J} = \{1, \ldots, n\}$ to n heterogeneous customer segments. Each product $j \in \mathcal{J}$ is associated with a subset $\mathcal{M}_j \subseteq \mathcal{M}$ and a fare f_j.[1] We refer to the set of all product-mode combinations as $\mathcal{J} \times \mathcal{M} = \{(j, o) : j \in \mathcal{J}, o \in \mathcal{M}_j\}$. Using this notation, we distinguish between two types of products. If $|\mathcal{M}_j| = 1$ for any product $j \in \mathcal{J}$, it is a *single-mode* product, otherwise it is a *multi-mode* product with $|\mathcal{M}_j| > 1$. We highlight that any subset of the existing modes can define a multi-mode product.

The company controls the sales of products over a finite time horizon $\mathcal{T} = \{1, \ldots, \tau\}$. The standard single-request demand model without choice behavior is assumed, i.e., we are given independent probabilities $\{p_{jt} : j \in \mathcal{J}, t \in \mathcal{T}\}$ and p_{jt} denotes the probability of a request for product j in period t. We assume that $\sum_{j \in \mathcal{J}} p_{jt} = 1$ for all $t \in \mathcal{T}$ without loss of generality. If the "zero-demand" event occurs with positive probability in period t, i.e., if $\sum_{j \in \mathcal{J}} p_{jt} < 1$, we can add a dummy product $j = 0$ in this period with zero capacity consumption, zero revenue, and probability $p_{0t} = 1 - \sum_{j \in \mathcal{J}} p_{jt}$.

We assume that c is the initial state of the network in the first period $t = 1$. In any later period $t \geq 2$, the residual capacity of resource $i \in \mathcal{I}$ can be less than c_i but not negative. Let the m-vector $x_t = (x_{it})$ describe the current state of the network for $t \in \mathcal{T}$. The product set $\mathcal{X}_t = \Pi_{i \in \mathcal{I}} \mathcal{X}_{it}$ denotes the state space in t, where $\mathcal{X}_{it}$ is the domain of x_{it} defined as

$$\mathcal{X}_{it} = \begin{cases} \{c_i\} & \text{if } t = 1 \\ \{0, \ldots, c_i\} & \text{if } t \geq 2 \end{cases} \quad \forall t \in \mathcal{T}, i \in \mathcal{I}.$$

It is further assumed that there must not be a time lag between the acceptance and the assignment of a customer request but, of course, the notification of the customer about the company's final decision may be delayed. Let the binary variables $u_t = \{u_{jot} : (j, o) \in \mathcal{J} \times \mathcal{M}\}$ indicate if we accept a request for product j *and* assign it to mode o. If $u_{jot} = 1$, we say more frequently that the product-mode combination (j, o) is offered in period t, even though the chosen mode o is clearly not presented to the customers before the purchase.

Let $V_t(x_t)$ denote the maximum expected revenue from period t onwards given that the network is currently in state x_t. Using the ith unit vector e_i with value one at position $i \in \mathcal{I}$ and zero elsewhere, we define the network revenue management problem with multiple modes (MNRM) as

[1] Assuming a given fixed cost c_o of using mode o, we can define product- and mode-specific net fares $f_{jo} = f_j - c_o$ for each $j \in \mathcal{J}$ and $(j, o) \in \mathcal{M}_j$. Then, our subsequent analysis still applies but we stick to the product-specific fares for ease of exposition.

the dynamic program

$$V_t(x_t) = \max \sum_{j\in\mathcal{J}} p_{jt}\left\{ f_j \sum_{o\in\mathcal{M}_j} u_{jot} + V_{t+1}(x_t - \sum_{o\in\mathcal{M}_j} u_{jot} \sum_{i\in\mathcal{I}} e_i a_{io})\right\} \tag{1a}$$

$$\text{s.t. } a_{io}u_{jot} \le x_{it} \quad \forall (j,o) \in \mathcal{J}\times\mathcal{M}, i \in \mathcal{I} \tag{1b}$$

$$\sum_{o\in\mathcal{M}_j} u_{jot} \le 1 \quad \forall j \in \mathcal{J} \tag{1c}$$

$$u_t \text{ binary} \tag{1d}$$

for all $t \in \mathcal{T}, x_t \in \mathcal{X}_t$ with boundary conditions defined by $V_{\tau+1}(\cdot) = 0$. Constraints (1b) ensure that we can choose a mode o to serve a product j customer only if we respect the current residual capacity x_{it} of all resources $i \in \mathcal{I}$ used by that mode. By (1c), a unique mode must be chosen if a request is accepted. The domain of the action variables u_{jot} is declared in (1d). The set of actions that are feasible in a given state x_t is denoted by $\mathcal{U}_t(x_t) = \{u_t : (1b)\text{–}(1d)\}$, and the set of feasible state-action pairs in $t \in \mathcal{T}$ by $S_t = \{(x_t, u_t) : x_t \in \mathcal{X}_t, u_t \in \mathcal{U}_t(x_t)\}$.

Let $O = \max_{j\in\mathcal{J}} |\mathcal{M}_j|$ denote the cardinality of a maximum mode set. Instances with $O = 1$ are referred to as NRM instances since model (1) then obviously reduces to the usual network revenue management problem. Instances of (1) with $O > 1$ are referred to as MNRM instances. Note that our model can be seen as an online version of the network revenue management model with flexible products suggested by Gallego *et al.* (2004) but we, in contrast, do not require additional assumptions regarding the product design.

We denote the maximum expected revenue $V_1(c)$ defined in (1) by V^{DP}. Solving the dynamic program for V^{DP} is intractable due to exponentially growing numbers of states and actions. This paper focuses on methods that approximate the value function $\{V_t(x_t) : t \in \mathcal{T}, x_t \in \mathcal{X}_t\}$.

3. Deterministic and Approximate Linear Programming Bounds

3.1. The DLP Bound

The deterministic linear programming approach is straightforward when products can have multiple modes. Define continuous variables $w = \{w_{jo} : (j,o) \in \mathcal{J}\times\mathcal{M}\}$ as the number w_{jo} of accepted requests for product j which we assign to mode o. Then, the deterministic linear program (DLP) is given by

$$V^{\mathrm{DLP}} = \max \sum_{(j,o)} f_j w_{jo} \tag{2a}$$

$$\text{s.t. } \sum_{(j,o)} a_{io} w_{jo} \le c_i \quad \forall i \in \mathcal{I} \tag{2b}$$

$$\sum_{o\in\mathcal{M}_j} w_{jo} \le \sum_{t\in\mathcal{T}} p_{jt} \quad \forall j \in \mathcal{J} \tag{2c}$$

$$w \ge 0. \tag{2d}$$

Constraints (2b) say that the initial capacity of all resources must not be exceeded. We cannot accept more than the given expected number of requests for any product because of (2c). The dual prices associated with constraints (2b) can be viewed as static value function approximation.

It means that the ith dual price is an estimate of the expected value of one unit of resource i's capacity at the beginning of the time horizon. The DLP model of Gallego *et al.* (2004) is a special instance in our DLP model. We show $V^{\mathrm{DLP}} \geq V^{\mathrm{DP}}$ in Section 4.

3.2. The affine ALP Bound

Approximate linear programming departs from the equivalent linear programming formulation (Puterman, 1994) of the dynamic program (1):

$$V^{\mathrm{DP}} = \min_{v} v_1(c) \tag{3a}$$

$$\text{s.t. } v_t(x_t) \geq \sum_{j\in\mathcal{J}} p_{jt}\left\{ f_j \sum_{o\in\mathcal{M}_j} u_{jot} + v_{t+1}(x_t - \sum_{o\in\mathcal{M}_j} u_{jot} \sum_{i\in\mathcal{I}} e_i a_{io}) \right\}$$

$$\forall t \in \mathcal{T}, (x_t, u_t) \in S_t. \tag{3b}$$

Here, we have real-valued decision variables $v = \{v_t(x_t) : t \in \mathcal{T}, x_t \in \mathcal{X}_t\}$ and we stipulate that $v_{\tau+1}(\cdot) = 0$. The linear program (3) is clearly as intractable as the dynamic program (1). Thus, we specify the functional approximation

$$v_t(x_t) \approx \sum_{i\in\mathcal{I}} x_{it}\vartheta_{it} \quad \forall t \in \mathcal{T}, x_t \in \mathcal{X}_t \tag{4}$$

with real-valued weights $\vartheta = \{\vartheta_{it} : i \in \mathcal{I}, t \in \mathcal{T}\}$. Plugging the form (4) into (3) and defining $\vartheta_{i,\tau+1} = 0, i \in \mathcal{I}$, we reach after some rearranging the affine approximate linear program (ALP)

$$V^{\mathrm{AF}} = \min_{\vartheta} \sum_{i\in\mathcal{I}} c_i\vartheta_{i1} \tag{5a}$$

$$\text{s.t. } \sum_{i\in\mathcal{I}} \left\{ (\vartheta_{it} - \vartheta_{i,t+1})x_{it} + \sum_{(j,o)} p_{jt}u_{jot}a_{io}\vartheta_{i,t+1} \right\} \geq \sum_{(j,o)} p_{jt}f_j u_{jot}$$

$$\forall t \in \mathcal{T}, (x_t, u_t) \in S_t, \tag{5b}$$

where $\sum_{(j,o)}$ is the sum over all product-mode combinations $(j,o) \in \mathcal{J}\times\mathcal{M}$. Likewise, we use $\sum_{(x,u)}$ to denote the sum over all $(x,u) = (x_t, u_t) \in S_t$, where the period t to which these state-action pairs refer is always clear from the context.

Now, model (5) has $m\tau$ variables but the number of constraints still grows exponentially with m and $|\mathcal{J}\times\mathcal{M}|$. We solve the dual problem of (5) given by

$$V^{\mathrm{AF}} = \max \sum_{t\in\mathcal{T}} \sum_{(x,u)} \Big(\sum_{(j,o)} p_{jt}f_j u_{jot} \Big) X^t_{xu} \tag{6a}$$

$$\text{s.t. } \sum_{(x,u)} x_{it} X^t_{xu} = \begin{cases} c_i & \text{if } t = 1 \\ \sum_{(x,u)} \Big(x_{i,t-1} - \sum_{(j,o)} p_{j,t-1}a_{io}u_{jo,t-1} \Big) X^{t-1}_{xu} & \text{if } t \geq 2 \end{cases}$$

$$\forall i \in \mathcal{I}, t \in \mathcal{T} \tag{6b}$$

$$X \geq 0. \tag{6c}$$

The continuous variables $X = \{X^t_{xu} : t \in \mathcal{T}, (x,u) \in S_t\}$ are associated with constraints (5b). The number of columns now increases exponentially in m and $|\mathcal{J} \times \mathcal{M}|$. This is why column generation is commonly used to compute V^{AF}. This amounts to decomposing the model (6) into a restricted master problem (RMP) and a pricing problem. Initially, the RMP contains only a small subset of columns, e.g., those representing the "do-nothing" policy in (6). Given the dual solution ϑ to the current RMP, new columns are priced out with the integer program

$$\zeta_t^{\mathrm{AFP}} = \max_{(x_t,u_t)\in S_t} \sum_{(j,o)} p_{jt}\Big(f_j - \sum_{i\in\mathcal{I}} a_{io}\vartheta_{i,t+1}\Big)u_{jot} - \sum_{i\in\mathcal{I}}(\vartheta_{it} - \vartheta_{i,t+1})x_{it} \quad \forall t \in \mathcal{T}. \tag{7}$$

We repeatedly generate new columns, add them to the RMP, and reoptimize it until no column with positive reduced profit ζ_t^{AFP} exists.

Note that we choose the functional form (4) instead of the more frequently used form $v_t(x_t) \approx \theta_t + \sum_{i\in\mathcal{I}} x_{it}\vartheta_{it}$ to ease our exposition. This choice is without loss of generality because if we plug the functional form with additional θ_t variables into the formulation (3) and if we further dualize the resulting ALP, we arrive at a problem defined by model (6) and the additional constraints

$$\sum_{(x,u)} X^t_{xu} = 1 \quad \forall t \in \mathcal{T}, \tag{8}$$

which do not affect the value V^{AF}. This is the result of our first lemma.

Lemma 1. *If $V^{AF\text{-}CONV}$ denotes the optimal value of the problem defined by model* (6) *and the convexity constraints* (8)*, then the equality $V^{AF\text{-}CONV} = V^{AF}$ holds.*

Proof. See Appendix A. □

We refer to (8) as *convexity constraints* and it turns out that Lemma 1 is useful again in Section 4. Moreover, Adelman (2007) pointed out that the convexity constraints suggest to interpret X^t_{xu} as an estimate of the probability of being in state x_t at time t and taking the action u_t. The optimization task defined by (6) is then to find a distribution of approximate state-action probabilities that maximizes the total expected revenue in (6a).

Observe that any feasible solution to the ALP (5) implies a feasible solution to the equivalent linear programming formulation (3) by the substitution (4). Thus, $V^{\mathrm{AF}} \geq V^{\mathrm{DP}}$ holds.

