

On the Multi-Kind BahnCard Problem

Mike Timm

University of Konstanz, Germany
mike.timm@uni-konstanz.de

Sabine Storandt

University of Konstanz, Germany
sabine.storandt@uni-konstanz.de

Abstract

The BahnCard problem is an important problem in the realm of online decision making. In its original form, there is one kind of BahnCard associated with a certain price, which upon purchase reduces the ticket price of train journeys for a certain factor over a certain period of time. The problem consists of deciding on which dates BahnCards should be purchased such that the overall cost, that is, BahnCard prices plus (reduced) ticket prices, is minimized without having knowledge about the number and prices of future journeys. In this paper, we extend the problem such that multiple kinds of BahnCards are available for purchase. We provide an optimal offline algorithm, as well as online strategies with provable competitiveness factors. Furthermore, we describe and implement several heuristic online strategies and compare their competitiveness in realistic scenarios.

2012 ACM Subject Classification Applied computing → Transportation; Theory of computation → Online algorithms

Keywords and phrases offline solution, competitiveness, traveller profiles

Digital Object Identifier 10.4230/OASICS.ATMOS.2020.2

1 Introduction

The original BahnCard problem [4] was inspired by the railway pass system of the German railway company. Buying a so called BahnCard 50 railway pass at the cost of 255€ entitles the holder to a 50% price reduction on all train ticket purchases in Germany within the next year. Similar railway pass systems exist in many other countries as well. The BahnCard problem consists of deciding on which dates a BahnCard should be purchased in order to minimize the overall cost for train journeys (including BahnCard prices and ticket prices). In the offline version of the problem, the stream of future journeys is known in advance. In the more interesting online version of the problem, one only has knowledge about past journeys but cannot foresee the future. The main goal is to come up with strategies for the online problem variant such that the so called competitiveness factor, the ratio of the resulting cost when using said online strategy and the best achievable cost of the corresponding offline problem, is as small as possible.

A BahnCard BC can be formally defined as a triple (C, T, β) where C denotes the BahnCard purchase cost, T the validity period (in days) and $\beta \in [0, 1)$ the price reduction factor for train tickets (a ticket with an original price of p costs $\beta \cdot p$ if the BahnCard is valid on the journey date). The BahnCard 50 (BC50) mentioned above can hence be described as $(255, 365, 0.5)$. A BahnCard 25 (BC25) would be expressed as the triple $(62, 365, 0.75)$. The BahnCard problem is an archetype of an online problem with a multitude of applications (e.g. TCP acknowledgment batching). It is also a generalization of the so called ski-rental problem, where one has to decide whether to rent skis for a certain price per day or to buy (unbreakable) skis at some point. In [4, 6], it was shown that there exists a deterministic online strategy for the BahnCard problem which achieves a competitiveness factor of $(2 - \beta)$,



© Mike Timm and Sabine Storandt;
licensed under Creative Commons License CC-BY

20th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2020).

Editors: Dennis Huisman and Christos D. Zaroliagis; Article No. 2; pp. 2:1–2:13



OpenAccess Series in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Konstanzer Online-Publikations-System (KOPS)

URL: <http://nbn-resolving.de/urn:nbn:de:bsz:352-2-805k5xro2u4a7>

and an $e/(e - 1 + \beta)$ competitive randomized online strategy (with matching lower bounds for both). Hence for $\beta = 0.5$, the expected online cost is only 1.2255 times the optimal offline cost although the online strategy can only utilize incomplete information.

Several extensions of the BahnCard and the ski-rental problem have been proposed in the literature to model complex real-world scenarios better. In this paper, we introduce the multi-kind BahnCard problem where instead of a single BahnCard we have the choice between k BahnCards (with different costs and price reduction factors). Note that in Germany, there are currently three types of BahnCards available for purchase and hence strategies for the original BahnCard problem are not suitable to obtain sensible solutions.

In the following, we study the multi-kind BahnCard problem from a theoretical and practical perspective.

1.1 Related Work

In the original introduction and discussion of the BahnCard problem [4], the BahnCard was assumed to have no expiration date. In that paper, the above mentioned competitiveness factors of $(2 - \beta)$ and $e/(e - 1 + \beta)$ for a deterministic and a randomized strategy were proven, respectively. In [6], it was shown that the same competitiveness results (and matching lower bounds) hold if the BahnCard has a finite expiration date. In [1], risk-reward competitive strategies were discussed, where an agent makes a forecast about his upcoming journeys and – depending on a chosen risk level – is rewarded if that forecast is correct. The model was further extended in [2], where risk and also interest rates were considered. In [3], a problem variant with two kinds of BahnCards was studied. There, not all BahnCards are available at the same time, though, but the second BC (with a better price reduction factor) is introduced later. For this very restricted scenario an optimal $2 - \frac{\beta_2}{\beta_1}$ deterministic strategy was presented. Note that this model differs significantly from the model that we study, as in our case the BahnCards are all available for purchase at the same time and our model also allows for more than two BahnCards.

