Results from the Verification of Models of Spectrum Auctions

Elaheh Ordoni, Jutta Mülle, and Klemens Böhm

Karlsruhe Institute of Technology (KIT),
Institute for Program Structures and Data Organization,
76131 Karlsruhe, Germany {elaheh.ordoni, jutta.muelle,
klemens.boehm}@kit.edu

Abstract. The revenue gained by spectrum auctions has been an essential source of governmental income. Even though numerous studies have been conducted in auction literature, many catastrophic results occurred in the real world. In this paper, we demonstrate how one can use verification techniques to improve the design of spectrum auctions, i.e., to prevent the unexpected outcomes to happen. To do so, we model the spectrum auction in BPMN and verify certain properties of the auction model. To do so, we assign different *capacity points* to the bidders and check how it affects the revenue. A capacity point defines the maximum number of products that a bidder can win. Our study reveals which assignment of capacity points to the bidders leads to the worst auctioneer's revenue.

1 Introduction

Spectrum auction revenue is a significant source of governmental income. Germany and the U.K., for example, have earned respectively 50.8 and 37.5 billion Euros in 2000 [7]. Although auctions should be designed so that undesirable outcomes do not occur, catastrophic results have happened in several cases. In the Netherlands, a political fiasco occurred in 2000 because of the low revenue of the Dutch UMTS auction [27]. In another example in the U.S., an auction policy that raised bid prices caused a loss of 30 MHz for a decade. This flaw cost around 70 billion dollars [13]. In yet another case [2], about half of the products were left out.

Literature offers two possible ways of studying auctions: (a) experimental analyses performed in laboratories with human subjects [14], and (b) theoretical analyses. Finding undesirable outcomes with either technique continues to be challenging: When it comes to (a), laboratories perform relatively few experiments. To illustrate, to investigate all experimental designs in [4], over 13 million experiments would be necessary. No institution would be able to accommodate such a setup. In (b), researchers use auction theory to predict equilibria, relying on assumptions regarding bidding behavior [26]. In general, rational behavior of bidders is part of the assumptions [17]. But this assumption is not always valid [15]. In spite of developing frameworks for truthful bidding under interference constraints [8], bidders can still be irrational. In consequence, design errors that go unnoticed can lead to catastrophic auction results.

To detect such cases, verification techniques can be applied before executing the auctions, i.e., one can detect unexpected outcomes before they actually happen. Authors in [24], for example, have proposed a Petri-Net-based approach to verify data-value-aware process models. In such processes, values of data objects such as the price of products play an important role, and process elements can modify these values while the process is executed. Our approach allows verifying certain properties of spectrum auctions. For example, one can derive the value of the lowest *auctioneer's revenue*, i. e., the sum of the final prices of the products. The evaluation in [24] only covers one setting, i.e., selling two products to three bidders with fixed auction parameters. So this study alone does not provide much information to auction designers.

In this paper, we report on the results of a systematic analysis of a real-world application, the German 4G spectrum auction, to sell one of the most valuable bandwidths, the 800 MHz band [5]. This auction has had four bidders and has sold six products. More specifically, we study the effect of so-called *capacity point*s in this auction. A capacity point is the maximum number of licenses that a bidder can win. This parameter prevents bidders from winning too many items. With this feature, an auction designer can guarantee a certain number of bidders being awarded a good and prevent bidders from forming a monopoly. Our study focuses on capacity points for two reasons: (1) The capacity points assigned to each bidder influence the revenue of the auctioneer. (2) In contrast to other parameters like the budgets of the bidders, the auctioneer controls the value of capacity points. In other words, he or she can change their number. We study the impact of capacity points by systematically distributing different capacities among the bidders and assigning a random budget according to [4]. Studying all such distributions in combination with all different budgets that are possible is future work. In particular, we consider the following research questions:

1. Which assignments of capacity points lead to the lowest and the highest revenue of the auctioneer?
2. With a certain allocation of the capacity points, how often the lowest revenue can happen for the auctioneer?
3. Does increasing capacity points always increase the revenue?
4. What is the best assignment of capacity points to the bidders, i.e., making the worst revenue possible not too low?
5. Does changing the capacity point of a single bidder always change the auction's outcome?

To verify properties of the process model of a spectrum auction, we make use of a Petri-Net based verification technique developed in [24]. We come up with a rigorous formulation of the above questions, referred to as properties in what follows, as CTL formulas [6]. To do so, we have verified more than 2 million properties. Our findings are interesting: Varying the capacity points does not always affect the revenue. More specifically, varying the capacity points of some bidders does not have any impact on the lowest revenue, whereas varying the points of other bidders dramatically changes the results. Our method allows identifying bidders whose capacity points have a significant impact on the lowest revenue, as well as the corresponding allocation of capacity points that leads to this outcome. Next, we have observed a trade-off between the 'extent of monopolism' vs. the 'expected revenue' which the allocation of capacity points may influence. This might help the auctioneer to assign capacities in line with his/her objectives.