3.3. The Reduction-Based Bound

Consider the linear program

$$V^{\mathrm{AFR}} = \max \sum_{t\in\mathcal{T}} \sum_{(j,o)} p_{jt} f_j q_{jot} \tag{9a}$$

$$\text{s.t. } y_{it} = \begin{cases} c_i & \text{if } t = 1 \\ y_{i,t-1} - \sum_{(j,o)} p_{j,t-1} a_{io} q_{jo,t-1} & \text{if } t \geq 2 \end{cases} \quad \forall t \in \mathcal{T}, i \in \mathcal{I} \tag{9b}$$

$$a_{io} q_{jot} \leq y_{it} \quad \forall t \in \mathcal{T}, (j,o) \in \mathcal{J} \times \mathcal{M}, i \in \mathcal{I} \tag{9c}$$

$$\sum_{o\in\mathcal{M}_j} q_{jot} \leq 1 \quad \forall t \in \mathcal{T}, j \in \mathcal{J} \tag{9d}$$

$$0 \leq q_t \leq 1, y_t \geq 0 \quad \forall t \in \mathcal{T}. \tag{9e}$$

Similar to the NRM case (Tong and Topaloglu, 2014), the continuous variables $q_t = \{q_{jot} : (j,o) \in \mathcal{J} \times \mathcal{M}\}$ and $y_t = \{y_{it} : i \in \mathcal{I}\}$ have a probabilistic interpretation: y_{it} reads as the expected amount of resource i's capacity available in t and q_{jot} reads as the expected number of accepted requests for product j that we assign to mode o in t. Then, the objective function (9a) maximizes the total expected revenue. The flow-balance constraints (9b) ensure for every $t \in \mathcal{T}$ and $i \in \mathcal{I}$ that the expected amount of resource i's capacity when entering period t equals the expected amount of resource i's capacity when entering the previous period $t-1$ minus resource i's expected capacity consumption in $t-1$. Constraints (9c) and (9d) are the probabilistic counterparts to constraints (1b) and (1c), respectively. The domains of the variables are defined in (9e).

The formulation (9) is fairly compact since the numbers of variables and constraints increase linearly in m, τ, and $|\mathcal{J} \times \mathcal{M}|$. It is referred to as *reduction* and we analyze next how it relates to the ALP (6). The fact that $V^{\text{AFR}} \geq V^{\text{DP}}$ is a consequence of this relation.

4. Ties Between ALP and Reduction

4.1. Nonintegrality of the Pricing Problem

For the settings with and without customer choice but without multiple modes, Tong and Topaloglu (2014) and Vossen and Zhang (2013) establish the equivalence between an affine ALP and its corresponding reduction by exploiting the integrality property of the pricing problem. We begin with an example that demonstrates the nonintegrality of the pricing problem in our setting. Using $\mathbb{1}_{\{\cdot\}}$ to denote the indicator function, model (7) of the pricing problem can be rewritten as

$$\zeta_t^{\text{AFP}} = \max \sum_{(j,o)} p_{jt}\Big(f_j - \sum_{i \in \mathcal{I}} \vartheta_{i,t+1} a_{io}\Big) u_{jot} - \sum_{i \in \mathcal{I}} (\vartheta_{it} - \vartheta_{i,t+1}) x_{it} \tag{10a}$$

$$\text{s.t. } a_{io} u_{jot} \leq x_{it} \quad \forall (j,o) \in \mathcal{J} \times \mathcal{M}, i \in \mathcal{I} \tag{10b}$$

$$\sum_{o \in \mathcal{M}_j} u_{jot} \leq 1 \quad \forall j \in \mathcal{J} \tag{10c}$$

$$\mathbb{1}_{\{t=1\}} c \leq x_t \leq c, u_t \geq 0 \tag{10d}$$

$$x_t, u_t \text{ integer.} \tag{10e}$$

Constraints (10b) and (10c) are identical to the dynamic program (1). (10a) defines the same objective of the pricing problem as (7). The variable domains are stated in (10d) and (10e).

Let (AFP-L) refer to the linear relaxation of (10) and let $\zeta_t^{\text{AFP-L}}$ denote the associated optimal objective function value. The following counterexample shows that the integrality gap can be positive, i.e., $\zeta_t^{\text{AFP-L}} - \zeta_t^{\text{AFP}} \geq 0$.

Example 1. Suppose that we are in any $t \in \mathcal{T}$. We assume three resources $i \in \{1,2,3\}$ with one unit of capacity each, three products $j \in \{a,b,c\}$ priced at \$10 and with arrival probability 0.1 each, and three operational modes consuming the resources as given by the matrix (a_{io}) below. The product-specific mode sets are assumed to be defined by $\mathcal{M}_a = \{1,2\}, \mathcal{M}_b = \{2,3\}$, and $\mathcal{M}_c = \{1,3\}$. The dual prices are given by the values $\vartheta_{1t}, \ldots, \vartheta_{3,t+1}$ below. An optimal solution to the integer program (10) has the non-zero variables $x_{2t} = x_{3t} = 1$ and $u_{a2t} = u_{b2t} = u_{c3t} = 1$, and it is valued with a maximum reduced profit of $\zeta_t^{\text{AFP}} = 2.73$. However, if we set all variables equal

to 1/2 , this solves the linear relaxation (AFP-L) with $\zeta_t^{\text{AFP-L}} = 2.76$.

$$(a_{io}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \vartheta_{1t} & \vartheta_{1,t+1} \\ \vartheta_{2t} & \vartheta_{2,t+1} \\ \vartheta_{3t} & \vartheta_{3,t+1} \end{pmatrix} = \begin{pmatrix} .5 & .4 \\ .3 & .2 \\ .4 & .3 \end{pmatrix}.$$

4.2. Relationship Between ALP and Reduction

Unlike Vossen and Zhang (2013) and Tong and Topaloglu (2014), we apply arguments from Dantzig-Wolfe decomposition for *integer* programs to analyze the relationship. This point of view has two major uses: First, we can link the pair of linear programs (6) and (9) with a corresponding pair of integer programs. This correspondence leads us to the conclusion that the tie between ALP and reduction is weak in general but can be strong in certain instances. Second, this perspective suggests the way we derive valid inequalities for problem (9) in Section 6.

The first integer program is given by

$$V^{\text{IAF}} = \max \sum_{t\in\mathcal{T}} \sum_{(x,u)} \Big(\sum_{(j,o)} p_{jt} f_j u_{jot} \Big) X_{xu}^t \tag{11a}$$

s.t. (6b)

$$\sum_{(x,u)} X_{xu}^t = 1 \quad \forall t \in \mathcal{T} \tag{11b}$$

$$y_{it} = \sum_{(x,u)} x_{it} X_{xu}^t \quad \forall t \in \mathcal{T}, i \in \mathcal{I} \tag{11c}$$

$$q_{jot} = \sum_{(x,u)} u_{jot} X_{xu}^t \quad \forall t \in \mathcal{T}, (j,o) \in \mathcal{J} \times \mathcal{M} \tag{11d}$$

$$X \geq 0 \tag{11e}$$

$$q_t, y_t \text{ integer} \quad \forall t \in \mathcal{T}. \tag{11f}$$

We refer to this formulation as affine approximate (mixed) integer program (AIP). There are as many as $\tau(|\mathcal{J} \times \mathcal{M}| + m)$ integer variables $q_t = \{q_{jot} : (j,o) \in \mathcal{J} \times \mathcal{M}\}$ and $y_t = \{y_{it} : i \in \mathcal{I}\}$. The continuous variables X are identical to the ALP (6).

Lemma 2. *The ALP* (6) *is the linear relaxation of the AIP* (11)*, i.e.,* $V^{IAF} \leq V^{AF}$ *holds.*

Proof. Obviously, constraints (11c) and (11d) are redundant in the linear relaxation of problem (11). Noting that Lemma 1 applies completes the proof. □

The second integer program takes the form

$$V^{\text{IAFR}} = \max \sum_{t\in\mathcal{T}} \sum_{(j,o)} p_{jt} f_j q_{jot} \tag{12a}$$

$$\text{s.t. } y_{it} = \begin{cases} c_i & \text{if } t = 1 \\ y_{i,t-1} - \sum_{(j,o)} p_{j,t-1} a_{io} q_{jo,t-1} & \text{if } t \geq 2 \end{cases} \quad \forall t \in \mathcal{T}, i \in \mathcal{I} \tag{12b}$$

$$(y_t, q_t) \in \mathcal{P}_t \quad \forall t \in \mathcal{T}, \tag{12c}$$

where we define the structure in (12c) as

$$\mathcal{P}_t = \{(y_t, q_t) : (9c) - (9e) \text{ and integer}\} \quad \forall t \in \mathcal{T}.$$

The formulation (12) is as compact as the reduction (9). Since the integrality constraints defined in (12c) are the only difference between these two compact formulations, this proves the next lemma.

Lemma 3. *The reduction* (9) *is the linear relaxation of the compact integer program* (9)*, i.e.,* $V^{IAFR} \leq V^{AFR}$ *holds.*

Let us consider the equivalent Dantzig-Wolfe reformulation of the compact formulation (12). For $t \in \mathcal{T}$, let $E_t = \{1, \ldots, |E_t|\}$ denote the finite index set of extreme points (y_t^p, q_t^p) of the convex hull $\text{conv}(\mathcal{P}_t)$. Using continuous variables $\lambda_t = \{\lambda_{pt} : t \in \mathcal{T}, p \in E_t\}$ for all $t \in \mathcal{T}$, the Dantzig-Wolfe reformulation of (12) can be written as:

$$V^{\text{IAFR}} = \max \sum_{t \in \mathcal{T}} \sum_{p \in E_t} \left[\sum_{(j,o)} p_{jt} f_j q_{jot}^p \right] \lambda_{pt} \tag{13a}$$

$$\text{s.t.} \sum_{p \in E_t} y_{it}^p \lambda_{pt} = \begin{cases} c_i & \text{if } t = 1 \\ \sum_{p \in E_{t-1}} \left[y_{i,t-1}^p - \sum_{(j,o)} p_{j,t-1} a_{io} q_{jo,t-1}^p \right] \lambda_{p,t-1} & \text{if } t \geq 2 \end{cases}$$

$$\forall i \in \mathcal{I}, t \in \mathcal{T} \tag{13b}$$

$$\sum_{p \in E_t} \lambda_{pt} = 1 \quad \forall t \in \mathcal{T} \tag{13c}$$

$$y_{it} = \sum_{p \in E_t} y_{it}^p \lambda_{pt} \quad \forall t \in \mathcal{T}, i \in \mathcal{I} \tag{13d}$$

$$q_{jot} = \sum_{p \in E_t} q_{jot}^p \lambda_{pt} \quad \forall t \in \mathcal{T}, j \in \mathcal{J}, o \in \mathcal{M}_j \tag{13e}$$

$$\lambda_t \geq 0 \quad \forall t \in \mathcal{T} \tag{13f}$$

$$q_t, y_t \text{ integer} \quad \forall t \in \mathcal{T}. \tag{13g}$$

Lemma 4. *The AIP* (11) *is equivalent to the integer reduction* (12)*, i.e.,* $V^{IAF} = V^{IAFR}$.

Proof. Suppose that any optimal solution $(\hat{X}, \hat{y}, \hat{q})$ to formulation (11) is given. We first show that $(\hat{X}, \hat{y}, \hat{q})$ is feasible in the formulation (12) and valued with the same objective function value. Define a solution $(y, q) = \{(y_t, q_t) : t \in \mathcal{T}\}$ according to

$$y_{it} = \hat{y}_{it} \quad \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{14}$$

$$q_{jot} = \hat{q}_{jot} \quad \forall j \in \mathcal{J}, o \in \mathcal{M}_j, t \in \mathcal{T}. \tag{15}$$

The objective function (12a) follows from substituting for X in (11a) using (11c). Constraints (12b) follow from plugging (11d) into (6b). Constraints (11b), (11c), and (11d) imply that $(\hat{y}_t, \hat{q}_t)$ is a convex combination of feasible state-action pairs $(x_t, u_t) \in S_t$. By (11f), this convex combination must be integral, and so $(\hat{y}, \hat{q}) \in S_t$. Noting that $\mathcal{P}_t$ is identical to S_t, it follows that $(\hat{y}_t, \hat{q}_t) \in \mathcal{P}_t$ for all $t \in \mathcal{T}$. Hence, (y, q) is consistent with constraints (12c), which shows $V^{\text{IAF}} \leq V^{\text{IAFR}}$.

We now interpose the equivalent Dantzig-Wolfe reformulation of the compact formulation (12) to show the reverse direction, $V^{\text{IAFR}} \leq V^{\text{IAF}}$. That means we take any feasible solution $(\hat{\lambda}, \hat{y}, \hat{q})$

to the extensive formulation (13) as given and then we derive a solution (X, y, q) which is feasible and equally valued in the extensive formulation (11). Define (y, q) just as in (14) and (15). Noting the definition of the extreme points (y_t^p, q_t^p) and the equality $\mathcal{P}_t = S_t$, it follows that $(y_t^p, q_t^p) \in S_t$ for all $t \in \mathcal{T}, p \in E_t$. Therefore, we define $X_{xu}^t = \hat{\lambda}_{pt}$ for each $(x, u) \in S_t, t \in \mathcal{T}$ if there is a $p \in E_t$ with $(x, u) = (y_t^p, q_t^p)$, and we set $X_{xu}^t = 0$ otherwise.