The ski-rental problem is a special case of the BahnCard problem. Here, the online problem is to decide for each day whether renting skis for a certain fee is sensible or whether skis should be bought at a given fixed price. As the skis are deemed unbreakable, there are no more decisions to make once they are purchased. This is one of the main differences to the BahnCard problem, where BahnCards expire over time and the decision when to buy a new one has to be answered repeatedly. The other difference is that in the ski-rental problem the price reduction factor can only be $\beta = 0$ as after the skis are purchased no rental fees occur at all. The BahnCard problem offers more flexibility as any β -value in $[0, 1)$ is possible there. The original ski-rental problem was proposed in [8] in the context of caching in multiprocessor systems. There, a simple optimal 2-competitive deterministic strategy was presented. In [7], an optimal randomized strategy was designed with a competitiveness factor of $e/(e - 1)$. The multi-slope ski-rental problem is an extension where one has the choice between buying skis as well as several lease options (e.g. after an initial fee of 100€, the skis can be leased for 10€ per day) which makes the problem more similar to the BahnCard problem. An e -competitive online randomized strategy for this problem was presented in [9]. In [10], the ski-rental problem with k discount options was discussed (the longer the rental duration the larger the discount) and a 4-competitive deterministic online strategy was described. Moreover, it was proven that no deterministic algorithm can have a smaller competitiveness ratio for sufficiently large choices of k . An alternative analysis for the ski-rental problem was conducted in [5], where not the worst case competitiveness ratio but the average-case competitiveness ratio was considered.

1.2 Contribution

We introduce the multi-kind BahnCard problem which is a generalization of the classical BahnCard problem, and establish the following results:

- We present an efficient graph-based algorithm for computing the optimal solution for the offline problem variant, where the stream of future journeys is known in advance. This enables us to experimentally evaluate the quality of online strategies.
- In our theoretical analysis, we determine the competitiveness factors of three online strategies. We show that there indeed exists a simple deterministic strategy that has bounded competitiveness. (For example, for the BahnCards currently available in Germany, the strategy is 4-competitive.)
- We motivate and design several other deterministic and randomized online strategies, and compare them in an experimental study. In our experiments, we consider real-world BahnCards as well as artificial settings with up to $k = 10$ BahnCards. Furthermore, we model different passenger profiles (e.g. commuter, business traveller) and empirically determine the best online strategy for each of them.

2 Formal Problem Definition

In an instance of the multi-kind BahnCard problem, we are given k BahnCards $BC_i = (C_i, T, \beta_i)$ for $i = 1, \dots, k$ where $C_i \in \mathbb{R}^+$ is the individual purchase price and $\beta_i \in [0, 1)$ the ticket price reduction factor within the validity period $T \in \mathbb{N}$. We assume $C_i \geq 1$ for $i = 1, \dots, k$, that is, BahnCards can not be arbitrarily cheap. Note that in compliance with the current standard real-world BahnCards and for ease of exposition, we assume that all BahnCards BC_1, \dots, BC_k have the same validity period T (in days). W.l.o.g we assume that $C_i < C_{i+1}$ and $\beta_i > \beta_{i+1}$. This can safely be assumed for uniform T as any BahnCard for which another BahnCard with lower or equal cost and an equal or lower reduction factor exists would never be a sensible purchase option.

The train journeys are given as a stream $\sigma = \sigma_1, \dots, \sigma_n$, each represented by a tuple $\sigma_j = (t_j, p_j)$, $j = 1, \dots, n$ where $t_j \in \mathbb{N}$ denotes the departure date and $p_j \in \mathbb{R}^+$ the price. Again we assume $p_j \geq 1$ to exclude arbitrarily cheap journeys. We always assume that $t_j < t_{j+1}$ holds, as multiple journeys on the same day can simply be accumulated into a single one by summing up their prices.

The multi-kind BahnCard problem then consists of deciding which kinds of BahnCards should be purchased on which dates. Hence the output is a set of tuples $\{(\tau_1, id_1), \dots, (\tau_l, id_l)\}$ where $\tau_i \in \mathbb{N}$ is the purchase date and $id_i \in \{1, \dots, k\}$ the index of the respective BahnCard. The induced costs are the summed costs for purchasing the chosen BahnCards $\sum_{i=1}^m C_{id_i}$ plus the summed (reduced) journey prices. We say a BahnCard BC is valid at time t if it was purchased on date τ and $t \in [\tau, \tau + T - 1]$ where T is the validity period of that BahnCard. Accordingly, a journey σ_j with price p_j induces a cost of p_j if there is no valid BahnCard at the departure date t_j . Otherwise, let $B_t \subseteq \{1, \dots, k\}$ be the set of BahnCards valid at time t . Then journey σ_j has an induced cost of $\min_{i \in B_t} \beta_i p_j$. That means, BahnCard reduction factors do not stack but the reduced price is determined by the valid BahnCard with the best reduction factor (as it is the case for real-world BahnCards as well).

In the offline multi-kind BahnCard problem, the journey stream is known in advance. In the online multi-kind BahnCard problem, at any date t only the journeys j with $t_j \leq t$ are known and the decision about buying or not buying a BahnCard (and which kind) on this date has to be made solely based on the known prefix of the journey stream and the past BahnCard purchase decisions.