Paper outline: Section 2 explains SMR spectrum auctions. Section 3 features our approach. Section 4 discusses the evaluation. Section 5 covers related work and Section 6 concludes.

2 Simultaneous Multi-Round (SMR) Auctions

For more than two decades, simultaneous multi-round (SMR) auctions have been the standard format for allocating spectrum licenses to bidders [22]. This auction type allows the sale of several products, i.e., spectrum licenses. A respective auction typically consists of several rounds of bidding. Before the auction starts, the auctioneer specifies the lowest acceptable price for each product, referred to as a *reserve price*. Bidders may bid simultaneously on zero, one, or multiple products in each round. In the type of auction we analyze here, each bidder has a separate budget for each product. This budget is a reasonable reflection of the bidder's valuation of the individual product. Thus, bidders are not able to use any leftover budget for a different product. Next, bidders do not issue combined bids on different products, unlike combinatorial auctions, in which they can bid on bundles of products. Additionally, there is a so-called *capacity rule* [18]: Each bidder has a *capacity*, the maximum number of products he or she may win. In the round following the previous one, the highest bid for each product will become the reserve price. This bid is made known to all bidders, but not the other bids. Bidders also do not know the bids their competitors are issuing in the current round. Bidding for a particular product ends when no new bids are submitted in a round. The winner of a product is the bidder with the highest standing bid.

3 Our Approach

We collect a dataset describing the different outcomes of a spectrum auction in order to answer the research questions. We obtain this dataset by verifying the respective model of the spectrum auction. In the following, we describe the verification approach used. We do so because the specifics of what we can verify (both on a functional and on a non-functional level) rely on it. Figure 1 serves as an overview. (1) We model the SMR auction in BPMN notation [23]. BPMN is a suitable language for the description of spectrum auction models in a visual way and allows the subsequent analysis using the existing verification techniques. (2) We transform the SMR process model into Petri Nets. We use plain Petri Nets as a target for the mapping because of the availability of efficient analysis techniques [1]. (3) Given the resulting Petri Nets, one must specify properties of SMR auction in a formal language such as Computation Tree Logic (CTL) for verification. (4) In the final step, Activity *model checking* verifies the properties against the resulting Petri Nets and outputs the verification results.

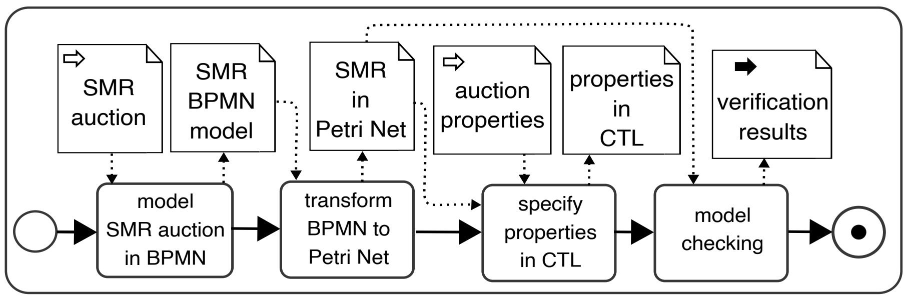

Fig. 1. Overview of verification procedure

3.1 SMR auction in BPMN model

In Fig. 2, we show a simplified version of an SMR auction in BPMN notation. The complete BPMN model is also available[1]. Observe that the model does not specify a certain bidding behavior. To do so, we issue a random bid that falls between the current price of a product and the budget of the qualified bidders. Doing without such an assumption is in some contrast to auction theory, which tends to focus on rational bidders, even though this kind of behavior is not guaranteed. We consider all possible valid bids to derive extreme outcomes of an auction, including the lowest prices that are possible. The BPMN model has three subprocesses. (1) The first subprocess examines whether a bidder can afford further products he or she has not won yet (*availability of bidders*). The auction continues with Subprocess *bidding of each bidder*. (2) Activity *place bid* issues a random bid. If a bidder has both budget and capacity left to acquire the product, he or she will always submit a bid. Activity *decrease capacity* reduces the bidder's capacity right after having won a product. Activity *remove bid* removes bids from bidders who have no capacity left. (3) Subprocess *winner determination* determines the new reserve prices and the winners. Until no more bids are submitted, these three subprocesses are repeated. Note that changing the parameter values (e.g., capacity points), but leaving the BPMN structure unchanged, gives us a different auction process model in our terminology.