Next, we check the feasibility of (X, y, q). X is nonnegative and (y, q) is integral by construction. Plugging (13d) into (13b) yields the same constraints as plugging (11c) into (6b). To see that (y, q) fulfills constraints (11c), note that

$$y_{it} = \hat{y}_{it} = \sum_{p \in E_t} y_{it}^p \hat{\lambda}_{pt} = \sum_{(x,u)} x_{it} X_{xu}^t \quad \forall t \in \mathcal{T}, i \in \mathcal{I},$$

where the first equality is the definition of y, the second holds because $(\hat{\lambda}, \hat{y})$ satisfies (13d), and the last equality is true by the definition of X. Likewise, the consistency with constraints (11d) can be verified. Thus, (X, y, q) is a feasible solution to the model (11). Moreover, if we substitute for λ in (13a) using (13e), the resulting objective function equals the objective function that is obtained by substituting for X in (11a) using (11d), which completes the proof. □

The above lemma implies that the AIP (11) provides next to the Dantzig-Wolfe reformulation (13) a second option of writing down the compact model (12) in an extensive way. The AIP contains a continuous variable X_{xu}^t for *every* feasible state-action pair. In the formulation (13), however, a continuous variable λ_{pt} is present only if a feasible state-action pair represents an extreme point of the convex hull of $\mathcal{P}_t$.

Lemmata 2, 3, and 4 lead to the main conclusion of this subsection.

Corollary 1. *For the ALP* (6) *and the reduction* (9)*, the inequality* $V^{AF} \leq V^{AFR}$ *holds.*

The equality $V^{\mathrm{AF}} = V^{\mathrm{AFR}}$ in Corollary 1 holds if the formulation of the pricing problem has no integrality gap (see also Lübbecke and Desrosiers, 2005). The strict inequality holds whenever the integrality gap is positive because in this case feasible solutions to the reduction may be fractional extreme points of the linear relaxed pricing problem. A column generation algorithm, in contrast, will price out only integral state-action pairs since the integrality constraints are kept in the pricing problem. As a result, the values V^{AFR} and V^{IAF} are farther apart than V^{AF} and V^{IAF}. We emphasize that Vossen and Zhang (2013) understand the reduction (9) as an aggregation, hence as a relaxation, of (6) which is a different argument to show the inequality of Corollary 1.

4.3. Final Assessment of Upper Bounds

Before we come to a final comparison of the upper bounds presented so far, we elaborate on the case where the strict inequality of Corollary 1 holds. More specifically, we analyze the question if the reduction-based upper bound can be arbitrarily worse than the ALP bound. Our answer is that the reduction can never perform worse than the DLP.

Lemma 5. *For the reduction* (9) *and the DLP* (2)*, the inequality* $V^{AFR} \leq V^{DLP}$ *holds.*

Proof. Suppose that we are given any feasible solution (q, y) to problem (9). Define a solution $w = \{w_{jo} : j \in \mathcal{J}, o \in \mathcal{M}_j\}$ to problem (2) according to

$$w_{jo} = \sum_{t \in \mathcal{T}} p_{jt} q_{jot} \qquad \forall j \in \mathcal{J}, o \in \mathcal{M}_j.$$

Plugging the definition of w into (9a) yields the objective function (2a). The demand constraints (2c) follow directly from the definition of w and the fact that $0 \leq q \leq 1$. To see that w satisfies the capacity constraints (2b), we aggregate constraints (9b) over t and rearrange terms to obtain

$$c_i = \sum_{t=1}^{\tau-1} \sum_{(j,o)} p_{jt} a_{io} q_{jot} + y_{i\tau} \qquad \forall i \in \mathcal{I}. \tag{16}$$

By multiplying both sides of constraints (9c) by p_{jt}, aggregating over o and j, we can conclude that

$$y_{it} = \Big(\sum_{j \in \mathcal{J}} p_{jt} \Big) y_{it} \geq \sum_{j \in \mathcal{J}} p_{jt} \sum_{o \in \mathcal{M}_j} a_{io} q_{jot} \qquad \forall t \in \mathcal{T}, i \in \mathcal{I},$$

where the equality holds since $\sum_j p_{jt} = 1$ for all t. Using this result together with (16) gives

$$\begin{aligned} c_i &\geq \sum_{t=1}^{\tau-1} \sum_{(j,o)} p_{jt} a_{io} q_{jot} + \sum_{(j,o)} p_{j\tau} a_{io} q_{jo\tau} \\ &= \sum_{(j,o)} a_{io} \sum_{t \in \mathcal{T}} p_{jt} q_{jot} \\ &= \sum_{(j,o)} a_{io} w_{jo} \qquad \forall i \in \mathcal{I}, \end{aligned}$$

where the second equality follows from the definition of w. This completes the proof. □

The above lemma of course implies that solving the DLP yields an upper bound on V^{DP}, and it allows us to state the final order.

Corollary 2. *The following relationship holds:* $V^{DP} \leq V^{AF} \leq V^{AFR} \leq V^{DLP}$.

5. The Impact of Product Design

In this section, we analyze the particular class of instances in which multi-mode products are flexible products. We first introduce two assumptions to define these instances more precisely before we derive, under these additional assumptions, an equivalent reformulation of the pricing problem (10). By means of this reformulation, we can establish the integrality property of the pricing problem, thereby achieving the equivalence between the ALP (6) and its corresponding reduction (9).

To facilitate the discussion, we introduce a few more notation. Let $\mathcal{J}^{\mathrm{si}} = \{j \in \mathcal{J} : |\mathcal{M}_j| = 1\}$ denote the set of all $n^{\mathrm{si}} = |\mathcal{J}^{\mathrm{si}}|$ single-mode products and let $\mathcal{J}^{\mathrm{mu}} = \{j \in \mathcal{J} : |\mathcal{M}_j| > 1\}$ denote the set of all $n^{\mathrm{mu}} = |\mathcal{J}^{\mathrm{mu}}|$ multi-mode products. Throughout this section, if we refer to a single-mode product or to a multi-mode product, we use either the index $h \in \mathcal{J}^{\mathrm{si}}$ or the index $l \in \mathcal{J}^{\mathrm{mu}}$, respectively. Whenever the particular type of a product is not relevant, the index $j \in \mathcal{J}$ is used.

5.1. Flexible Products with Discounted Fares

According to Gallego and Phillips (2004), a flexible product is a set of two or more alternative single-mode products serving the same market. Hence, a flexible product is a multi-mode product with a restricted definition of the associated mode set. Consider the formal definition below.

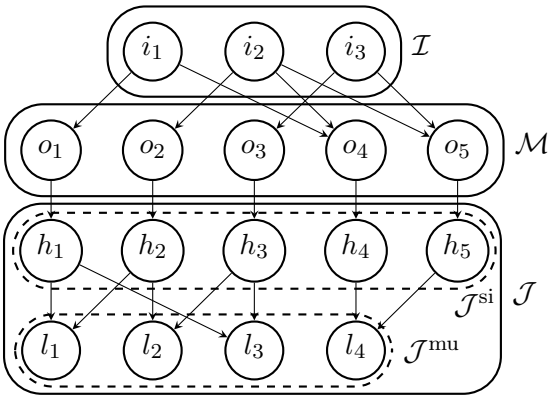

Figure 1: Illustration of a nested mode set structure for $n^{\text{si}} = n^{\text{o}} = 5$, $m = 3$, and $n^{\text{mu}} = 4$ double-mode products

Definition 1. *A multi-mode product $l \in \mathcal{J}^{mu}$ is a flexible product if it consists of one or more constituting single-mode products $h \in \mathcal{J}^{si}$. For any given $h \in \mathcal{J}^{si}$ and $l \in \mathcal{J}^{mu}$, product h is said to constitute product l if they share a common mode, i.e., if $\mathcal{M}_l \cap \mathcal{M}_h \neq \varnothing$.*

For the remainder of this section, we assume that all multi-mode products are flexible products. Noting the condition of Definition 1, this assumption is equivalent to the condition that for every product-mode combination (l, o) at least one single-mode product requires the same resources as mode o, or more formally,

$$\forall l \in \mathcal{J}^{\text{mu}}, o \in \mathcal{M}_l : \exists h \in \mathcal{J}^{\text{si}} \text{ with } \mathcal{M}_h \subset \mathcal{M}_l.$$

We refer to this condition as *nested* mode set structure. For illustration, the graph of Figure 1 depicts a numerical example of a nested mode set structure. This graph consists of one node for each resource $i \in \mathcal{I}$, one node for each mode $o \in \mathcal{M}$, and one node for each product $j \in \mathcal{J}$. The following arcs connect these nodes: One arc from each resource $i \in \mathcal{I}$ to a mode $o \in \mathcal{M}$ if $a_{io} = 1$, one arc from each mode $o \in \mathcal{M}$ to product $h \in \mathcal{J}^{\text{si}}$ if $o \in \mathcal{M}_h$. Finally, an arc goes from product $h \in \mathcal{J}^{\text{si}}$ to product $l \in \mathcal{J}^{\text{mu}}$ if h constitutes l. Note that no arcs between the individual modes and the multi-mode products are needed to represent their resource requirements.

In addition to the nested mode set structure, we assume that any multi-mode product gets a discount off the minimum fare associated with the set of its constituting products. This is not a necessary but a typical pricing strategy used when selling flexible products (Gallego and Phillips, 2004). Thus, given discount factors $0 \leq \delta_l \leq 1$ are assumed for $l \in \mathcal{J}^{\text{mu}}$ and we define the discounted fares

$$f_l = \delta_l \min_{h \in \mathcal{J}^{\text{si}}} \{f_h : h \text{ constitutes } j\} \quad \forall l \in \mathcal{J}^{\text{mu}} \tag{17}$$

5.2. Integrality Property of the Pricing Problem

To increase readability, we drop the mode index o from every u_{hot} variable which refers to a single-mode product $h \in \mathcal{J}^{\text{si}}$. For $t \in \mathcal{T}$, we further use the notation $u_t^{\text{si}} = \{u_{ht} : h \in \mathcal{J}^{\text{si}}\}$ and $u_t^{\text{mu}} = \{u_{lot} : l \in \mathcal{J}^{\text{mu}}, o \in \mathcal{M}_l\}$, which allows us to state $u_t = (u_t^{\text{si}}, u_t^{\text{mu}})$. The unique mode of product $h \in \mathcal{J}^{\text{si}}$ is denoted by o_h.

Now, consider the following integer program:

$$\zeta_t^{\text{AFP2}} = \max \Bigg\{ \sum_{h\in\mathcal{J}^{\text{si}}} p_{ht}\Big(f_h - \sum_{i\in\mathcal{I}} \vartheta_{i,t+1} a_{io_h}\Big) u_{ht} - \sum_{i\in\mathcal{I}} (\vartheta_{it} - \vartheta_{i,t+1}) x_{it} + \sum_{l\in\mathcal{J}^{\text{mu}}} \sum_{o\in\mathcal{M}_l} p_{lt}\Big(f_l - \sum_{i\in\mathcal{I}} \vartheta_{i,t+1} a_{io}\Big) u_{lot} \Bigg\} \tag{18a}$$

$$\text{s.t. } a_{io_h} u_{ht} \le x_{it} \quad \forall h \in \mathcal{J}^{\text{si}}, i \in \mathcal{I} \tag{18b}$$

$$\sum_{o\in\mathcal{M}_l} u_{lot} \le 1 \quad \forall l \in \mathcal{J}^{\text{mu}} \tag{18c}$$

$$u_{lot} \le u_{ht} \quad \forall l \in \mathcal{J}^{\text{mu}}, o \in \mathcal{M}_l, h \in \mathcal{J}^{\text{si}} : o_h = o \tag{18d}$$

$$\mathbb{1}_{\{t=1\}} c \le x_t \le c, 0 \le u_t^{\text{si}} \le 1, u_t^{\text{mu}} \ge 0 \tag{18e}$$

$$x_t, u_t^{\text{si}}, u_t^{\text{mu}} \text{ integer.} \tag{18f}$$

Using the equation $u_t = (u_t^{\text{si}}, u_t^{\text{mu}})$, note that the objective function (18a) and the domains in (18e) are identical to formulation (10). Also, constraints (18b) and (18c) are identical to the constraints in (10b) and (10c) which refer to single-mode products, $j \in \mathcal{J} \setminus \mathcal{J}^{\text{mu}}$. The linking constraints (18d) ensure that a product-mode combination (l, o) is offered only if all products $h \in \mathcal{J}^{\text{si}}$ which constitute l and for which $o_h = o$ are offered simultaneously via $u_{ht} = 1$.

It is important to note that u_t^{mu} is entirely decoupled from x_t in formulation (18). But before we make use of this key fact, the formulations (18) and (10) are compared in the following lemma.

Lemma 6. *If the mode set structure is nested and all multi-mode product fares are discounted, then the formulation* (18) *of the pricing problem is equivalent to* (10)*, i.e., $\zeta_t^{AFP2} = \zeta_t^{AFP}$ for all $t \in \mathcal{T}$.*

Proof. It is not hard to see that any feasible solution to the model (18) is feasible in model (10). The converse does not hold in general because constraints (18d) forbid solutions of the form

$$\begin{aligned} &u_{lot} = 1 \text{ for some } (l, o) \text{ with } l \in \mathcal{J}^{\text{mu}}, o \in \mathcal{M}_l \\ &\text{and } u_{ht} = 0 \text{ for at least one } h \in \mathcal{J}^{\text{si}} \text{ with } o_h = o. \end{aligned} \tag{19}$$

We show that such solutions are not optimal in formulation (10). Suppose that an optimal solution $(u_t, x_t) = (u_t^{\text{si}}, u_t^{\text{mu}}, x_t)$ to (10) is given and that it takes the form (19), i.e., it violates (18d). Let (l, o, h) refer to any violated constraint (18d). The optimality of u_t^{mu} implies $f_l \ge \sum_{i\in\mathcal{I}} \vartheta_{i,t+1} a_{io}$. The definition of a discounted fare (17) implies that $f_l \le f_h$. Hence, setting $u_{ht} = 1$ cannot decrease but it can increase the objective value because $f_h \ge \sum_{i\in\mathcal{I}} \vartheta_{i,t+1} a_{io}$. This contradicts the optimality of u_t^{si} and completes the proof. □

Let the linear relaxation of model (18) be denoted by (AFP2-L) and its optimal value by $\zeta_t^{\text{AFP2-L}}$. Note that the argument by means of which we analyze solutions of the form (19) in the proof of Lemma 6 assumes that the u_{ht} and u_{lot} variables are binary. Clearly, this is not necessarily true for the linear relaxations of (10) and (18). If these variables are allowed to be fractional, we can use a slightly more general argument, thereby obtaining the equivalence between the linear relaxations (AFP-L) and (AFP2-L) under the same conditions of Lemma 6.