3 An Optimal Offline Algorithm

In the offline problem variant, the dates and prices of all upcoming journeys are available in advance which allows to make fully informed decisions.

3.1 Graph-Based Algorithm for the One-Kind BahnCard Problem

For the classical BahnCard problem with only a single BahnCard (C, T, β) , a graph-based approach to deduce the best offline algorithm for any journey stream σ was described in [4]. The weighted journey-graph $G(V, E)$ is constructed as follows. Each journey σ_j is represented as a node $v_j \in V$ for $j = 1, \dots, n$. Furthermore, a dummy node v_{n+1} is introduced with a corresponding dummy date $t_{n+1} = \infty$. Then consecutive journeys in the stream are connected via directed edges in G ; more precisely the edges (v_j, v_{j+1}) are contained in E for $j = 1, \dots, n$ with cost p_j , respectively. To model the option to buy a BahnCard on every date on which some journey happens additional edges are introduced. Observe that it does never make sense to purchase a BahnCard on a date without a journey, as then shifting the purchase to the next upcoming journey would allow to use the respective BahnCard further into the future without increasing any costs. A BahnCard purchased on the departure date t_j of journey σ_j is valid up to date $t_j + T - 1$. Let $\sigma_{q>j}$ be the journey with the earliest departure date that does exceed $t_j + T - 1$, then the edge (v_j, v_q) with costs $C + \sum_{l=j}^{q-1} \beta \cdot p_l$ is added to E . As for every journey node there are now two outgoing edges (one modelling to not buy a BahnCard on the respective departure date and the other to buy it), the total graph size is in $\mathcal{O}(n)$. The cost of a shortest path from v_1 to v_{n+1} in this graph then equals the optimal cost achievable for the offline BahnCard problem. As the graph is a directed acyclic graph, this shortest path can be computed in linear time in the number of journeys.

3.2 Extension to Multi-Kind BahnCards

What changes if k different BahnCards are available for purchase? We make the following crucial observation: *In a solution for the multi-kind BahnCard problem, it can be optimal to purchase a BahnCard while another BahnCard is still valid.* An example is given in Table 1.

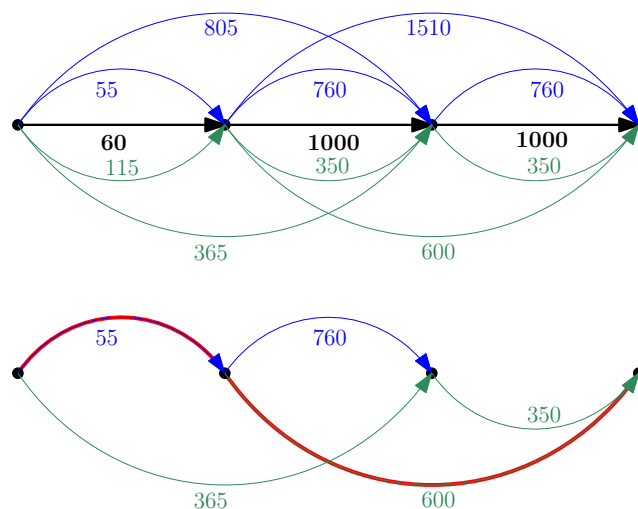
Note that this is a significant difference to the one-kind BahnCard problem, where it never makes sense to purchase a new BahnCard before the old one expired. Accordingly, it is not enough to simply extend the above described graph by one edge per journey and BahnCard type. Instead, we also have to insert edges that model the decision to let a valid BahnCard be replaced by a better one. For this purpose, we add for all BahnCards BC_i for $i = 1, \dots, k$ and all journeys σ_j an edge $(v_j, v_{q>i})$ to all nodes where $t_q \in [t_j + 1, t_j + T - 1]$ and to the first node where $t_q > t_j + T - 1$ with costs $C_i + \sum_{l=i}^{q-1} \beta_i \cdot p_l$, respectively. Note that this introduces parallel edges of which of course only the cheapest one has to be kept in the graph. Furthermore note, that for the BahnCard BC_k with the best reduction factor, it indeed never makes sense to buy another BahnCard before this one expires. Therefore, for this BahnCard only the edges as described for the one-kind BahnCard model have to be added. This makes our approach a valid generalization of the graph construction for the one-kind BahnCard problem, i.e. for $k = 1$ we get the same graph as described in [4]. But we can now deal with arbitrarily large values of k as well. In Figure 1, the graph for the example discussed in Table 1 is shown before and after edge pruning.

In the worst case, graph construction takes $\mathcal{O}(kn \cdot \min\{T, n\})$ time and the number of graph edges is in $\mathcal{O}(n \cdot \min\{T, n\})$ after pruning parallel edges. The latter is then also the time to compute the shortest path.

■ **Table 1** Example with two BahnCards and three journeys. The optimal solution is to first buy BC_1 on date 1 and then BC_2 on date 300, which induces a total cost of $10 + 0.75 \cdot 60 + 100 + 0.25 \cdot 1000 + 0.25 \cdot 1000 = 655$. Note that at the moment BC_2 is purchased, BC_1 is still valid.