[1] `https://doi.org/10.5445/IR/1000143697`

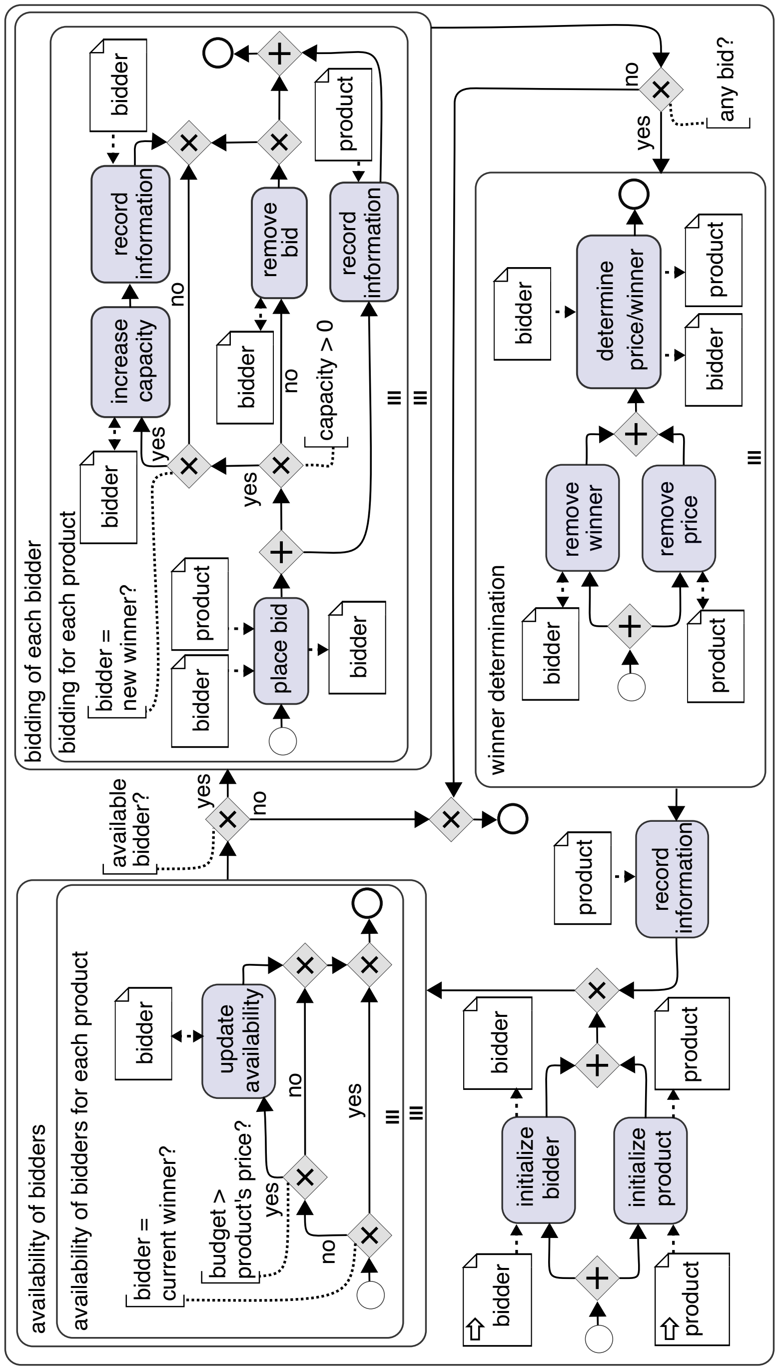

Fig. 2. Simplified process model of an SMR auction in BPMN notation.

3.2 SMR Process Model to Petri Nets

We use an existing verification framework [24], to verify properties of the SMR auction. It transforms the BPMN model of the auction into a Petri net. We use plain Petri Nets, in contrast to, say, colored Petri Nets, as the target of the mapping. This is because efficient analysis techniques are available [19]. Another reason is that plain Petri nets provide counterexamples when they verify a property. This makes it relatively easy for the designer of the model to detect where the unintended behavior of the process occurs and to fix it as necessary. To illustrate, think of spectrum auctions. When the model checker finds a path leading to the lowest revenue, this helps the auction designer with that chore.

3.3 Specification of Properties

The result of the transformation just described is a Petri Net representing the semantics of the use of data values in the process. To verify a spectrum auction, one must specify properties in a formal language such as Computation Tree Logic (CTL) [6]. In the following, we show how such properties referring to data values can be defined in CTL.

Definition 1. *(Data Property [24]). A Data Property ϕ is a CTL formula in which an atomic formula refers to either an activity/event, or a data value in a process model.*

Example 1. The data property for the question: "Can *product*.2 have a *price* of 2 at the end of the auction?" is:

$$EF(product.1.price.2 \wedge e.end).$$

In this formula, "*product*.2.*price*.2" is the "price" of 2 for "product.2". The atomic formula "*e.end*" is an end event, i. e., represents the end of the process.

