Journal of Open Aviation Science (2024), Vol.**2**
doi:10.59490/joas.2024.8144

JOAS

EDITORIAL

Reviews and Responses for
Gradient-based smart predict-then-optimize framework for aircraft arrival scheduling problem

Authors: Go Nam Lui and Soner Demirel

Reviewers: David Gianazza, Eri Itoh, Raúl Sáez

Editor: Xavier Olive

1. Original paper

DOI for the original paper: https://doi.org/10.59490/joas.2024.7891

2. Review - round 1

2.1 Reviewer 1

The authors present a novel framework aimed at the aircraft arrival scheduling problem, with a special focus on adverse weather conditions. The methodology proposed integrates both trajectory prediction and scheduling optimization. The text is well organized and clear. However, some aspects should be clarified/improved:

The authors mention in Line 54 a "time window". What is exactly this aircraft time window and how is it computed? Are the authors referring to the attainable time window by an aircraft at the final approach fix for instance? Or is it something else?

In line with Comment 1, it is not clear for me if the authors are considering any aircraft performance model to compute a time window at the final approach fix. How are they able to know what are the arriving aircraft capabilities? i.e., how much delay could be absorbed by the aircraft, etc.

Are the authors considering the approach procedures of Gatwick airport when applying their framework? It is not clear from me from the text.

Are the authors considering wake turbulence categories to set the $s_{i,j}$ parameter? (i.e., required time separation?); I recommend the authors to use realistic separation values and rerun the framework if needed.

Is any fairness considered when assigning delays to the arriving aircraft? If not, the authors might risk assigning huge delays to one individual aircraft, which might not be operationally sound.

Most of the results the authors are presenting are related to the performance of the proposed framework. I am missing some information in the results regarding the delay assigned per aircraft, the final scheduling order, etc. I strongly recommend the authors to choose one of the instances, specify which are the arriving aircraft, what is the final order after running the framework, which is the delay assigned per aircraft, etc.

Some typos, grammar errors, format considerations:

The font size should be increased for Figure 1, Figure 2 and Figure 3 (specially legend and axis information). In addition, the authors should include the axis information for Figure 2 a and b (e.g., Latitude and Longitude (in degrees) for fIGURE 2a)

Line 73: "As stated in the research [...] real-time applications": please rewrite, not grammatically correct.

Line 111: "fir" should be replaced by "for"

Line 234:"Located [...] Central London": please rewrite, the sentence seems incomplete.

Line 251: "freeze conditions are" (instead of "is")

Table 1: the plural of "aircraft" is "aircraft" (without the "s")

Table 1: "Area of interest" (without the "s")

Line 265: Since the scope of our data "is"

Line 276: we will present (without the "s")

2.2 Reviewer 2

In this paper, the authors propose to apply a smart predict-then-optimize (SPO) method to the arrival scheduling problem (ASP), and provide results on a case-study at London Gatwick airport. Standard approaches to this problem usually first predict the landing time of each aircraft, typically using a Machine Learning method with a sum-of-squares or absolute error loss to measure the cost of the landing time prediction error, and in a second phase solve the arrival scheduling problem using the predicted landing times.
The proposed SPO framework follows the same two steps, but the "decision loss" function used to predict the landing times incorporates the resolution of the aircraft scheduling problem. From what I understand, the rationale behind the choice of a decision loss incorporating the ASP is not to improve the landing prediction times, but rather to mitigate the impact of the landing times prediction errors on the final scheduling.
As solving the ASP only makes sense when considering a number of flights requiring to land on the same runway in a given time interval, the landing time prediction problem must be solved simultaneously for this same number of flights, which is set at 15 in the paper.
The paper is clearly in the scope of JOAS. It is well structured and easy to read, overall, despite a few points that need clarification. I think it is a really good idea to explore the use of SPO in the context of aircraft arrival scheduling, that is worth publishing. There are however a few major issues that require to be clarified or addressed. This is why I recommend that the authors resubmit a revised version of the paper taking into account the suggestions listed below.

The first major issue is a potential data leakage. The traffic instances are made of 15 consecutive arriving flights, as described in Algorithm 1. A same day of traffic can provide a number of traffic instances simply by sliding a window of size 15 over the temporally ordered flights of the day. The training and test sets ratio is 8:2, as stated at the end of section 4.2. This would seem to indicate that the training and test samples were simply randomly drawn according to this ratio from the traffic instances produced by algorithm 1. Consequently, the test set most probably contains traffic instances from the same day as some instances in the training set, differing only by a few flights (at the begining and end of the sequence). Such data leakage usually results in optimistically biased results on the test set. I would recommend to split the training and test data by considering different days, or at least sequences of flights that do not overlap.