	C_i	T_i	β_i
BC_1	10	365	0.75
BC_2	100	365	0.25

σ_1	σ_2	σ_3
(1,60)	(300,1000)	(400,1000)



■ **Figure 1** Graph visualization for the example instance described in Table 1. In the upper image, the thick black edges represent the individual journeys with their original prices. The blue edges encode the possibilities to buy BC_1 including edges that model premature expiration. The green edges encode the respective possibilities for BC_2 . In the lower image, the pruned graph is shown. Firstly, all edges which encode premature expiration of BC_2 were discarded, as for the BahnCard with the best reduction factor those are not necessary. Secondly, among the remaining parallel edges, all but the cheapest one were discarded. The shortest path from the leftmost to the rightmost node (with a cost of 655) is depicted in red.

4 Online Strategies with Provable Competitiveness

In this section, we will analyze three online strategies and investigate their competitiveness with respect to the optimal offline solution. As the competitiveness usually depends on the characteristics of the available BahnCards (similarly to the one-kind BahnCard problem), we will discuss the implications of our results considering the real BahnCards currently available in Germany. Their characteristics are summarized in Table 2.

For the standard BahnCard problem, the deterministic strategies ALWAYS, NEVER and SUM were analyzed in [4, 6]. The ALWAYS strategy is to buy a BahnCard whenever there is a journey on the current date but the last BahnCard already expired. The NEVER strategy is to never buy a BahnCard. The SUM strategy is to sum up the ticket prices of the journeys until they exceed a certain threshold and to then buy a BahnCard. We will now consider generalizations of these strategies for the multi-kind BahnCard problem.

■ **Table 2** Characteristics of German BahnCards.

	C	T	β
BC25	62	365	0.75
BC50	255	365	0.50
BC100	4395	365	0.00

4.1 Always-Top-Algorithm

The Always-Top-Algorithm (AT) always buys the BahnCard BC_k (which has the best reduction factor β_k) if there is a journey on the current date but there is no valid BahnCard at this moment.

► **Lemma 1.** *The AT-algorithm is $C_k + 1$ competitive.*

Proof. Let σ be the stream of journeys and let an interval $I = [t_j, t_q]$ denote a time period such that the AT algorithm bought a BC_k at time t_j , and the journey q is the departure date of the last journey which is still within the validity period of that purchased BahnCard. The induced costs of the AT algorithm in interval I can hence be expressed as $c_{AT}^I = C_k + \beta_k \cdot \sum_{t_j \in I} p_j$.

If the optimal offline solution also purchases a BC_k somewhere within I , then $c_{AT}^I \leq c_{OPT}^I$ holds. For the worst case analysis, we hence assume that the optimal strategy does not include the purchase of BC_k in I , but either the purchase of other BahnCards with a smaller reduction factor or no BahnCard purchase at all. The competitiveness can be expressed as $\frac{C_k}{c_{OPT}^I} + \frac{\beta_k \cdot \sum_{t_j \in I} p_j}{c_{OPT}^I}$. We observe that the second term cannot be bigger than 1 as the optimal solution does only achieve a reduction factor $\geq \beta_{k-1}$. The first term cannot become larger than C_k as the denominator c_{OPT}^I either contains the purchase cost of a BahnCard or an unreduced ticket price, and hence cannot be smaller than 1. Combining both terms, we get an upper bound on the competitiveness of $C_k + 1$. ◀

Considering the real-world BahnCards given in Table 2, the algorithm would always buy the BahnCard BC100 which reduces the ticket prices to 0. According to our analysis, this results in a competitiveness factor of 4396.

4.2 Never-Algorithm

The Never-Algorithm never buys any BahnCard regardless of the journey stream σ . The costs can thus be expressed as $c_{NEVER} = \sum_{j=1}^n p_i$. In the worst case, the optimal solution would be to use the Always-Top-Algorithm described above, i.e. the accumulated ticket price are always large enough such that it is worth to buy the most expensive BahnCard with the best reduction factor, resulting in $c_{OPT} = C_k + \beta_k \cdot \sum_{j=1}^n p_j$. Accordingly, the competitiveness factor is unbounded. Especially for $\beta_k = 0$ the ratio of c_{NEVER} and c_{OPT} grows proportional to the summed unreduced ticket prices, and therefore a constant upper bound on this ratio cannot be determined. Looking at the real-world setting from Table 2, we observe that the worst case value $\beta_k = 0$ indeed is assumed here for the BC100.

4.3 B-SUM-Algorithm

The above considerations imply that to achieve some practically useful competitiveness, the online strategy should make the decision when to purchase a BahnCard more carefully.

We now investigate the so called B-SUM-algorithm which is an extension to the $(2 - \beta)$ -competitive SUM-algorithm for the one-kind BahnCard problem. The idea behind this algorithm is to always buy the most expensive BahnCard BC_k once the accumulated costs of the previous journeys (where we had no valid BahnCard) reach the critical value of said BahnCard. The critical value is defined as $crit_k = \frac{C_k}{1 - \beta_k}$. The intuition is that if accumulated ticket prices in an interval equal $crit_k$, the induced costs of not having a BahnCard and having purchased BahnCard BC_k at the beginning of the interval are the same.