Hence, the result $\zeta_t^{\text{AFP-L}} = \zeta_t^{\text{AFP2-L}}$ for all $t \in \mathcal{T}$ together with Lemma 6 allows us to state that the formulation (10) has no integrality gap if and only if (18) has no integrality gap. The next lemma states that the latter condition is true.

Theorem 1. *If the mode set structure is nested and all multi-mode product fares are discounted, there exists an optimal integral solution to the linear program (AFP2-L).*

Proof. Suppose that we are given any optimal solution $(u_t^*, x_t^*) = (u_t^{\mathrm{si}*}, u_t^{\mathrm{mu}*}, x_t^*)$ to (AFP2-L) in any $t \in \mathcal{T}$. First, it is not hard to see that $(u_t^{\mathrm{si}*}, x_t^*)$ is an optimal solution to the problem which results from dropping $u_t^{\mathrm{mu}*}$ from the model (AFP2-L). For this restricted model, the existence of an optimal integral solution is known (Tong and Topaloglu, 2014).

Therefore, we take a copy $(u_t^{\mathrm{si}}, u_t^{\mathrm{mu}}, x_t) = (u_t^{\mathrm{si}*}, u_t^{\mathrm{mu}*}, x_t^*)$ and set $u_t^{\mathrm{mu}} = 0$ to obtain the solution $(u_t^{\mathrm{si}}, 0, x_t)$ that is locally optimal in (AFP2-L) and integral, by definition. Then, define

$$u_{lot} = \begin{cases} 1 & \text{if } u_{ht}^* = 1 \ \forall h \in \mathcal{J}^{\mathrm{si}} : o_h = o, \\ & o = \omega_l, \text{ and } f_l \geq \sum_{i \in \mathcal{I}} \vartheta_{i,t+1} a_{i\omega_l} \quad \forall l \in \mathcal{J}^{\mathrm{mu}}, o \in \mathcal{M}_l, \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

where $\omega_l = \arg\min_{o \in \mathcal{M}_l} \{\sum_{i \in \mathcal{I}} \vartheta_{i,t+1} a_{io}\}$ for $l \in \mathcal{J}^{\mathrm{mu}}$ denotes a least cost mode (ties are broken arbitrarily). By (20), the updated solution $(u_t^{\mathrm{si}}, u_t^{\mathrm{mu}}, x_t)$ is feasible in (AFP2-L) and integral, but we still need to analyze whether this solution is optimal.

Consider any product $h \in \mathcal{J}^{\mathrm{si}}$ not offered in the given optimal solution via $u_{ht}^* = 0$. Note that this can have the following reasons: There is not enough capacity by (18b) or the product is not profitable because $f_h < \sum_{i \in \mathcal{I}} \vartheta_{i,t+1} a_{io_h}$. For any product-mode combination $(l, o), l \in \mathcal{J}^{\mathrm{mu}}, o \in \mathcal{M}_l$ which is constituted by h and for which $o = o_h$, the first case implies that (l, o) is infeasible. The second case implies that the combination (l, o) is feasible but not profitable since product l's fare (17) is discounted. In either case, the decision to reject (l, o) via $u_{lot} = 0$ defined by (20) is optimal.

It remains to examine whether (20) makes an optimal assignment if a product has one or more profitable modes. Thus, we consider now any $l \in \mathcal{J}^{\mathrm{mu}}$ which fulfills $f_j \geq \sum_{i \in \mathcal{I}} \vartheta_{i,t+1} a_{io}$ for at least one $o \in \mathcal{M}_l$. Note that if the product-mode combination (l, o) is profitable, then every constituting product $h \in \mathcal{J}^{\mathrm{si}}$ with $o_h = o$ is also profitable by (17). From the optimality of $(u_t^{\mathrm{si}*}, u_t^{\mathrm{mu}*})$, it follows for every profitable product-mode combination (l, o) that the constraints in (18d) referring to (l, o) can be replaced by $u_{lot}^* \leq 1$. Using this result and constraints (18c), product j's maximum total contribution to the objective function (18a) is given by

$$\begin{aligned} &\sum_{o \in \mathcal{M}_l} \left[f_l - \sum_{i \in \mathcal{I}} \vartheta_{i,t+1} a_{io} \right] u_{lot}^* = \max_{o \in \mathcal{M}_l} \left\{ f_l - \sum_{i \in \mathcal{I}} \vartheta_{i,t+1} a_{io} \right\} \\ &= f_l - \min_{o \in \mathcal{M}_l} \left\{ \sum_{i \in \mathcal{I}} \vartheta_{i,t+1} a_{io} \right\} = f_l - \sum_{i \in \mathcal{I}} \vartheta_{i,t+1} a_{i\omega_l} = \sum_{o \in \mathcal{M}_l} \left[f_j - \sum_{i \in \mathcal{I}} \vartheta_{i,t+1} a_{io} \right] u_{lot}, \end{aligned}$$

where the third equality above is true by the definition of ω_l and the last equality follows from (20). Thus, $(u_t, x_t) = (u_t^{\mathrm{si}}, u_t^{\mathrm{mu}}, x_t)$ is integral and optimal in (AFP2-L), which completes the proof. □

Our proof of Theorem 1 has an algorithmic implication. The pricing problem does not need to be solved from scratch for the entire set of products $\mathcal{J}$. It can be solved in two phases: First, all multi-mode products are dropped from the model, i.e., a restricted pricing problem without u_t^{mu} is solved. The restricted pricing problem in every $t \in \mathcal{T}$ has $\sum_{l \in \mathcal{J}^{\mathrm{mu}}} |\mathcal{M}_l|$ variables less and $m \sum_{l \in \mathcal{J}^{\mathrm{mu}}} |\mathcal{M}_l| + n^{\mathrm{mu}}$ constraints less than the full pricing problem. In addition, one can exploit network flow structure of the restricted pricing problem (Tong and Topaloglu, 2014) and solve this problem more efficiently. Once we know all optimal decisions u_{ht}, the second phase begins in which we determine the $\sum_{l \in \mathcal{J}^{\mathrm{mu}}} |\mathcal{M}_l|$ optimal decisions u_{lot} according to the update rule (20).

6. Strengthening the Reduction

The fact that the upper bound produced by the reduction can deviate from the ALP bound gives rise to the question whether the gap $V^{\mathrm{AF}} - V^{\mathrm{AFR}}$ can be shrunk efficiently. The cutting plane method devised in this section serves that purpose. Note the major difference between our method and the cut generation procedure of Tong and Topaloglu (2014). These authors intend to solve a compact formulation by dynamically generating constraints which are necessary to describe the given problem. In contrast, our proposed method is triggered once an optimal solution to the reduction (9) is available. We then separate valid inequalities to strengthen the formulation, i.e., we receive upper bounds tighter than V^{AFR} in return for the additional effort.

6.1. Polyhedral Structure of the Pricing Problem

Our approach to identify valid inequalities for the reduction (9) is to identify inequalities that are valid for the compact integer model AIP (12). Using Lemma 3, any such valid inequality preserves all feasible solutions to the integer program (12) but it may separate fractional solutions to its linear relaxation (9). Noting constraints (12c), the analysis of the polyhedral structure of the pricing problem (10) is clearly one source of valid inequalities.

First, we show that (10) can be viewed as a pure binary program. Note that the objective function coefficients associated with the x_{it} variables are nonpositive because of the time monotonicity of ϑ_{it} in an optimal solution to the ALP (5). In Appendix C, we prove this and other structural properties. As a result, any x_{it} variable takes either zero, one, or c_i in an optimal solution to the pricing problem (see also Adelman, 2007). The case $x_{it} = c_i$ occurs only in period $t = 1$ due to the restricted domain of the x_{it} variables in this period. We can handle this by setting $x_{i1} = 1$ for $i \in \mathcal{I}$ and adding the constant term $\sum_{i\in\mathcal{I}}(\vartheta_{it} - \vartheta_{i,t+1})(c_i - 1)$ to the objective function (10a) in period $t = 1$. For the sake of the argument, we simply replace the original domains (10d)–(10e) by

$$x_t, u_t \text{ binary} \quad \forall t \in \mathcal{T}. \tag{21}$$

Moreover, using the notation $\bar{x}_t = \{\bar{x}_{it} : i \in \mathcal{I}\}$ to represent the complement of the decision variables x_t, i.e., $\bar{x}_{it} = 1 - x_{it} \in \{0, 1\}$ for $i \in \mathcal{I}, t \in \mathcal{T}$, the constraints (10b) can be rewritten as

$$u_{jot} + \bar{x}_{it} \leq 1 \quad \forall (j, o) \in \mathcal{J} \times \mathcal{M}, i \in \mathcal{I} : a_{io} = 1. \tag{22}$$

We can now describe the set of feasible solutions to the pricing problem (10) by writing

$$\mathcal{P}_t = \{(u_t, \bar{x}_t) : \sum_{o\in\mathcal{M}_j} u_{jot} \leq 1 \ \forall j \in \mathcal{J}, (22), \text{ and binary}\} \quad \forall t \in \mathcal{T}.$$

The set of feasible solutions to a set-packing problem is defined as follows.

Definition 2. *The bounded polyhedron $\mathcal{S} = \{\mu : \tilde{A}\mu \leq 1 \text{ and binary}\}$ with a set of columns $\mathcal{Q}$, a set of rows $\mathcal{F}$, a binary $|\mathcal{F}| \times |\mathcal{Q}|$ coefficient matrix $\tilde{A}$, and a $|\mathcal{Q}|$-vector $\mu = (\mu_q)$ of binary variables is called a set-packing structure.*

The idea of the following proof is to cast the pricing problem (10) as a set-packing problem.

Theorem 2. *The pricing problem (10) has a set-packing structure, i.e., $\zeta_t^{AFP} = \max\{\omega_t^\top \mu : \mu \in \mathcal{S}\}$ for all $t \in \mathcal{T}$, where $\omega_t = (\omega_{qt})$ is a $|\mathcal{Q}|$-vector of weights.*

Proof. The proof begins with the construction of the set of columns $\mathcal{Q}$. Let each column $q \in \mathcal{Q}$ refer to either a product-mode combination, denoted by $q = (j, o) \in \mathcal{J} \times \mathcal{M}$, or a resource, denoted by $q = (i, 0) \in \mathcal{I} \times \{0\}$. Thus, $|\mathcal{Q}| = |\mathcal{J} \times \mathcal{M}| + m$ and we define the weights according to

$$\omega_{qt} = \begin{cases} p_{jt}\Big(f_j - \sum_{i \in \mathcal{I}} \vartheta_{i,t+1} a_{io}\Big) & \text{if } q = (j, o) \in \mathcal{J} \times \mathcal{M} \\ \vartheta_{i,t+1} - \vartheta_{it} & \text{if } q = (i, 0) \in \mathcal{I} \times \{0\} \end{cases} \quad \forall q \in \mathcal{Q}, t \in \mathcal{T}$$

It follows the construction of the set of rows $\mathcal{F}$. We define a row $\mathcal{F}_p \in \mathcal{F}$ as a subset of columns $\mathcal{F}_p \subseteq \mathcal{Q}, p = 1, \ldots, |\mathcal{F}|$. Specifically, for each $(j, o, i) \in \mathcal{J} \times \mathcal{M} \times \mathcal{I}$ with $a_{io} = 1$ define a row $\mathcal{F}_p = \{q \in \mathcal{Q} : q = (j, o) \text{ or } q = (i, 0)\}$, and for each $j \in \mathcal{J}$ define a row $\mathcal{F}_p = \{q \in \mathcal{Q} : q = (j, o)\ \forall o \in \mathcal{M}_j\}$. Then, $|\mathcal{F}| = |\mathcal{J} \times \mathcal{M} \times \mathcal{I}| + n$ and the incidence matrix $\tilde{A} = (\tilde{a}_{pq})$ can be derived as follows: For all $q \in \mathcal{Q}$ and $p = 1, \ldots, |\mathcal{F}|$ set $\tilde{a}_{pq} = 1$ if $q \in \mathcal{F}_p$ (0, otherwise). Observe that any solution (u_t, x_t) in $\mathcal{P}_t$ has a corresponding solution μ in $\mathcal{S}$ for any $t \in \mathcal{T}$. Finally, the definition of ω_t yields the claim $\zeta_t^{\text{AFP}} = \max\{\omega_t^\top \mu : \mu \in \mathcal{S}\}$ for $t \in \mathcal{T}$. This completes the proof. □

6.2. A New Cutting Plane Method

Theorem 2 grants access to the set of inequalities which are valid for the set-packing problem, e.g., the well-known clique cuts and odd-cycle cuts. In the following, we devise a cutting-plane method based on the separation of odd-cycle cuts. These cuts are not necessarily facet-defining inequalities for the set-packing problem (Padberg, 1973) but their exact separation can be done efficiently (Grötschel *et al.*, 1988).