To detect the lowest revenue of the auctioneer, we first find the lowest final *price* of *products*, starting with the reserve prices, using the property in Example 1. In case this property is not satisfied, we now verify a new property with an increased price. We continue increasing prices until there is a state that fulfills the property. Next, we detect the winner who won a product at a certain price. To do so, for each bidder who has a budget equal to or higher than the final price of the product, we check whether they can be the winner.

Example 2. Suppose that the lowest price for *product*.1 is 4, and the budget of *bidder*.1 is higher than 4. Then the property to check is:

$$EF(p.product.1.price.4 \wedge product.1.bidder.1 \wedge e.end).$$

In this property, "p.product.1.price.4" fixes the price for *product*.1 to 4, and "product.1.bidder.1" expresses that *product*.1 belongs to *bidder*.1, i.e., *bidder*.1 wins this product.

If the model checker can find a state, i. e., an execution path fulfilling the property, we record the bidder whose budget is sufficient as the winner of the product and continue to check whether other bidders can be the winner as well. In case two bidders (e.g., *bidder*.1 and *bidder*.2) are the potential winners for a certain product, we continue with both cases. The first case is to fix *bidder*.1 as the winner, decrease his/her capacity points and verify the price of other products, i.e., to check whether the other products can be sold for a certain price. The second case is to identify *bidder*.2 as the winner, decrease her/his capacity point, and continue verifying the other products.

4 Evaluation

In the following, we describe the auction parameters used in the process models to be verified, Section 4.1. In Section 4.2, we first report on characteristics of the verification procedure itself and then describe the dataset obtained that we will then use to answer those research questions. Using the obtained dataset results, we address the research questions listed in Section 1; see Section 4.3.

4.1 Evaluation Setting

The SMR auction model evaluated here consists of 4 bidders and 6 products. This is exactly in line with the German 4G spectrum auction to allocate licenses belonging to the 800 MHz band. Having four different bidders in the auction results in $3^4 = 81$ different assignments of capacity points. Each assignment results in a different process model. However, assignments where the sum of capacity points is less than 6 cannot happen. This is because the number of products to be auctioned off is 6, so the sum of capacity points must be at least 6 to sell all of them. After reducing such assignments, we get 76 pairwise different process models, i.e., bidders have different capacity points. Table 1 lists the capacity points of the bidders in each process model. At first, we keep the capacity points of *Bidder*.1 at 1 and vary the capacity points of the other bidders (Processes 1 to 22). In order to do this, we keep the capacity points of *bidder*.2 at 1, and we vary the capacities of *bidder*.3 and *bidder*.4 (Processes 1 to 6). Therefore, we keep the 's capacity points of *bidder*.3 at 1 and change the ones of bidder.4 from 1 to 3. The capacity points of 2 and 3 are distributed the same among the bidders. To each bidder we have assigned a random budget for a certain product in the range of [2..10], similarly to [4].

4.2 Verification of Properties

The number of properties to be verified in each process model varies between 24,844 and 29,880. Namely, verifying the price of a certain product might be fast, i.e., without having to verify many properties. For example, when the lowest final price is 2, the higher prices do not need to be verified. However, when the bidders have higher budgets and enough capacity points left, the competition is harder