► **Theorem 2.** *The B-SUM algorithm is $\frac{2}{\beta_{k-1}}$ -competitive.*

Proof. Let $\tau_1, \tau_2, \dots, \tau_q$ bet the dates on which the optimal offline algorithm buys a BC_k . This induces consecutive intervals $[0, \tau_1), [\tau_1, \tau_2), \dots, [\tau_{q-1}, \tau_q), [\tau_q, \infty)$. If the optimal algorithm never buys a BC_k then there is only a single interval $[0, \infty)$.

We now want to compare the cost of the B-SUM algorithm in each interval $I = [\tau_i, \tau_{i+1})$ with the optimal cost in that interval. In all but the first interval, the optimal solution buys a BC_k . Note that this can only be optimal if the accumulated ticket prices in interval I exceed $crit_k$. Now we consider B-SUM. We subdivide interval I in four subintervals I_1, I_2, I_3, I_4 (some of them possibly empty). In I_1 , B-SUM still has a valid BC_k purchased before τ_i . In I_2 , B-SUM has no valid BahnCard. At the beginning of I_3 , B-SUM purchases a BC_k . Its expiration then initializes interval I_4 . Note that the optimal strategy does not have a valid BC_k in I_4 as well as the optimal strategy purchased its BC_k earlier than B-SUM and another purchase of a BC_k at time τ_{i+1} marks the beginning of a completely new interval.

We first consider only the intervals I_1, I_2, I_3 . The cost of the optimal solution is lower bounded by $c_{OPT} \geq C_k + \beta_k \sum_{t_j \in I_1, I_2, I_3} p_j$. For B-SUM, we have $c_{B-SUM} = \beta_k \sum_{t_j \in I_1} p_j + \sum_{t_j \in I_2} p_j + C_k + \beta_k \sum_{t_j \in I_3} p_j$. Therefore, the ratio of c_{B-SUM} and c_{OPT} is maximized if $\sum_{t_j \in I_2} p_j$ is as large as possible. But as B-SUM buys a new BC_k as soon as the accumulated ticket price since the last expiration exceed $crit_k$, we conclude that $\sum_{t_j \in I_2} p_j < crit_k = \frac{C_k}{1 - \beta_k}$. Plugging this in, we get a competitiveness of $(2 - \beta_k)$ in compliance with the one-kind BahnCard problem. But we still have to consider I_4 . In I_4 , as observed above, the optimal strategy does not have a valid BC_k . Therefore, we can lower bound the optimal costs in I_4 as $c_{OPT} \geq \beta_{k-1} \sum_{t_j \in I_4} p_j$. In case the summed ticket prices in I_4 are smaller than $crit_k$, B-SUM will not purchase another BC_k and hence its cost in I_4 is $\sum_{t_j \in I_4} p_j$, leading to a competitiveness ratio of $\frac{1}{\beta_{k-1}}$. If the accumulated costs in I_4 however exceed $crit_k$, then B-SUM purchases BC_k again. Let $b \geq 1$ be the number of BahnCards B-SUM purchases in I_4 . Then the critical value was exceeded b times, leading to a lower bound of $c_{OPT} \geq \beta_{k-1} \cdot b \cdot \frac{C_k}{1 - \beta_k}$. The costs for B-SUM are upper bounded by $c_{B-SUM} \leq bC_k + b \frac{C_k}{1 - \beta_k}$. The ratio of those two is then upper bounded by $\frac{2 - \beta_k}{\beta_{k-1}} \leq \frac{2}{\beta_{k-1}}$. The same analysis applies to the very first interval $[0, \tau_1)$ in which the optimal strategy does not have a valid BahnCard BC_k as well.

Therefore, the competitiveness of the B-SUM algorithm is $\max(2 - \beta_k, \frac{2}{\beta_{k-1}})$ which is dominated by the latter. ◀

As according to our model $\beta_{k-1} > \beta_k \geq 0$ holds, the competitiveness factor is finite for all possible β_{k-1} . Using values of the real-world BahnCards described in Table 2, we see that $\beta_{k-1} = 0.5$ and hence the resulting competitiveness of B-SUM is 4.

5 Heuristic Online Strategies

In this section we will present further online strategies for the multi-kind BahnCard problem. While those do not come with provable competitiveness guarantees, we will observe their instance-based competitiveness in various scenarios in the experiments later on.

5.1 Choosing a BahnCard u.a.r in T (RU-INT)

The RU-INT algorithm either buys one of the k BahnCards or no BahnCard uniformly at random in each time period T . Although this approach might produce arbitrarily bad solutions, we expect it to be better than the Always-Algorithm and Never-Algorithm making on average. In addition, RU-INT also offers a baseline for the other heuristics since we obviously aim to be superior to random purchases.

5.2 Summing up in T (SUM-INT)

The SUM-INT algorithm sums up the costs for a journey stream σ over one validity period T of the BahnCards and then buys the BahnCard with the highest critical value reached for the next period T . Then the algorithm repeats. This means the algorithm will alternate between buying a BahnCard and not buying any BahnCard each interval.