Given any $t \in \mathcal{T}$, constraints (9c) and (9d) of the reduction (9) can be represented as an undirected graph (V, E), known as conflict graph. The node set V of the conflict graph has a node for each y_{it} variable and a node for each q_{jot} variable. We define the node set using the notation of the previous subsection, $V = \mathcal{Q} = V^{\mathcal{J}} \cup V^{\mathcal{I}}$, where $V^{\mathcal{J}}$ denotes the subset of nodes associated with product-mode combinations $(j, o) \in \mathcal{J} \times \mathcal{M}$ and $V^{\mathcal{I}}$ is the subset of nodes associated with resources $(i, 0) \in \mathcal{I} \times \{0\}$. An edge connects two nodes if the corresponding variables occur simultaneously in (9c) or in (9d).

Let the subgraph $C = (V(C), \bar{E}) \subseteq (V, E)$ denote a cycle with an odd number of edges. Using $\bar{y}_{it} = 1 - y_{it}$ for $i \in \mathcal{I}, t \in \mathcal{T}$, the odd-cycle inequalities are given by

$$\sum_{(j,o) \in V^{\mathcal{J}}(C)} q_{jot} + \sum_{(i,0) \in V^{\mathcal{I}}(C)} \bar{y}_{it} \leq \frac{|V(C)| - 1}{2} \quad \forall \text{ odd cycles } C \subseteq (V, E), t \in \mathcal{T}. \tag{23}$$

We now state the separation problem taking a feasible solution (y, q) to the reduction (9) as given. For each edge $\{r, s\} \in E$, define the edge cost $z_{rs}^t = 1/2(1 - \gamma_{rt} - \gamma_{st})$, where

$$\gamma_{rt} = \begin{cases} q_{jot} & \text{if } r = (j, o) \in V^{\mathcal{J}} \\ \bar{y}_{it} & \text{if } r = (i, 0) \in V^{\mathcal{I}} \end{cases} \quad \forall r \in V, t \in \mathcal{T}.$$

Note that only the node weights γ_{rt} vary in time, while the node set V and the edge set E of the conflict graph are constant with respect to time. It follows from (23) that finding a violated odd-cycle inequality amounts to decide whether there exists an odd cycle $C \subseteq (V, E)$ with

$$\zeta_t^{\text{SEP}}(C) = \sum_{\{r,s\} \in \bar{E}} z_{rs}^t < 1/2. \tag{24}$$

(a) Constraint system

$$
\begin{array}{llllllll}
q_{a1t} & & & & + \bar{y}_{1t} & & \leq & 1 \\
& q_{a2t} & & & & + \bar{y}_{2t} & \leq & 1 \\
& & q_{b2t} & & & + \bar{y}_{2t} & \leq & 1 \\
& & & q_{b3t} & & & + \bar{y}_{3t} \leq & 1 \\
& & & & q_{c1t} + \bar{y}_{1t} & & \leq & 1 \\
& & & & & q_{c3t} + \bar{y}_{3t} \leq & 1 \\
q_{a1t} + q_{a2t} & & & & & & \leq & 1 \\
& & q_{b2t} + q_{b3t} & & & & \leq & 1 \\
& & & & q_{c1t} + q_{c3t} & & \leq & 1
\end{array}
$$

(b) Conflict graph

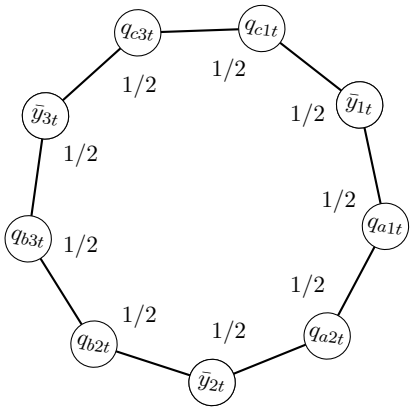

Figure 2: Constraint system and corresponding conflict graph of Example 1

We denote the problem associated with verifying (24) by $(\text{SEP})_t$ for $t \in \mathcal{T}$. Lemma (9.1.11) in Grötschel *et al.* (1988), for instance, states that there exists a polynomial time algorithm to solve $(\text{SEP})_t$. If the value ζ_t^{SEP} is interpreted as the length of an odd cycle, we can verify the existence of an odd cycle shorter than 1/2 by solving at most $\mathcal{O}(|V|)$ shortest-path problems.

The complete cutting plane method is sketched in Algorithm 1 in Appendix B. Let $\mathcal{R}_k$ denote the index set of rows that correspond to violated odd-cycle cuts found in iteration $1, \ldots, k$. In each iteration k, we solve the separation problem $(\text{SEP})_t$ for all $t \in \mathcal{T}$. If an odd-cycle cut is violated in t, we add the row index (C, t) to $\mathcal{R}_k$. $\text{AFR}(\mathcal{R}_k)$ refers to the model obtained from adding all rows in $\mathcal{R}_k$ to the reduction (9).

We finally revisit Example 1 to demonstrate that (23) indeed happens to be violated.

Example 1 (cont'd). Using $\bar{y}_{it} = 1 - y_{it}$, the set of feasible solutions to the linear program (AFP-L) of Example 1 can be written as the constraint system depicted in Figure 2a. Figure 2b shows the corresponding conflict graph. Adding up all these constraints, we get

$$2 \left[\sum_{(j,o)} q_{jot} + \sum_{i \in \mathcal{I}} \bar{y}_{it} \right] \leq 9.$$

The coefficient 2 on the left-hand side of the above inequality can be interpreted as the number of edges to which each of the nine nodes is incident. Dividing both sides of this inequality by 2 and rounding off the resulting right hand side value yields the new cut

$$\sum_{(j,o)} q_{jot} + \sum_{i \in \mathcal{I}} \bar{y}_{it} \leq \frac{9-1}{2}.$$

This cut forbids the optimal solution to the linear program (AFP-L) of Example 1 with all variables taking 1/2 since $9/2 = 4.5 > (9-1)/2 = 4$.

7. Computational Tests

The main goal of this section is to assess the findings of Sections 5 and 6 from a computational point of view. We describe our instances in more detail and also a few details on our implementations before reporting the numerical results of our computational experiments.

7.1. Test Instances

We created two groups of test instances, "RMF" and "RMM". These instances include data from the well-known instances of Topaloglu (2009) designed to evaluate algorithms for the ordinary network revenue management problem (NRM). They have a time horizon of length $\tau \in \{200, 600\}$.

RMF Instances. Each instance is characterized by a tuple $(\tau, N, \alpha, \kappa, n^{\mathrm{mu}})$. It includes the complete resource and product data of the original NRM instance, i.e., the capacities, the itineraries, and the fares. These itineraries are defined over a hub-and-spoke flight network with a single hub and N spokes. Each itinerary is combined with two fare classes where the revenue of the high-fare class is $\kappa \in \{4, 8\}$ times the revenue of the low-fare class. We added a varying number $n^{\mathrm{mu}} \in \{10, 15, 20, 30\}$ of double-mode products to these itinerary-fare class combinations. Two different types of mode sets were defined:

1. Mode sets containing two single-leg (hub→non-hub) itineraries.
2. Mode sets containing two double-leg (non-hub→non-hub) itineraries.

Note that the double-mode products are flexible products because they are constituted by single-mode products, the original itinerary-fare class combinations. For each double-mode product we defined a single discounted fare which is 25% smaller than the lowest fare associated with any of its constituting products, and we generated random arrival rates.

RMM Instances. Each instance is characterized by a tuple $(\tau, O, \alpha, \kappa, n^{\mathrm{mu}})$. There are no single-mode products ($n^{\mathrm{si}} = 0$). The number n^{mu} of multi-mode products varies between 40 and 144 each of which has two fare classes. The fares and probabilities were taken from the original NRM instance. We use the maximum cardinality of a mode set $O \in \{4, 5, 6, 8\}$ to indicate the complexity of the instances. A mode is a route through the network which is defined by $N^{\mathrm{O}} + N^{\mathrm{I}} + N^{\mathrm{D}}$ nodes and $m \in \{12, 14, 16, 18\}$ resources linking the nodes. A node represents either an origin (O), an intermediate stop (I), or a destination (D). These parameters also vary, i.e., $N^{\mathrm{O}}, N^{\mathrm{I}} \in \{2, 3\}$ and $N^{\mathrm{D}} \in \{4, 5\}$. A resource connects origins and intermediate stops or intermediate stops and destinations but not origins and destinations. Figure 3 depicts an instance with $N^{\mathrm{O}} = N^{\mathrm{I}} = 2$, $N^{\mathrm{D}} = 4$, and $m = 12$. There are 12 single-resource modes and 24 double-resource modes in total. For example, consider a product that departs from node A and ends at one of the two intermediate stops C or D. This product has two modes which are given by the mode set $\{1, 3\}$. Consider another product with three modes going from node B via C to either E, F, or G. This product is defined by the mode set $\{(1, 5), (1, 7), (1, 8)\}$.

Appendix D provides a summary of the characteristics of all test instances. We measure the tightness of the leg capacities using the load factor

$$\alpha = \frac{\sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{J}} \mathbb{1}_{\{\exists o \in \mathcal{M}_j : a_{io} = 1\}} p_{jt}}{\sum_{i \in \mathcal{I}} c_i}.$$

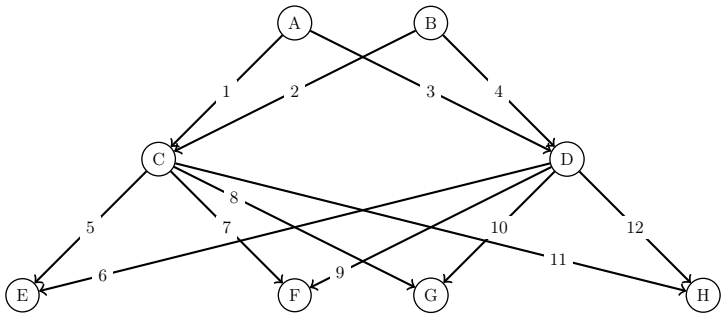

Figure 3: Example network with $N^{\mathrm{O}} = 2$ origins, $N^{\mathrm{I}} = 2$ intermediate stops, $N^{\mathrm{D}} = 4$ destinations; nodes are numbered from A to H, arcs from 1 to 12

7.2. Computational Setup

All tests were performed on a standard PC with an Intel(R) Core(TM) i7-2600 running at 3.4 GHz with 16 GB of main memory. Algorithms were coded in C++ and compiled in release mode with MS-Visual Studio 2010. CPLEX 12.5 serves as general-purpose mixed-integer programming solver and we allowed CPLEX to allocate two threads.

We compute the ALP bound V^{AF} by solving the model (6) with the convexity constraints (8) column generation because these additional constraints stabilize the algorithm. We adapt the basic configuration as described by Adelman (2007) which includes:

- Batchwise addition of generated columns.
- Addition of variables to the initial (primal) RMP in order to impose
  - the dual constraints $\vartheta_{it} \geq \vartheta_{i,t+1}, i \in \mathcal{I}, t \in \mathcal{T}$ and
  - the dual constraints $\theta_t \geq \theta_{t+1}, t \in \mathcal{T}$.

We prove that at least one optimal solution satisfies these additional dual constraints in Appendix C. As a side note, it is well-known that column generation algorithms can benefit from the stabilizing effect of such "dual-optimal inequalities" (Ben Amor *et al.*, 2006; Gschwind and Irnich, 2014).

Columns with positive reduced profit smaller than a minimum threshold value of 0.001 were discarded. In all our computations, column generation was terminated if the sum of the reduced profits in the current iteration was greater than the optimal value of the current RMP multiplied by the parameter Ω_{cg} which we set to $1 \cdot 10^{-6}$.

We implemented the cutting plane method described in Section 6.2 using a Fibonacci heap implementation of Dijkstra's algorithm to compute the shortest paths in the separation routine. If the violation of an odd-cycle cut was greater than or equal to a predefined minimum violation of 0.01, this cut made it into the strengthened model, otherwise the cut was discarded. During our experiments, batchwise addition of cuts turned out to be a better strategy than adding only the most violated cut. The initial reduction (9) was solved without dynamic constraint generation.

7.3. Numerical Results

This section summarizes the results of our computational experiments. In Appendix E, we report our numerical results in more detail.

Table 1 presents our results with respect to the RMF instances. All values are averages obtained by aggregating over instances with the same $(\tau, N, n^{\mathrm{mu}})$. Columns two to four show the average upper bounds on the maximum expected revenue V^{DP} produced by solving the ALP (6), the reduction (9), and the DLP (2), respectively. In columns five and six, we see the average number of seconds that were required to compute V^{AF} and V^{AFR}, respectively. The the last column shows the average relative improvement of V^{AFR} over V^{DLP}.