Table 1. The distribution of capacity points in each process model

process model	capacities	process model	capacities	process model	capacities
Process 1	[1, 1, 1, 3]	Process 27	[2, 1, 2, 2]	Process 53	[3, 1, 2, 1]
Process 2	[1, 1, 2, 2]	Process 28	[2, 1, 2, 3]	Process 54	[3, 1, 2, 2]
Process 3	[1, 1, 2, 3]	Process 29	[2, 1, 3, 1]	Process 55	[3, 1, 2, 3]
Process 4	[1, 1, 3, 1]	Process 30	[2, 1, 3, 2]	Process 56	[3, 1, 3, 1]
Process 5	[1, 1, 3, 2]	Process 31	[2, 1, 3, 3]	Process 57	[3, 1, 3, 2]
Process 6	[1, 1, 3, 3]	Process 32	[2, 2, 1, 1]	Process 58	[3, 1, 3, 3]
Process 7	[1, 2, 1, 2]	Process 33	[2, 2, 1, 2]	Process 59	[3, 2, 1, 1]
Process 8	[1, 2, 1, 3]	Process 34	[2, 2, 1, 3]	Process 60	[3, 2, 1, 2]
Process 9	[1, 2, 2, 1]	Process 35	[2, 2, 2, 1]	Process 61	[3, 2, 1, 3]
Process 10	[1, 2, 2, 2]	Process 36	[2, 2, 2, 2]	Process 62	[3, 2, 2, 1]
Process 11	[1, 2, 2, 3]	Process 37	[2, 2, 2, 3]	Process 63	[3, 2, 2, 2]
Process 12	[1, 2, 3, 1]	Process 38	[2, 2, 3, 1]	Process 64	[3, 2, 2, 3]
Process 13	[1, 2, 3, 2]	Process 39	[2, 2, 3, 2]	Process 65	[3, 2, 3, 1]
Process 14	[1, 2, 3, 3]	Process 40	[2, 2, 3, 3]	Process 66	[3, 2, 3, 2]
Process 15	[1, 3, 1, 1]	Process 41	[2, 3, 1, 1]	Process 67	[3, 2, 3, 3]
Process 16	[1, 3, 2, 1]	Process 42	[2, 3, 1, 2]	Process 68	[3, 3, 1, 1]
Process 17	[1, 3, 1, 3]	Process 43	[2, 3, 1, 3]	Process 69	[3, 3, 1, 2]
Process 18	[1, 3, 2, 1]	Process 44	[2, 3, 2, 1]	Process 70	[3, 3, 1, 3]
Process 19	[1, 3, 2, 2]	Process 45	[2, 3, 2, 2]	Process 71	[3, 3, 2, 1]
Process 20	[1, 3, 2, 3]	Process 46	[2, 3, 2, 3]	Process 72	[3, 3, 2, 2]
Process 21	[1, 3, 3, 1]	Process 47	[2, 3, 3, 1]	Process 73	[3, 3, 2, 3]
Process 22	[1, 3, 3, 2]	Process 48	[2, 3, 3, 2]	Process 74	[3, 3, 3, 1]
Process 23	[2, 1, 1, 2]	Process 49	[2, 3, 3, 3]	Process 75	[3, 3, 3, 2]
Process 24	[2, 1, 2, 1]	Process 50	[3, 1, 1, 1]	Process 76	[3, 3, 3, 3]
Process 25	[2, 1, 2, 2]	Process 51	[3, 1, 1, 2]		
Process 26	[2, 1, 2, 3]	Process 52	[3, 1, 1, 3]		

between bidders, i.e., the product is not sold for a low price, and one must verify more properties to identify the final lowest prices. Figure 3 graphs the number of properties verified in each process model. In Process 76, for example, we have verified 29,880 properties.

In the process models on the right side of the figure, the sum of capacity points tends to be higher than on the left side. This means that more properties on process models in which the sum of capacity points is higher need to be verified, See Figure 4. Observe that the distribution of capacity points among bidders matters as well. In Process 50, for example, Bidder 1 has a capacity of 3, and the other bidders each have a capacity of 1. Since the first bidder has a high budget for the products and the other bidders have few capacity points left, this process model requires fewer properties to be verified. In total, we have verified

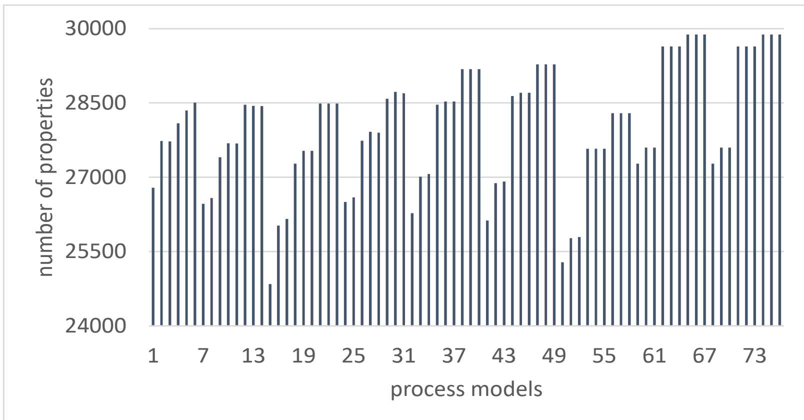

Fig. 3. Number of properties verified in each process model

2,129,637 properties. The longest verification time required to verify properties in each process model varies between 6 to 14 seconds. In particular, it makes a big difference for verification time if a bidder still has capacity left or not. In the second case, verification turns out to be false very quickly. The average time to verify properties is about 1 second in all process models. The maximum time to do so is 14 seconds and occurs with Process Models 39, 40, and 49. Each model has a standard deviation of one to two seconds for verification times. Figure 5 represents the time required to verify all properties in each process model. This number tends to grow with the total number of capacity points, as does the total verification time. Verifying each process model has required almost 6 hours and 41 minutes on average to verify all properties, i.e., 487 hours in total to finish the experiment. Across all process models, the standard deviation of verification time has been around 2 hours.

4.3 Research Questions

In this section, we answer the research questions described in Section 1.