This algorithm is sensible if the traveller has similar travelling habits in consecutive years as then every second year the perfect BahnCard is chosen.

5.3 Critical single journeys (S-CRIT)

The S-CRIT algorithm always checks if any BahnCard would be profitable for a single journey (the current journey) and then buys the most fitting one according to the critical value, i.e., the one with the highest index out of those that have reached the critical value. Note that it is not necessarily the case that the critical values are monotonically increasing with the index i . Although we have $C_i < C_{i+1}$ and $\beta_i > \beta_{i+1}$, it could happen that $\frac{C_i}{1-\beta_i} > \frac{C_{i+1}}{1-\beta_{i+1}}$. Accordingly, there might be BahnCards that the S-CRIT algorithm never purchases, as a BahnCard with better price reduction factor and lower critical value exists. For the real-world BahnCards described in Table 2, though, the critical values are 248€, 510€, 4395€ in that order and hence S-CRIT could choose any of them depending on the ticket price. If no critical value is reached the algorithm does not buy a BahnCard and proceeds to the next journey. This approach makes sense for travellers with very few but very expensive journeys.

5.4 Continuing with the reduced costs of the previous interval (RED-CRIT)

The RED-CRIT algorithm sums up the journey costs as long as no critical value of any BahnCard is reached. Once a critical value is reached it buys the most fitting BahnCard according to the critical value (i.e., the one with the highest critical value that was reached), sets the current costs to the summed up reduced costs of the journeys in the validity period of the chosen BahnCard, and starts again by checking if a critical value is reached. This approach makes sense because we buy a BahnCard once it would have been profitable to do so and then use the costs during that period to take the traveller's habits into consideration for the next interval.

6 Experimental Evaluation

To evaluate the competitiveness of the algorithms we conduct experiments with different traveller profiles comparing the results to the optimal offline solution. We begin by using the real-world BahnCards and extend the experiments to randomly generated BahnCards. All experiments are executed on a desktop PC with an Intel(R) Core(TM) i7-6700K processor (4 cores @ 4.00Ghz) and 64GB DDR4 RAM.

6.1 Profiles

We will consider three main train traveller types: commuters, occasional travellers, and businessmen. The traveller profiles are realized as vectors where an entry represents the number of journeys on that day, i.e., a zero on days where no journeys occur, a one on a day where only one journey occurs and a two for the commuters accounting for the way to work and the way back home. Note that multiple journeys happening on the same day are just regarded as a single journey with aggregated costs in all our algorithms, as we assume that a BahnCard is either valid for the full day or not valid at all on that day. Therefore, more fine-grained information – as the exact time of the ticket purchase – is not relevant here. The following profiles were used to create different scenarios:

- **The commuter.** We distinguish between a low price, a mid price and a high price commuter. Journeys happen on workdays and always cost 5€, 15€ or 35€(one-way), respectively. Thereby, each journey has a 95% chance to happen.
- **The occasional traveller.** Journeys can happen on every day with a 1% chance and costs range between 50€ and 1000€.
- **The businessman.** Journeys can happen on every day with a 10% chance and costs range between 50€ and 1000€.

6.2 Results for Real-World BahnCards

For each profile we created five vectors with the length of $y \in \{2, 5, 10, 20, 40\}$ years (multiplied by 365 to have an entry for each day), computed the BahnCard schedule for every heuristic and calculated the competitiveness ratios by comparing them to the optimal solution. This process has been done 20 times for each parameter pair (year, profile) and the means of the ratios has been taken to gain insight on the general performance of the algorithms in different scenarios. We will now look at each profile's results starting with the commuters.

6.2.1 Low price commuter

For the the low price commuter profile RED-CRIT always performs the best with an average competitiveness of 1.0653 while B-SUM always has the worst average competitiveness ratio (on average 2.3196). For B-SUM this makes sense, because we always buy the most expensive BahnCard and as mentioned before a journey for this profile always costs 5€ meaning the most expensive BahnCard will most likely be too expensive and a cheaper one would have been more profitable.

6.2.2 Mid price commuter

Contrary to the low price commuter the B-SUM algorithm performs much better with higher prices as explained before. Surprisingly the RU-INT algorithm does fairly well in this scenario. Looking at the experiments more closely though, this can be explained by the fact that the optimal solution actually buys the BahnCard 50 at the start of every year in the schedules of the mid price commuter. Given the fact that there are only three BahnCards to choose from with the BahnCard 50 being the best one in every year the chances of being significantly worse than the optimal solution are not that high. Despite all that RED-CRIT prevails as the best algorithm in this scenario as shown in Table 3.

2:10 On the Multi-Kind BahnCard Problem

■ **Table 3** Average competitiveness ratios for the mid price commuter scenario and real.world BahnCards.