	Upper bounds			CPU seconds		Δ (%)
$(\tau, N, n^{\mathrm{mu}})$	V^{AF}	V^{AFR}	V^{DLP}	AF	AFR	$V^{\mathrm{DLP}} - V^{\mathrm{AFR}}$
$(200, 4, 10)$	22,442.3	22,442.3	22,568.7	33.1	0.5	0.6
$(200, 5, 15)$	22,576.9	22,577.0	22,693.7	64.5	0.8	0.6
$(200, 6, 20)$	23,607.7	23,607.9	23,810.9	128.4	1.2	0.9
$(200, 8, 30)$	20,659.4	20,659.5	20,859.8	426.0	2.8	1.0
$(600, 4, 10)$	37,941.8	37,941.9	38,122.6	146.3	1.7	0.5
$(600, 5, 15)$	39,343.9	39,344.0	39,532.8	639.2	3.4	0.5
$(600, 6, 20)$	31,659.1	31,659.2	31,918.9	2,358.1	6.3	0.9
$(600, 8, 30)$	TL	28,361.7	28,680.2	TL	18.5	1.0

Table 1: Average upper bounds and average computation times for the RMF instances; the time limit TL was set to 7200 seconds

By construction of the RMF instances, the numbers in the second and third column are nearly identical. This is not surprising since all multi-mode products are flexible products and therefore the reduction (9) is guaranteed to be equivalent to the ALP (6). The deviation of V^{AFR} from V^{AF} in some instances indicates that the column generation algorithm terminated prematurely due to our stopping criterion based on the value Ω_{cg}. The average ALP bound outperforms the average DLP bound in all cases as expected. The entry "TL" means that it was not possible to solve the ALP within a time limit which we set to 7200 seconds.

The results regarding the RMM instances are depicted in Tables 2 and 3. These instances are designed to be harder to solve than the RMF instances in the sense that all products have multiple modes but are no flexible products. Thus, the average relative gaps between V^{AF} and V^{AFR} shown in the last column Table 2 are strictly positive. The reduction-based bounds are quite close to

	Upper bounds			CPU seconds		Δ (%)	
$(\tau, O, n^{\mathrm{mu}})$	V^{AF}	V^{AFR}	V^{DLP}	AF	AFR	$V^{\mathrm{DLP}} - V^{\mathrm{AF}}$	$V^{\mathrm{AF}} - V^{\mathrm{AFR}}$
$(200, 4, 40)$	25,664.8	25,794.6	25,825.4	102.0	1.8	0.7	0.5
$(200, 5, 60)$	25,528.5	25,684.9	25,730.7	139.2	5.0	0.8	0.7
$(200, 6, 84)$	27,420.5	27,604.9	27,640.8	321.0	16.6	0.9	0.7
$(200, 8, 144)$	22,842.6	22,973.7	23,024.5	629.2	80.8	0.9	0.6
$(600, 4, 40)$	38,249.0	38,363.6	38,438.8	791.5	11.0	0.5	0.3
$(600, 5, 60)$	38,525.1	38,678.6	38,745.4	933.4	45.1	0.6	0.4
$(600, 6, 84)$	33,036.4	33,206.9	33,264.0	2,308.2	131.8	0.7	0.6

Table 2: Average upper bounds and computation times for the RMM instances

	AFR with Odd-Cycle Cuts			AFR vs. OC (%)	
$(\tau, O, n^{\mathrm{mu}})$	V^{OC}	CPU seconds	#Iter.	Ratio of gaps	Ratio of times
$(200, 4, 40)$	25,732.7	4.3	3.7	0.47	0.04
$(200, 5, 60)$	25,653.1	15.1	4.8	0.19	0.11
$(200, 6, 84)$	27,548.3	88.6	5.3	0.29	0.28
$(200, 8, 144)$	23,756.2	404.9	8.2	0.44	0.64
$(600, 4, 40)$	38,313.8	24.7	4.8	0.42	0.03
$(600, 5, 60)$	38,640.4	80.5	5.2	0.24	0.09
$(600, 6, 84)$	33,152.7	384.9	6.2	0.32	0.17

Table 3: Improved average upper bounds using odd-cycle cuts and average computation times for the RMM instances

the ALP bounds, i.e., the value V^{AFR} deviates from V^{AF} typically by less than 1%. Nevertheless, these gaps indicate a potential benefit from strengthening the reduction (9). Note that the upper bound V^{AFR} is always tighter than V^{DLP} as indicated by the corresponding average relative gaps given in the seventh column of Table 2.

Table 3 gives insights into the effectiveness of the odd-cycle cuts. We use V^{OC} to refer to the upper bound that is available after termination of the cutting plane method, i.e., the optimal objective function value of the model $\mathrm{AFR}(\mathcal{R}_k)$ in the last iteration k of Algorithm 1. The average of the improved upper bounds is shown in the second column of Table 3. The average number of seconds elapsed during the solution of the initial model (9) and the ensuing cutting plane iterations are given in the third column. The fourth column shows the average number of cutting plane iterations. Moreover, the ratios stated in the last two columns of Table 3 are defined as follows:

$$\text{ratio of gaps} = \frac{V^{\mathrm{AFR}} - V^{\mathrm{OC}}}{V^{\mathrm{AFR}} - V^{\mathrm{AF}}}$$

and

$$\text{ratio of times} = \frac{\#\text{seconds to compute } V^{\mathrm{OC}}}{\#\text{seconds to compute } V^{\mathrm{AF}}}.$$

The gap $V^{\mathrm{AFR}} - V^{\mathrm{AF}}$ is the absolute gap between the reduction-based bound and the ALP bound. The gap $V^{\mathrm{AFR}} - V^{\mathrm{OC}}$ is the absolute gap after improving the reduction-based bound using Algorithm 1. Thus, "ratio of gaps" reads as the fraction of $V^{\mathrm{AFR}} - V^{\mathrm{AF}}$ that can be eliminated by separating the odd-cycle cuts. Moreover, "ratio of times" means the percentage lead by which the time to compute V^{OC} is ahead of the time to compute V^{AF} using column generation. That is, a ratio of CPU seconds equal to one or greater means that there is no advantage in using Algorithm 1 because we can have a stronger bound than V^{OC} with the same or less computational effort.

Because the average ratio of times is less than one in all instances, our cutting plane method can always improve the reduction-based bounds while being computationally competitive. The first row of Table 3 indicates that the gap between V^{AF} and V^{AFR} can be nearly halved in not more than 4% of the average time required to solve the ALP directly. The average ratio of gaps and the average ratio of times over all RMM instances are 34% and 19%, respectively. However, it can be seen that the separation of odd-cycle cuts becomes more time-consuming as the parameter O increases because O affects the number and size of the shortest-path problems to be solved.

8. Conclusion

We proposed an affine approximate linear program and a reduction for the network revenue management problem with multiple modes. We showed that the particular design of a multi-mode product has a crucial influence on the accuracy of the reduction. For the special case with flexible products, we showed that the compact model produces an upper bound on the maximum expected revenue which is as tight as that of the extensive model. For general multi-mode products, we developed a new cutting plane method motivated by a set-packing structure that we identified within the reduction. We also demonstrated that valid inequalities can be efficiently separated in order to improve the reduction-based bound.

Two streams of future research suggest themselves as a result of this paper. First, one could ask if our results regarding the flexible products extend to other product designs in a similar way. Second, it might be promising to further exploit the set-packing structure. The solution of the approximate linear program via column generation could benefit from tailored pricing solvers, and the reduction can be further strengthened, e.g., by the heuristic separation of stronger valid inequalities such as clique cuts. We finally remark that our analysis extends to separable piecewise-linear approximation in theory but the resulting reduction is still hard to solve.

References

Adelman, D. (2007). Dynamic bid prices in revenue management. *Operations Research*, **55**(4), 647–661.

Adelman, D. and Mersereau, A. J. (2008). Relaxations of weakly coupled stochastic dynamic programs. *Operations Research*, **56**(3), 712–727.

Ben Amor, H., Desrosiers, J., and Valério de Carvalho, J. M. (2006). Dual-Optimal Inequalities for Stabilized Column Generation. *Operations Research*, **54**(3), 454–463.

Bertsekas, D. P. (2012). *Dynamic programming and optimal control*. Athena Scientific, Belmont, 4 edition.

Chen, S., Gallego, G., Li, M. Z., and Lin, B. (2010). Optimal seat allocation for two-flight problems with a flexible demand segment. *European Journal of Operational Research*, **201**(3), 897–908.

Chen, V., Günther, D., and Johnson, E. (2003). Routing Considerations in Airline Yield Management. In T. A. Ciriani, G. Fasano, S. Gliozzi, and R. Tadei, editors, *Operations Research in Space and Air*, pages 333–350. Springer, Boston.

Farias, V. F. and van Roy, B. (2007). An Approximate Dynamic Programming Approach to Network Revenue Management. Technical report, Massachusetts Institute of Technology, Cambridge and MA.

Gallego, G. and Phillips, R. (2004). Revenue Management of Flexible Products. *Manufacturing and Service Operations Management*, **6**(4), 321–337.

Gallego, G., Iyengar, G., Phillips, R., and Dubey, A. (2004). Managing Flexible Products on a Network. Technical report, Columbia University, New York.

Grötschel, M., Lovász, L., and Schrijver, A. (1988). *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and combinatorics*. Springer, Berlin.

Gschwind, T. and Irnich, S. (2014). Dual Inequalities for Stabilized Column Generation Revisited. Technical Report LM-2014-03, Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz, Mainz and Germany. Forthcoming in: INFORMS Journal on Computing.

Kunnumkal, S. and Talluri, K. (2014). On the tractability of the piecewise-linear approximation for general discrete-choice network revenue management. Technical report, Universitat Pompeu Fabra.

Kunnumkal, S. and Topaloglu, H. (2010). Computing time-dependent bid prices in network revenue management problems. *Transportation Science*, **44**(1), 38–62.

Lübbecke, M. E. and Desrosiers, J. (2005). Selected Topics in Column Generation. *Operations Research*, **53**(6), 1007–1023.

Meissner, J. and Strauss, A. (2012). Network revenue management with inventory-sensitive bid prices and customer choice. *European Journal of Operational Research*, **216**(2), 459–468.

Padberg, M. W. (1973). On the facial structure of set packing polyhedra. *Mathematical programming*, **5**(1), 199–215.

Petrick, A., Gönsch, J., Steinhardt, C., and Klein, R. (2010). Dynamic control mechanisms for revenue management with flexible products. *Computers & Operations Research*, **37**(11), 2027–2039.

Post, D. (2010). Variable opaque products in the airline industry: A tool to fill the gaps and increase revenues. *Journal of Revenue and Pricing Management*, **9**(4), 292–299.

Powell, W. B. (2011). *Approximate dynamic programming*. J. Wiley & Sons, Hoboken, 2nd edition.

Puterman, M. L. (1994). *Markov decision processes*. Wiley, New York.

Schweitzer, P. J. and Seidmann, A. (1985). Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, **110**(2), 568–582.

Talluri, K. and van Ryzin, G. (1998). An analysis of bid-price controls for network revenue management. *Management Science*, **44**(11), 1577–1593.

Talluri, K. T. (2001). Airline revenue management with passenger routing control: A new model with solution approaches. *International Journal of Services Technology and Management*, **2**(1), 102–115.

Talluri, K. T. and van Ryzin, G. (2004). *The theory and practice of revenue management*. Kluwer Academic Publishers, Boston.

Tong, C. and Topaloglu, H. (2014). On the Approximate Linear Programming Approach for Network Revenue Management Problems. *INFORMS Journal on Computing*, **26**(1), 121–134.

Topaloglu, H. (2009). Using Lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. *Operations Research*, **57**(3), 637–649.

Vossen, T. W. M. and Zhang, D. (2013). Reductions of Approximate Linear Programs for Network Revenue Management. Technical report, University of Colorado at Boulder, Boulder.

Williamson, E. L. (1992). *Airline network seat inventory control*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge.

Zhang, D. and Adelman, D. (2009). An Approximate Dynamic Programming Approach to Network Revenue Management with Customer Choice. *Transportation Science*, **43**(3), 381–394.

Appendix

A. Proof of Lemma 1

The proof of Lemma 1 is based on the fact that the value $V^{\text{AF-}\theta}$ defined by the linear program

$$V^{\text{AF-}\theta} = \theta_1 + \min_{\vartheta} \sum_{i\in\mathcal{I}} c_i \vartheta_{i1}$$
$$\text{s.t. } \theta_t - \theta_{t+1} + \sum_{i\in\mathcal{I}} \left[(\vartheta_{it} - \vartheta_{i,t+1}) x_{it} + \sum_{(j,o)} p_{jt} u_{jot} a_{io} \vartheta_{i,t+1} \right] \geq \sum_{(j,o)} p_{jt} f_j u_{jot}$$
$$\forall t \in \mathcal{T}, (x_t, u_t) \in S_t.$$

is insensitive with respect to the choice of the real values $\theta = \{\theta_t : t \in \mathcal{T}\}$. We use the notation (AF-θ) to refer to the above linear program for a given θ, and we use $\vartheta(\theta) = \{\vartheta_{it}(\theta) : i \in \mathcal{I}, t \in \mathcal{T}\}$ to denote a solution to problem (AF-θ). The notation $\vartheta(\theta)$ emphasizes the dependency of this solution on the given choice of θ.

The claim is analogous to Proposition 3 of Adelman and Mersereau (2008) but the proof below is different.