1. Which assignments of the capacity points lead to the lowest and highest revenue? Figure 6 shows the lowest possible revenue when assigning different capacity points to the bidders. In general, the lowest revenue for higher total numbers of capacity points is relatively high. The lowest revenue is 19 and happens in Process 1, when capacity points are $[1, 1, 1, 3]$. Some of the lowest revenues, i.e.,

Fig. 4. The average number of properties per total number of capacity points

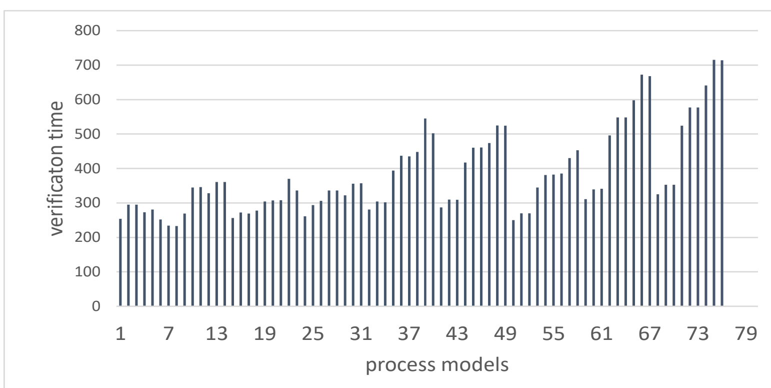

Fig. 5. Time required to verify all properties in each process model

22, 28, 33, and 34 never occur in any of the process models. The revenue of 26 only occurs once, when the capacity points are $[2, 3, 1, 1]$. The lowest revenue in Process Models 62 to 67 is higher than that of the other process models. Here it is 35. In Process Models 68 to 70, the revenue is 29, probably in line with Bidder 3 having only one capacity point. In Process Models 71 to 76 where this number is again 3, the lowest revenue becomes higher again. In the process models with the maximum lowest revenue, the capacity point of Bidder 1 is always 3, the ones of Bidder 2 and Bidder 3 vary between 2 and 3, and the one of Bidder 4 varies from 1 to 3. We see that the lowest revenue is maximal when Bidder 1 has 3 capacity points. This might be because the budget of Bidder 1 for certain products is higher than that of the other bidders, and assigning a higher capacity point to this bidder increases the revenue.

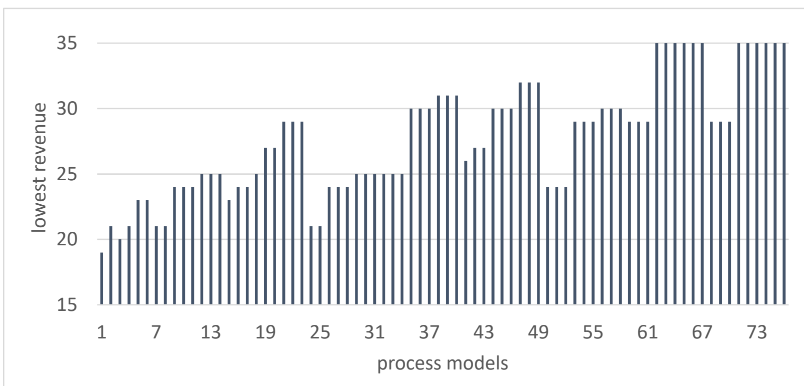

Fig. 6. The lowest possible revenue of each process model

2. With a certain allocation of the capacity points, how often can the lowest revenue occur? It is important for the auctioneer to know the lowest possible revenue before the auction. However, how often the lowest revenue can happen matters as well. Figure 7 represents the number of scenarios which lead to the lowest possible revenue. In general, when the lowest revenue increases, the number of scenarios yielding the lowest revenue increases as well. The lowest revenue in process 76 is higher than the lowest revenue of the other process models. In other words, the worst possible scenario is not too bad, as the revenue is 35. However, the number of scenarios which lead to the lowest revenue is relatively high. In total, 180 scenarios out of 720 lead to the lowest revenue. In contrast, Process 7, to give an example, has the minimal revenue of 21, but this only happens six times out of 720 scenarios.