Years	SUM-INT	B-SUM	S-CRIT	RU-INT	RED-CRIT
2	1.4969	1.4907	1.8737	1.3285	1.2961
5	1.5685	1.4150	1.8740	1.3080	1.1844
10	1.4943	1.4133	1.8738	1.3130	1.1468
20	1.4917	1.4418	1.8740	1.2800	1.1262
40	1.4921	1.3932	1.8742	1.3251	1.1159

6.2.3 High price commuter

In the case of the high price commuter we observe another increase in competitiveness for the B-SUM algorithm (average 1.6831) which again makes sense because the overall costs have increased as well. Likewise the competitiveness of the S-CRIT algorithm decreased heavily from the low price to the high price commuter (average from 1.6642 to 3.9521). This is to be expected though, since none of the journeys have a high enough price to reach the critical value of any BahnCard, thus the algorithm never buys a BahnCard which will get worse as the overall costs increase. Again the overall best choice is RED-CRIT (average 1.3602).

6.2.4 The occasional traveller

Due to the sparseness of journeys of the occasional traveller profile the S-CRIT algorithm almost performs as well as the RED-CRIT algorithm (averages 1.0945 versus 1.0697) with the latter again being the overall best choice. But as the journey streams are more diverse for the occasional traveller than e.g. for the commuters, we also observe larger variations in the performance of the different strategies. Figure 2 shows an example illustration for a 2-year period.

6.2.5 The businessman

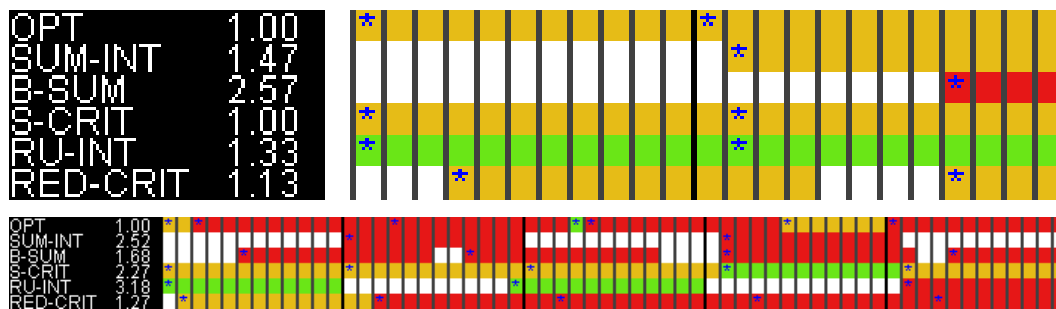
For the businessman profile, RED-CRIT again was the best approach (average 1.3237). Interestingly though, the second best strategy in this scenario appears to be B-SUM (average 1.7721). This is apparently the case because the high ticket prices make up for the lack of journeys. The other heuristics performed worse compared to the previously considered profiles. Figure 2 shows an example illustration for a 5-year period.

6.3 Results for Artificial BahnCards

After analyzing the competitiveness ratio of the algorithms in respect to the real-world BahnCards we will now look at three different scenarios with randomly generated BahnCards:

1. BahnCards with evenly distributed β
2. BahnCards with similar β
3. BahnCards with heavily differing β

For each model we drew ten betas and computed the price by choosing a base price of $base = 40$ and taking the result of $\frac{base}{\beta - \frac{\beta}{5}}$ as the price of the respective BahnCard to gain reasonably realistic costs.



■ **Figure 2** Compressed overview of the different strategies for the occasional traveller in a 2-year and the business man in a 5-year period (for different journey streams). Each column indicates a month. BahnCard purchases are marked by stars. A cell is coloured green if in the respective month there was a valid BC25, yellow for BC50, and red for BC100.

■ **Table 4** Average competitiveness ratios for the mid price commuter and evenly distributed BahnCards.

Years	SUM-INT	B-SUM	S-CRIT	RU-INT	RED-CRIT
2	2.9744	3.1245	4.3332	3.0584	1.8038
5	3.2434	2.5167	4.3330	3.1507	1.6830
10	2.9712	2.5191	4.3323	3.1763	1.6460
20	2.9732	2.3736	4.3300	3.0469	1.6284
40	2.9723	2.3686	4.3317	2.9640	1.6155

6.3.1 BahnCards with evenly distributed β

For this model we choose $k = 10$ intervals of same size, e.g., for $i \in \{0, \dots, k-1\}$ the interval is $(1 - \frac{i+1}{k}, 1 - \frac{i}{k}]$ and pick a β from each of the intervals uniformly at random. This results in evenly distributed betas in $(0, 1]$.

Overall this model produced results very similar to the real-world BahnCards with the exception that the competitiveness ratios were generally worse. This is illustrated by the mid price commuter example in Table 4. While the S-CRIT algorithm performed the worst in the real-world counterpart it did not perform nearly as bad as in this model. Even with a BahnCard having a β of 0.9149 the journeys of the mid price commuter did not reach the critical value and thus the S-CRIT algorithm again performed like the Never-Algorithm not buying any BahnCard at all.

This leads to the conclusion that the wider variety of BahnCards causes the ratios to be worse overall as the optimal solution has even more profitable choices than in the real-world example.