Lemma 7. *$V^{AF\text{-}\theta} = V^{AF\text{-}\theta'}$ for any choice of $\theta, \theta' \in \mathbb{R}^{\tau}$.*

Proof. Choose any two $\theta, \theta' \in \mathbb{R}^{\tau}$. Assume that $\vartheta(\theta)$ is optimal in problem (AF-θ). Next, we define a solution $\vartheta(\theta')$ to problem (AF-θ') such that $\vartheta_{it}(\theta') = (\theta_t - \theta'_t)/m + \vartheta_{it}(\theta)$ for all $i \in \mathcal{I}, t \in \mathcal{T}$. This definition directly implies that $\vartheta(\theta')$ is feasible in problem (AF-θ') and that its objective function value equals $V^{\text{AF-}\theta}$. Hence, $V^{\text{AF-}\theta} \geq V^{\text{AF-}\theta'}$. The reverse inequality is follows by interchanging θ and θ' in the above argument. The proof is complete because of our arbitrary choice of θ, θ'. □

Lemma 1. *If $V^{AF\text{-}CONV}$ denotes the optimal value of the problem defined by model* (6) *and constraints* (8)*, then the equality $V^{AF\text{-}CONV} = V^{AF}$ holds.*

Proof. By Lemma 7 we can rewrite problem (5) as

$$V^{\text{AF}} = \min_{\theta,\vartheta} \theta_1 + \sum_{i\in\mathcal{I}} c_i \vartheta_{i1}$$
$$\text{s.t. } \theta_t - \theta_{t+1} + \sum_{i\in\mathcal{I}} \left[(\vartheta_{it} - \vartheta_{i,t+1}) x_{it} + \sum_{(j,o)} p_{jt} u_{jot} a_{io} \vartheta_{i,t+1} \right] \geq \sum_{(j,o)} p_{jt} f_j u_{jot}$$
$$\forall t \in \mathcal{T}, (x_t, u_t) \in S_t.$$

Noting that the dual of this linear program is given by (6) and (8) completes the proof. □

B. Cutting Plane Algorithm

This section provides a description of the cutting plane method of Section 6.2.

Algorithm 1: Cutting plane method

Set $k := 1, \mathcal{R}_k := \varnothing$
repeat
  Solve AFR($\mathcal{R}_k$)
  for $t \in \mathcal{T}$ **do**
    Solve (SEP)$_t$ for $\zeta_t^{\text{SEP}}(C)$
    if $\exists$*odd cycle* C *with* $1/2 - \zeta_t^{SEP}(C) > 0$ **then**
      Select odd cycle C
      Update $\mathcal{R}_{k+1} := \mathcal{R}_k \cup \{(C, t)\}$
  Set $k := k + 1$
until $\mathcal{R}_k = \mathcal{R}_{k-1}$

C. Proof of Structural Properties

In this section, we analzye optimal solutions to the general affine ALP

$$V^{\text{AF}} = \min_{\theta,\vartheta} \theta_1 + \sum_{i\in\mathcal{I}} c_i \vartheta_{i1} \tag{27a}$$

$$\text{s.t. } \theta_t - \theta_{t+1} + \sum_{i\in\mathcal{I}} \left[(\vartheta_{it} - \vartheta_{i,t+1}) x_{it} + \sum_{(j,o)} p_{jt} u_{jot} a_{io} \vartheta_{i,t+1} \right] \geq \sum_{(j,o)} p_{jt} f_j u_{jot}$$
$$\forall t \in \mathcal{T}, (x_t, u_t) \in S_t, \tag{27b}$$

where we use the convention that $\vartheta_{i,\tau+1} = 0, i \in \mathcal{I}$ and $\theta_{\tau+1} = 0$. More specifically, we show that at least one optimal solution to problem (27) exists in which θ and ϑ are both nonnegative and monotonic over time.

Let X denote any feasible solution to the dual of (27) that is given by problem (6) and the convexity constraints (8). We adapt the proof technique used by Adelman (2007) verify the properties. Therefore, we start with the definition of the earliest period in which resource i's capacity is expected to be sold:

$$t_i = \min\{t \in \mathcal{T} : \exists (j,o) \in \mathcal{J} \times \mathcal{M}, (x_t, u_t) \in S_t \text{ with } X^t_{xu} > 0, u_{jot} = 1, a_{io} = 1\} \quad \forall i \in \mathcal{I}.$$

It is possible to show that Lemma 1 of Adelman (2007) applies to the MNRM case. This result states that, for all $i \in \mathcal{I}$, there exists a $X^t_{xu} > 0$ with $x_{it} < c_i$ for all $t > t_i$, and that $X^t_{xu} > 0$ in any period $t \leq t_i$ only if $x_{it} = c_i$. For ease of exposition, we assume that $t_i = 1$ for all $i \in \mathcal{I}$ but the analysis below applies if the first time resource i is expected to be used is later than the first period.

Theorem 3. *There exists an optimal solution* (θ^*, ϑ^*) *to problem* (5) *that satisfies:*

(i) $\vartheta^*_{it} \geq \vartheta^*_{i,t+1}$ *for all* $i \in \mathcal{I}, t \in \mathcal{T}$,
(ii) $\vartheta^*_{it} \geq 0$ *for all* $i \in \mathcal{I}, t \in \mathcal{T}$,
(iii) $\theta^*_t \geq \theta^*_{t+1}$ *for all* $t \in \mathcal{T}$,
(iv) $\theta^*_t \geq 0$ *for all* $t \in \mathcal{T}$.

Proof. To verify part (i), suppose that we are given an optimal primal-dual pair $((\theta^*, \vartheta^*), X^*)$. Choose any resource $l \in \mathcal{I}$. Since there exists an (x_t, u_t) that satisfies $X^{*t}_{xu} > 0$ and $x_{lt} < c_l$ for all $t > 1$, constraints (27b) and the complementary slackness imply the equality

$$0 = \theta_{t+1} - \theta_t + \sum_{(j,o)} p_{jt}\Big[f_j - \sum_{i\in\mathcal{I}} a_{io}\vartheta^*_{i,t+1}\Big]u_{jot} + \sum_{i\in\mathcal{I}}(\vartheta^*_{i,t+1} - \vartheta^*_{it})x_{it} \quad \forall t > 1. \tag{28}$$

Let $x'_t = x_t$ and set $x'_{lt} = x_{lt} + 1$. Since $x'_{lt} < x_{lt} \leq c_l$, we have $\mathcal{U}_t(x_t) \subset \mathcal{U}_t(x'_t)$ and thus any action $u_t \in \mathcal{U}_t(x_t)$ is also feasible in state x'_t. By the feasibility of (θ^*, ϑ^*), the inequalities in (27b) which refer to state x'_t can be written as

$$\begin{aligned} 0 &\geq \theta_{t+1} - \theta_t + \sum_{(j,o)} p_{jt}\Big[f_j - \sum_{i\in\mathcal{I}} a_{io}\vartheta^*_{i,t+1}\Big]u_{jot} + \sum_{i\in\mathcal{I}}(\vartheta^*_{i,t+1} - \vartheta^*_{it})x'_{it} \\ &= \theta_{t+1} - \theta_t + \sum_{(j,o)} p_{jt}\Big[f_j - \sum_{i\in\mathcal{I}} a_{io}\vartheta^*_{i,t+1}\Big]u_{jot} + \sum_{i\in\mathcal{I}}(\vartheta^*_{i,t+1} - \vartheta^*_{it})x_{it} + (\vartheta^*_{l,t+1} - \vartheta^*_{lt}) \end{aligned} \tag{29}$$

for all $t > 1$. Subtracting (28) from (29) yields property (i) for resource l and all $t > 1$. Regarding period $t = 1$, it is without loss of optimality to assume that $\vartheta^*_{l1} \not< \vartheta^*_{l2}$ by a similar argument as in Step 3 of Adelman's (2007) proof of his Theorem 2. Since our choice of l was arbitrary, part (i) of this lemma holds.

Part (ii) is a direct consequence of property (i) and the boundary conditions $\vartheta_{i,\tau+1} = 0, i \in \mathcal{I}$. Part (iii) follows because $(x_t, u_t) = (0, 0)$ is a feasible state-action pair in any $t \in \mathcal{T}$ and thus $0 \geq \theta^*_{t+1} - \theta^*_t, t \in \mathcal{T}$ is implied by constraints (5b). Finally, part (iv) follows directly from part (iii) and $\theta_{\tau+1} = 0$ for all $t \in \mathcal{T}$. This completes the proof. □

D. Characteristics of Test Instances

The following table summarizes the parameters which characterize the size and complexity of our test instances.

#Periods (τ)	#Nonhub locations (N)	#Resources (m)	#Modes ($\lvert\mathcal{M}\rvert$)	#Products (n)	Card. of max. mode set (O)
$\{200, 600\}$	4	8	20	50	2
$\{200, 600\}$	5	10	30	75	2
$\{200, 600\}$	6	12	42	104	2
$\{200, 600\}$	8	16	72	174	2

Table 4: Characteristics of RMF instances

#Periods (τ)	#Sources (N^{O})	#Interm. stops (N^{H})	#Sinks (N^{D})	#Resources (m)	#Modes ($\lvert\mathcal{M}\rvert$)	#Products (n)	Card. of max. mode set (O)
$\{200, 600\}$	2	2	4	12	28	40	4
$\{200, 600\}$	2	2	5	14	34	60	5
$\{200, 600\}$	3	2	5	16	46	84	6
200	2	3	4	18	42	144	8

Table 5: Characteristics of RMM instances

E. Detailed Computational Results

The six tables of this section document the detailed numerical results of our computational experiments with the 48 RMF instances and the 42 RMM instances. All tables shown here have the same format as the corresponding tables described Section 7.3. That means the column headers of the Tables 6-7, 8-9, and 10-11 are the same as those of Table 1, 2, and 3, respectively.

	Upper bounds			CPU seconds		Δ (%)
$(\tau, N, \alpha, \kappa, n^{\mathrm{mu}})$	V^{AF}	V^{AFR}	V^{DLP}	AF	AFR	$V^{\mathrm{DLP}} - V^{\mathrm{AFR}}$
$(200, 4, 1.1, 4, 10)$	18,105.2	18,105.2	18,147.0	25.0	0.5	0.2
$(200, 4, 1.1, 8, 10)$	28,846.3	28,846.4	28,888.1	26.2	0.5	0.1
$(200, 4, 1.3, 4, 10)$	17,658.1	17,658.1	17,774.6	39.9	0.6	0.7
$(200, 4, 1.3, 8, 10)$	28,398.1	28,398.1	28,515.8	39.9	0.5	0.4
$(200, 4, 1.8, 4, 10)$	15,455.1	15,455.5	15,672.8	35.0	0.4	1.4
$(200, 4, 1.8, 8, 10)$	26,190.7	26,190.7	26,413.9	32.4	0.4	0.8
$(200, 5, 1.1, 4, 15)$	18,050.6	18,050.6	18,084.3	56.6	0.8	0.2
$(200, 5, 1.1, 8, 15)$	28,526.5	28,526.5	28,560.1	55.6	0.8	0.1
$(200, 5, 1.3, 4, 15)$	17,672.3	17,672.3	17,738.8	74.2	0.7	0.4
$(200, 5, 1.3, 8, 15)$	28,148.1	28,148.1	28,214.6	66.1	0.7	0.2
$(200, 5, 1.8, 4, 15)$	16,311.4	16,311.9	16,556.4	69.5	0.8	1.5
$(200, 5, 1.8, 8, 15)$	26,752.5	26,752.6	27,008.2	65.0	0.8	0.9
$(200, 6, 1.1, 4, 20)$	19,073.3	19,073.3	19,106.6	107.8	1.3	0.2
$(200, 6, 1.1, 8, 20)$	30,210.4	30,210.5	30,243.8	106.7	1.1	0.1
$(200, 6, 1.3, 4, 20)$	18,552.3	18,552.5	18,758.7	158.1	1.2	1.1
$(200, 6, 1.3, 8, 20)$	29,684.0	29,684.1	29,895.7	126.3	1.0	0.7
$(200, 6, 1.8, 4, 20)$	16,510.3	16,510.6	16,866.8	142.4	1.3	2.1
$(200, 6, 1.8, 8, 20)$	27,616.1	27,616.2	27,993.9	129.2	1.2	1.3
$(200, 8, 1.1, 4, 30)$	16,700.2	16,700.2	16,740.2	385.6	2.6	0.2
$(200, 8, 1.1, 8, 30)$	26,311.3	26,311.3	26,351.3	309.8	2.5	0.2
$(200, 8, 1.3, 4, 30)$	16,293.7	16,293.7	16,458.4	390.9	2.8	1.0
$(200, 8, 1.3, 8, 30)$	25,897.2	25,897.2	26,064.5	397.5	2.7	0.6
$(200, 8, 1.8, 4, 30)$	14,586.0	14,586.5	14,969.3	653.4	3.5	2.6
$(200, 8, 1.8, 8, 30)$	24,167.9	24,168.1	24,575.2	418.8	2.6	1.7

Table 6: Upper bounds and computation times for the RMF instances with $\tau = 200$ periods