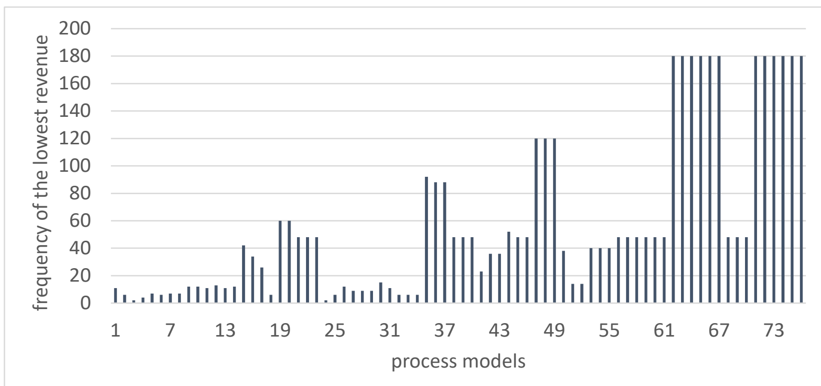

Fig. 7. The frequency of happening the lowest revenue in each process model

3. Does increasing capacity points always increase the revenue? As shown in Figure 6, the lowest revenue increases with the capacity points of bidders. This is because, when bidders have lower capacities, they cannot win a product, even though they can afford it, and a bidder with a lower budget gets the chance to win the product. But when the auctioneer increases the capacities, this becomes less frequent because the bidder with a higher budget has capacity left to win the product. However, increasing the number of capacities may lead to a monopoly. So, when the capacity points increases, the chance of earning a higher revenue is higher; at the same time, the chance of monopoly gets also higher. At this point, an auctioneer can trade off 'extent of monopolism' vs. 'expected revenue' and assign capacities accordingly.

4. What is the best assignment of capacity points to the bidders, i.e., making the worst possible revenue not too low? As explained, the lowest revenue is maximum in Processes 62 to 67 and 71 to 76. On one side, one can take an allocation of capacity points from there to avoid bad outcomes. However, the process model where bidders have fewer capacity points (Process 69) might be a better choice, since it prevents from any monopoly to some extent. On the other side, the process models in which the lowest revenue is maximal lead to this lowest revenue more often. Put differently, the number of scenarios that lead to the lowest revenue is relatively high.

5. Does changing the capacity point of a single bidder change the outcome of the auction? Another interesting observation is that increasing capacity points does not always change the revenue of the auctioneer. For example, the lowest revenue of the auction is 24 by assigning capacity points of $[3, 1, 1, 1]$. It is the same when we increase the capacity points of Bidder 4. So, when the auctioneer increases the capacity points to avoid the lowest revenues, it also matters which bidder obtains more points. Another example is that changing the capacity of only Bidder 1 by 1 point changes the lowest revenue of the auction from 23 to 25. This shows that the capacity of this bidder has a significant effect on the final prices.

5 Related Work

We first review verification techniques, to give an indication why the verification technique used here is appropriate for the use case studied. Second, we summarize studies on spectrum auctions that bear a relationship with this article.

Verification of Process Models. Many approaches exist for the verification of process models. In the following, we only mention work that is relevant for this article. [24] studies the verification of data-aware process models that allow for modification of data values. That approach only allows verifying spectrum auctions with three bidding parties and products at most. [9] uses colored Petri Nets to verify data-aware process models. The entire domain of data objects is modeled: To represent a data object with n distinct colors, they generate n new states. In consequence, there are too many states, and the verification procedure becomes computationally expensive.

Various abstraction techniques have been developed with the aim of reducing the size of the processes models and, thus, the state space [10, 16]. An abstract value is constructed by combining all the unnecessary values into one and then determining the values of the data objects necessary for verification. There is a risk that such techniques produce incorrect results. This effect may occur when elements of the process, like activities, modify the values of data objects. For example, when activities increase the price of a product during the process execution. As a result, the new value of price which might have been unnecessary before, now is relevant for verification and changes the execution of the process model. Another approach [20] abstracts from a process model and evaluates all data objects in each abstracted process fragment for three sets of rules. Each rule maintains or deletes a data object in the process model. The rules provided in [20] might change data values, and this may falsify verification results. A symbolic abstraction approach is used in [12] to support data modifications based on decision tables. The approach consists of a list of conditions and expressions for inputs and outputs. When an activity modifies the value of a large domain object, the abstraction technique featured in [12] is ineffective. In addition, they cannot provide counterexamples in case of an undesirable outcome.

Spectrum Auctions. Regarding spectrum auctions, many studies have been conducted. An auction-theoretic analysis of simultaneous ascending auctions is in [21]. A study by [11] demonstrates the limitations of theoretical analyses of simultaneous multi-round (SMR) auctions. [25] compares the design of the 3G auction in the UK and in Germany. In [3], the 3G auction is analyzed using auction theory. [5] analyses the 4G auction from a theoretical perspective. Despite the auction ending efficiently, the authors concluded that implicit collusion to achieve low prices is possible. Having said this, rationality remains to be an assumption behind all these mechanisms and frameworks.

6 Conclusions

The revenue of spectrum auctions has been an important source of income for governments. In this paper, we have identified extreme outcomes and the factors leading to them in a systematic manner. We have done this by means of existing verification techniques. In particular, we have focused on the lowest possible revenue of a spectrum auction, with different *capacity points*. We have compared the outcomes of process models with different capacity points and explained how an auction designer can take our analysis as a starting point to improve existing designs. In future work, we plan to analyze the impact of other auction parameters, such as the budget of bidders, on the revenue.