6.3.2 BahnCards with similar β

For this model we choose k intervals of same size, e.g., for $i \in \{0, \dots, k-1\}$ the interval is $(1 - \frac{i+1}{k}, 1 - \frac{i}{k}]$ and choose one interval uniformly at random to draw k betas from. This results in very similar BahnCards. In this model the solutions provided by the algorithms are very close to the optimal solution across all the traveller profiles, the worst ratio being 1.0740 meaning the choice of BahnCard has very little impact on the competitiveness ratio (as to be expected for BahnCard with only minor differences).

■ **Table 5** Average competitiveness ratios for the businessman scenario and heavily differing BahnCards.

Years	SUM-INT	B-SUM	S-CRIT	RU-INT	RED-CRIT
2	9.7906	9.7523	2.0797	8.1918	4.7597
5	9.7403	9.6896	2.5465	8.5503	4.9725
10	9.8425	9.7971	2.7149	8.1206	5.8743
20	9.7691	9.7239	2.3683	8.3782	7.4911
40	9.8559	9.8106	2.6068	8.5319	8.8573

6.3.3 BahnCards with heavily differing β

For this model we choose k intervals of same size, e.g., for $i \in \{0, \dots, k-1\}$ the interval is $(1 - \frac{i+1}{k}, 1 - \frac{i}{k}]$ and draw $\lceil \frac{k}{2} \rceil$ betas from the first interval and $\lfloor \frac{k}{2} \rfloor$ betas from the last interval. This results in a bimodal distribution of BahnCards with a heavy gap between the two partitions.

Of all the models this one produced the worst competitiveness ratios almost reaching an average of 10 in some cases as can be seen in Table 5 regarding the businessman scenario. In this scenario S-CRIT seems to be the best algorithm with a competitiveness ratio of around 2.5 on average. Contrary to all the other algorithms S-CRIT buys cheap BahnCards just like the optimal solution leading to a fairly good competitiveness while buying one of the more expensive BahnCards (drawn from the first interval) has a very detrimental effect.

7 Conclusions and Future Work

In this paper, we have extended the classical BahnCard problem to the multi-kind BahnCard problem. We presented a simple online strategy with provable competitiveness but showed that in practical scenarios custom-tailored heuristic strategies are often superior. An obvious open question is whether there are other strategies with provably better competitiveness. In particular, a strategy that ensures a constant competitiveness independent of the purchase costs and price reduction factors of the BahnCards would be worth investigating. Further, the scenario where the validity periods of the BahnCards are allowed to differ would be of theoretical and practical interest. Indeed, bus tickets valid for a week or a month could also be seen as realizations of a BahnCard with a price reduction factor of $\beta = 0$. Incorporating different validity periods in the optimal offline algorithm is straightforward. But the design and analysis of the online strategies would be affected. In addition, it would be interesting to extend the model even further. For example, in Germany, certain special offer discounts can only be combined with the BahnCard 25 but not with the BahnCard 50, which affects the competitiveness of our proposed strategies. There are also non-standard types of BahnCards where the validity period depends on certain events (e.g. the so called Sieger BahnCard was only valid during the soccer championship and only as long as the German team was not eliminated). Flexible validity periods would add yet another level of uncertainty to the model and would demand the development of novel online strategies.

References

- 1 Lili Ding, Chunlin Xin, and Jian Chen. A risk-reward competitive analysis of the bahncard problem. In *International Conference on Algorithmic Applications in Management*, pages 37–45. Springer, 2005.
- 2 Lili Ding, Yinfeng Xu, and Shuhua Hu. The bahncard problem with interest rate and risk. In *International Workshop on Internet and Network Economics*, pages 307–314. Springer, 2005.
- 3 LL Ding and YF Xu. New results for online bahncard problem. *Information, An International Interdisciplinary Journal*, 12:523–536, 2009.
- 4 Rudolf Fleischer. On the bahncard problem. In *Computing and Combinatorics*, pages 65–74, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.
- 5 Hiroshi Fujiwara and Kazuo Iwama. Average-case competitive analyses for ski-rental problems. *Algorithmica*, 42(1):95–107, 2005.
- 6 Anna R Karlin, Claire Kenyon, Dana Randall, and Dana Randall. Dynamic tcp acknowledgement and other stories about $e/(e-1)$. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 502–509. ACM, 2001.
- 7 Anna R. Karlin, Mark S. Manasse, Lyle A. McGeoch, and Susan Owicki. Competitive randomized algorithms for nonuniform problems. *Algorithmica*, 11(6):542–571, 1994.
- 8 Anna R Karlin, Mark S Manasse, Larry Rudolph, and Daniel D Sleator. Competitive snoopy caching. *Algorithmica*, 3(1-4):79–119, 1988.
- 9 Zvi Lotker, Boaz Patt-Shamir, and Dror Rawitz. Rent, lease, or buy: Randomized algorithms for multislope ski rental. *SIAM Journal on Discrete Mathematics*, 26(2):718–736, 2012.
- 10 Guiqing Zhang, Chung Keung Poon, and Yinfeng Xu. The ski-rental problem with multiple discount options. *Information Processing Letters*, 111(18):903–906, 2011.