	Upper bounds			CPU seconds		Δ (%)
$(\tau, N, \alpha, \kappa, n^{\mathrm{mu}})$	V^{AF}	V^{AFR}	V^{DLP}	AF	AFR	$V^{\mathrm{DLP}} - V^{\mathrm{AFR}}$
$(600, 4, 1.1, 4, 10)$	31,180.1	31,180.2	31,234.9	173.2	1.7	0.2
$(600, 4, 1.1, 8, 10)$	50,037.8	50,037.9	50,092.6	151.7	1.7	0.1
$(600, 4, 1.3, 4, 10)$	28,945.3	28,945.5	29,174.4	160.2	1.9	0.8
$(600, 4, 1.3, 8, 10)$	47,786.6	47,786.7	48,032.1	131.6	1.7	0.5
$(600, 4, 1.8, 4, 10)$	25,430.4	25,430.5	25,671.8	143.7	1.6	0.9
$(600, 4, 1.8, 8, 10)$	44,270.5	44,270.6	44,529.5	117.3	1.6	0.6
$(600, 5, 1.1, 4, 15)$	31,839.9	31,840.0	31,905.8	1,406.4	2.9	0.2
$(600, 5, 1.1, 8, 15)$	50,806.8	50,806.9	50,872.7	1,056.3	2.8	0.1
$(600, 5, 1.3, 4, 15)$	30,581.0	30,581.1	30,740.3	355.6	3.6	0.5
$(600, 5, 1.3, 8, 15)$	49,509.8	49,509.9	49,688.4	341.6	3.7	0.4
$(600, 5, 1.8, 4, 15)$	27,212.9	27,213.1	27,533.2	379.1	4.1	1.2
$(600, 5, 1.8, 8, 15)$	46,112.9	46,113.1	46,456.1	296.4	3.1	0.7
$(600, 6, 1.1, 4, 20)$	26,029.7	26,029.8	26,104.6	5,247.5	4.7	0.3
$(600, 6, 1.1, 8, 20)$	41,473.6	41,473.7	41,549.8	5,506.6	6.6	0.2
$(600, 6, 1.3, 4, 20)$	24,365.2	24,365.3	24,664.8	1,002.7	7.3	1.2
$(600, 6, 1.3, 8, 20)$	39,779.7	39,779.8	40,101.4	680.5	5.0	0.8
$(600, 6, 1.8, 4, 20)$	21,452.0	21,452.1	21,830.7	962.2	7.4	1.7
$(600, 6, 1.8, 8, 20)$	36,854.5	36,854.6	37,261.9	749.4	6.6	1.1
$(600, 8, 1.1, 4, 30)$	-	23,344.6	23,446.7	-	17.7	0.4
$(600, 8, 1.1, 8, 30)$	-	37,039.5	37,150.1	-	11.7	0.3
$(600, 8, 1.3, 4, 30)$	-	21,876.2	22,238.5	-	22.2	1.6
$(600, 8, 1.3, 8, 30)$	-	35,551.6	35,935.0	-	17.4	1.1
$(600, 8, 1.8, 4, 30)$	-	19,350.3	19,807.2	-	23.2	2.3
$(600, 8, 1.8, 8, 30)$	-	33,008.0	33,503.7	-	18.8	1.5

Table 7: Upper bounds and computation times for the RMF instances with $\tau = 600$ periods

	Upper bounds			CPU seconds		Δ (%)	
$(\tau, O, \alpha, \kappa, n^{\mathrm{mu}})$	V^{AF}	V^{AFR}	V^{DLP}	AF	AFR	$V^{\mathrm{DLP}} - V^{\mathrm{AF}}$	$V^{\mathrm{AFR}} - V^{\mathrm{AF}}$
$(200, 4, 1.7, 4, 40)$	20,679.1	20,780.4	20,801.6	126.9	1.9	0.6	0.5
$(200, 4, 1.7, 8, 40)$	33,718.8	33,815.4	33,841.6	108.9	1.9	0.4	0.3
$(200, 4, 1.9, 4, 40)$	19,400.3	19,525.6	19,554.0	99.3	1.8	0.8	0.6
$(200, 4, 1.9, 8, 40)$	32,433.9	32,560.5	32,593.9	89.7	1.8	0.5	0.4
$(200, 4, 2.3, 4, 40)$	17,365.7	17,525.5	17,560.7	93.4	1.7	1.1	0.9
$(200, 4, 2.3, 8, 40)$	30,391.1	30,560.4	30,600.7	93.8	1.9	0.7	0.6
$(200, 5, 1.8, 4, 60)$	20,667.8	20,789.1	20,820.9	132.0	6.5	0.7	0.6
$(200, 5, 1.8, 8, 60)$	33,874.7	33,982.0	34,032.7	137.6	6.5	0.5	0.3
$(200, 5, 2.0, 4, 60)$	19,150.1	19,308.1	19,343.0	135.2	5.2	1.0	0.8
$(200, 5, 2.0, 8, 60)$	32,342.6	32,498.8	32,554.7	130.0	4.5	0.7	0.5
$(200, 5, 2.4, 4, 60)$	16,980.2	17,171.0	17,210.6	153.2	3.9	1.3	1.1
$(200, 5, 2.4, 8, 60)$	30,155.6	30,360.2	30,422.4	147.1	3.4	0.9	0.7
$(200, 6, 1.8, 4, 84)$	22,204.1	22,266.0	22,274.7	302.2	18.3	0.3	0.3
$(200, 6, 1.8, 8, 84)$	35,433.8	35,498.5	35,507.0	321.5	17.7	0.2	0.2
$(200, 6, 2.0, 4, 84)$	21,206.8	21,412.0	21,450.2	321.8	18.5	1.1	1.0
$(200, 6, 2.0, 8, 84)$	34,421.3	34,637.1	34,682.3	341.8	17.5	0.8	0.6
$(200, 6, 2.3, 4, 84)$	19,031.6	19,295.8	19,349.4	323.0	14.3	1.6	1.4
$(200, 6, 2.3, 8, 84)$	32,225.4	32,519.8	32,581.4	315.7	13.0	1.1	0.9
$(200, 8, 1.7, 4, 144)$	19,098.2	19,193.6	19,230.6	593.8	88.9	0.7	0.5
$(200, 8, 1.7, 8, 144)$	30,853.5	30,966.8	31,006.4	593.8	83.9	0.5	0.4
$(200, 8, 1.8, 4, 144)$	18,173.3	18,300.7	18,351.8	628.7	83.9	1.0	0.7
$(200, 8, 1.8, 8, 144)$	29,918.1	30,070.4	30,127.5	618.3	78.5	0.7	0.5
$(200, 8, 2.1, 4, 144)$	16,170.1	16,336.8	16,406.4	711.2	68.7	1.4	1.0
$(200, 8, 2.1, 8, 144)$	27,837.3	28,056.1	28,143.1	660.1	66.1	1.1	0.8

Table 8: Upper bounds and computation times for the RMM instances with $\tau = 200$ periods

	Upper bounds			CPU seconds		Δ (%)	
$(\tau, O, \alpha, \kappa, n^{\mathrm{mu}})$	V^{AF}	V^{AFR}	V^{DLP}	AF	AFR	$V^{\mathrm{DLP}} - V^{\mathrm{AF}}$	$V^{\mathrm{AFR}} - V^{\mathrm{AF}}$
$(600, 4, 1.8, 4, 40)$	30,152.3	30,220.7	30,302.2	1,078.5	12.8	0.5	0.2
$(600, 4, 1.8, 8, 40)$	49,808.4	49,893.6	49,979.9	668.9	11.8	0.3	0.2
$(600, 4, 2.1, 4, 40)$	28,867.6	28,973.3	29,046.5	858.4	10.0	0.6	0.4
$(600, 4, 2.1, 8, 40)$	48,531.7	48,647.9	48,724.2	727.8	10.4	0.4	0.2
$(600, 4, 2.6, 4, 40)$	26,239.2	26,385.8	26,451.1	683.0	10.9	0.8	0.6
$(600, 4, 2.6, 8, 40)$	45,894.8	46,060.4	46,128.9	732.3	10.4	0.5	0.4
$(600, 5, 2.0, 4, 60)$	31,102.2	31,219.7	31,266.9	886.1	65.3	0.5	0.4
$(600, 5, 2.0, 8, 60)$	51,028.4	51,143.0	51,206.5	873.4	49.1	0.3	0.2
$(600, 5, 2.3, 4, 60)$	28,905.7	29,057.7	29,115.1	881.6	43.2	0.7	0.5
$(600, 5, 2.3, 8, 60)$	48,822.4	48,978.9	49,054.7	891.8	41.4	0.5	0.3
$(600, 5, 2.8, 4, 60)$	25,695.2	25,875.5	25,944.7	1070.0	33.8	1.0	0.7
$(600, 5, 2.8, 8, 60)$	45,596.9	45,796.7	45,884.3	997.5	37.6	0.6	0.4
$(600, 6, 1.9, 4, 84)$	26,787.4	26,826.2	26,837.7	2,341.8	155.1	0.2	0.1
$(600, 6, 1.9, 8, 84)$	42,762.3	42,803.8	42,815.6	2,377.7	141.2	0.1	0.1
$(600, 6, 2.2, 4, 84)$	25,536.2	25,733.9	25,799.7	2,548.1	119.1	1.0	0.8
$(600, 6, 2.2, 8, 84)$	41,492.0	41,703.4	41,777.4	2,137.6	139.1	0.7	0.5
$(600, 6, 2.6, 4, 84)$	22,852.2	23,103.6	23,187.8	2,288.6	121.2	1.4	1.1
$(600, 6, 2.6, 8, 84)$	38,788.4	39,070.4	39,165.5	2,155.2	115.2	1.0	0.7

Table 9: Upper bounds and computation times for the RMM instances with $\tau = 600$ periods

	AFR with Odd-Cycle Cuts			AFR vs. OC	
$(\tau, O, \alpha, \kappa, n^{\mathrm{mu}})$	V^{OC}	CPU seconds	#Iter.	Ratio of gaps	Ratio of CPU sec.
$(200, 4, 1.7, 4, 40)$	20,733.2	4.1	3	0.47	0.03
$(200, 4, 1.7, 8, 40)$	33,773.2	3.7	3	0.44	0.03
$(200, 4, 1.9, 4, 40)$	19,463.0	5.8	6	0.50	0.06
$(200, 4, 1.9, 8, 40)$	32,502.5	4.0	3	0.46	0.04
$(200, 4, 2.3, 4, 40)$	17,442.3	4.0	4	0.52	0.04
$(200, 4, 2.3, 8, 40)$	30,481.8	3.9	3	0.46	0.04
$(200, 5, 1.8, 4, 60)$	20,763.5	16.9	5	0.21	0.13
$(200, 5, 1.8, 8, 60)$	33,975.9	16.7	5	0.06	0.12
$(200, 5, 2.0, 4, 60)$	19,265.6	14.9	5	0.27	0.11
$(200, 5, 2.0, 8, 60)$	32,478.0	15.1	5	0.13	0.12
$(200, 5, 2.4, 4, 60)$	17,111.5	12.3	4	0.31	0.08
$(200, 5, 2.4, 8, 60)$	30,323.9	14.9	5	0.18	0.10
$(200, 6, 1.8, 4, 84)$	22,252.0	85.2	5	0.23	0.28
$(200, 6, 1.8, 8, 84)$	35,484.5	86.6	5	0.22	0.27
$(200, 6, 2.0, 4, 84)$	21,345.6	99.0	6	0.32	0.31
$(200, 6, 2.0, 8, 84)$	34,575.3	85.9	5	0.29	0.25
$(200, 6, 2.3, 4, 84)$	19,203.3	94.9	6	0.35	0.29
$(200, 6, 2.3, 8, 84)$	32,428.9	80.2	5	0.31	0.25
$(200, 8, 1.7, 4, 144)$	19,151.0	391.4	8	0.45	0.66
$(200, 8, 1.7, 8, 144)$	30,914.7	451.7	9	0.46	0.76
$(200, 8, 1.8, 4, 144)$	18,246.7	442.5	9	0.42	0.70
$(200, 8, 1.8, 8, 144)$	29,998.0	351.6	7	0.48	0.57
$(200, 8, 2.1, 4, 144)$	16,266.7	396.6	8	0.42	0.56
$(200, 8, 2.1, 8, 144)$	27,959.8	395.7	8	0.44	0.60

Table 10: Improved upper and computation times for the RMM instances with $\tau = 200$ periods

	AFR with Odd-Cycle Cuts			AFR vs. OC	
$(\tau, O, \alpha, \kappa, n^{\mathrm{mu}})$	V^{OC}	CPU seconds	#Iter.	Ratio of gaps	Ratio of CPU sec.
$(600, 4, 1.8, 4, 40)$	30,193.6	22.7	5	0.40	0.02
$(600, 4, 1.8, 8, 40)$	49,865.7	19.6	3	0.33	0.03
$(600, 4, 2.1, 4, 40)$	28,929.9	27.6	6	0.41	0.03
$(600, 4, 2.1, 8, 40)$	48,601.2	23.1	4	0.40	0.03
$(600, 4, 2.6, 4, 40)$	26,313.9	29.7	7	0.49	0.04
$(600, 4, 2.6, 8, 40)$	45,978.2	25.3	4	0.50	0.03
$(600, 5, 2.0, 4, 60)$	31,188.0	78.8	5	0.27	0.09
$(600, 5, 2.0, 8, 60)$	51,127.0	83.9	5	0.14	0.10
$(600, 5, 2.3, 4, 60)$	29,011.9	77.9	5	0.30	0.09
$(600, 5, 2.3, 8, 60)$	48,947.8	77.2	5	0.20	0.09
$(600, 5, 2.8, 4, 60)$	25,819.5	86.1	6	0.31	0.08
$(600, 5, 2.8, 8, 60)$	45,748.3	79.3	5	0.24	0.08
$(600, 6, 1.9, 4, 84)$	26,813.4	367.6	6	0.33	0.16
$(600, 6, 1.9, 8, 84)$	42,790.1	362.0	6	0.33	0.15
$(600, 6, 2.2, 4, 84)$	25,670.5	404.2	6	0.32	0.16
$(600, 6, 2.2, 8, 84)$	41,639.3	412.8	6	0.30	0.19
$(600, 6, 2.6, 4, 84)$	23,021.7	413.0	7	0.33	0.18
$(600, 6, 2.6, 8, 84)$	38,981.3	349.5	6	0.32	0.16

Table 11: Improved upper bounds and computation times for the RMM instances with $\tau = 600$ periods