References

1. Van der Aalst, W.M.: The application of petri nets to workflow management. Journal of circuits, systems, and computers **8**(01), 21–66 (1998)
2. Ausubel, L.M., Cramton, P., Milgrom, P.: The clock-proxy auction: A practical combinatorial auction design. Handbook of Spectrum Auction Design pp. 120–140 (2006)
3. Bichler, M., Gretschko, V., Janssen, M.: Bargaining in spectrum auctions: A review of the German auction in 2015. Telecomm. Policy **41**(5–6), 325–340 (2017). https://doi.org/10.1016/j.telpol.2017.01.005
4. Brunner, C., Goeree, J.K., Holt, C.A., Ledyard, J.O.: An experimental test of flexible combinatorial spectrum auction formats. American Economic Journal: Microeconomics **2**(1), 39–57 (2010)
5. Cramton, P., Ockenfels, A.: The german 4g spectrum auction: Design and behaviour (2017)

6. Emerson, E.A.: Temporal and modal logic. In: Formal Models and Semantics, pp. 995–1072. Elsevier (1990)
7. Engelmann, D., Grimm, V.: Bidding behaviour in multi-unit auctions–an experimental investigation. The Economic Journal **119**(537), 855–882 (2009)
8. Gandhi, S., Buragohain, C., Cao, L., Zheng, H., Suri, S.: A general framework for wireless spectrum auctions. In: Proc. DySPAN. pp. 22–33 (2007). https://doi.org/10.1109/DYSPAN.2007.12
9. Ghilardi, S., Gianola, A., Montali, M., Rivkin, A.: Petri nets with parameterised data: modelling and verification. arXiv preprint arXiv:2006.06630 (2020)
10. Groefsema, H., van Beest, N., A-Cervantes, A.: Efficient conditional compliance checking of business process models. Computers in Industry **115**, 103181 (2020)
11. Gul, F., Stacchetti, E.: Walrasian equilibrium with gross substitutes. J. Econ. Theory **87**(1), 95–124 (1999). https://doi.org/10.1006/jeth.1999.2531
12. Haarmann, S., Batoulis, K., Weske, M.: Compliance checking for decision-aware process models. In: International Conference on Business Process Management. pp. 494–506. Springer (2018)
13. Hazlett, T.W., Muñoz, R.E., Avanzini, D.B.: What really matters in spectrum allocation design. Nw. J. Tech. & Intell. Prop. **10**, viii (2011)
14. Kagel, J.H., Levin, D.: Behavior in multi-unit demand auctions: Experiments with uniform price and dynamic vickrey auctions. Econometrica **69**(2), 413–454 (2001). https://doi.org/10.1111/1468-0262.00197
15. Kirchkamp, O., Reiss, J.P.: Heterogeneous bids in auctions with rational and markdown bidderstheory and experiment. Tech. rep. (2008)
16. Knuplesch, D., Ly, L.T., Rinderle-Ma, S., Pfeifer, H., Dadam, P.: On enabling data-aware compliance checking of business process models. In: International Conference on Conceptual Modeling. pp. 332–346. Springer (2010)
17. Krishna, V.: Auction Theory. Academic Press (2009)
18. Kwasnica, A.M., Sherstyuk, K.: Multiunit auctions. J. Econ. Surv. **27**(3), 461–490 (2013)
19. Lohmann, N., Verbeek, E., Dijkman, R.: Petri net transformations for business processes – a survey. In: Transactions on Petri Nets and Other Models of Concurrency II, pp. 46–63. Springer (2009). https://doi.org/10.1007/978-3-642-00899-3_3
20. Meyer, A., Weske, M.: Data support in process model abstraction. In: International Conference on Conceptual Modeling. pp. 292–306. Springer (2012)
21. Milgrom, P.: Putting auction theory to work: The simultaneous ascending auction. J. Polit. Econ. **108**(2), 245–272 (2000). https://doi.org/10.1086/262118
22. Milgrom, P., Milgrom, P.R.: Putting auction theory to work. Cambridge University Press (2004)
23. OMG: Business Process Model and Notation (BPMN), Version 2.0 (January 2011), `http://www.omg.org/spec/BPMN/2.0`
24. Ordoni, E., Mülle, J., Böhm, K.: Veryfying workflow models with data values – a case study of smr spectrum auctions. In: IEEE Conference on Business Informatics. pp. 181–190. IEEE (2020)
25. Seifert, S., Ehrhart, K.M.: Design of the 3G spectrum auctions in the UK and Germany: an experimental investigation. Ger. Econ. Rev. **6**(2), 229–248 (2005). https://doi.org/10.1111/j.1465-6485.2005.00128.x
26. Vickrey, W.: Counterspeculation, auctions, and competitive sealed tenders. J. Finance **16**(1), 8–37 (1961). https://doi.org/10.2307/2977633
27. Wolfstetter, E.: The swiss umts spectrum auction flop: Bad luck or bad design? Available at SSRN 279683 (2001)