The second issue is the performance of the multi-layer perceptron (MLP) on Fig.3 (right). I find it strange that MLP is unable to fit the training data better than linear regression (LR) in the two-stage approach. Have the neural network inputs been normalized? Also the scale on Fig. 3 (left) is such that it is difficult to actually compare the training curves of LR and MLP with SPO+. The same phenomenon may occur but be hidden by the difference in scale on the left and right figures.

The third issue concerns the results on the test sets. Figure 4 compares two SPO-based method (MLP+SPO+ and LR+SPL+) with a baseline two-stage method (LR+Two-stage), showing the normalized regret values for different weather conditions. Given the scale of the normalized regret on this figure, I strongly suggest doing a statistical test (e.g. Wilcoxon pairwise test) to be more certain about the comparative performances of the different methods. Also, it would be nice to have overall results (all weather situations considered).

I have also noted a few minor points listed below: - Predicted landing time (as defined on line 145) is called "transit time" on line 201 and further.

Equation 10 (line 214): shouldn't is be z*(c) instead of z(c)?

Most figures are barely readable. Please increase figure size and/or police size

Caption of Fig. 4 is not clear ("during the training process"..."on the test sets"). I guess the training data is used to normalize the regret computed on the test sets?

2.3 Reviewer 3

This paper proposed an application of the smart predict-then-optimize (SPO) framework to the Arrival Scheduling Problem (ASP) within terminal maneuvering area. The effectiveness was shown through a case study experiments targeting arrival traffic at London Gatwick airport. The experimental results demonstrated the effectiveness of the proposing framework while adapting adverse weather conditions comparing with the traditional First-Come-First-Serve (FCFS) costs. Overall, the paper is well-written with literature reviews, scientific approaches, experimental results, and discussion. It would be beneficial to discuss the variation in delay times for each aircraft and the number of position shiftings in the discussion.

Specific comments

In Section 2, the authors refer to multiple papers collectively, for example, [6, 7, 8, 9] and [13, 14, 15, 16, 17, 18], but please explain the characteristics of each individual paper.

In Section 3, how did authors balance delay times in the entire arrival traffics? How does this methodology resolve the inequality where only certain aircraft experience significant delays while other aircraft arrive at their ideal times?

Does the proposing framework suggest the ideal arrival scheduling online? If so, in the en-route airspace before entering TMA, at what point can scheduling be proposed?

Conventionally, it is well-known that the FCFS cost is higher than the cost for optimized sequences. However, current arrival manager employs FCFS because FCFS is manageable for air traffic controllers, and it is fair to airlines. The effectiveness of the proposed method will likely require further discussion from the perspectives of ease of air traffic control and fairness to airlines.

3. Response - round 1

The authors would like to thank the reviewers for their valuable reviews and feedback. We have considered each comment carefully and summarized our responses in this document. The revised

manuscript includes highlighted updates reflecting these changes. We have completed thorough English editing throughout the document. The title has been shortened to 'Gradient-based smart predict-then-optimize framework for aircraft arrival scheduling problem' to be more concise. We have also ensured that our main content remains within the 14-page limit. Below, we respond to each reviewer's comment in detail.

3.1 Response to reviewer 1

The authors present a novel framework aimed at the aircraft arrival scheduling problem, with a special focus on adverse weather conditions. The methodology proposed integrates both trajectory prediction and scheduling optimization. The text is well organized and clear. However, some aspects should be clarified/improved:

Response

We appreciate the reviewer's positive comments on the language of our paper. We have carefully reviewed the limitations highlighted and made revisions to address the concerns, as described below.

The authors mention in Line 54 a "time window". What is exactly this aircraft time window and how is it computed? Are the authors referring to the attainable time window by an aircraft at the final approach fix for instance? Or is it something else?

In line with Comment 1, it is not clear for me if the authors are considering any aircraft performance model to compute a time window at the final approach fix. How are they able to know what are the arriving aircraft capabilities? i.e., how much delay could be absorbed by the aircraft, etc.

Response

The time window here in the literature review refers to the interval $[E_i, L_i]$, during which an aircraft i can feasibly land at the runway, where:

1. E_i (earliest time) is the minimum time achievable via speed adjustments (e.g., flying at V_{max} without violating speed limits),
2. L_i (latest time) is the maximum delay absorbable through speed reductions, path stretching, or holding patterns, bounded by fuel/operator constraints.

In our implementation, we adopt the benchmark from several significant studies [1, 2][a], where these bounds are defined as:

$$E_i = T_i - 60$$

$$L_i = T_i + 1800$$

While this benchmark simplifies aircraft-specific performance, (e.g., it does not dynamically model BADA parameters), it provides a tractable framework for scheduling algorithms. We clarify this distinction in Section 2 and 4.2 of the revised manuscript.

[a] http://data.recherche.enac.fr/ikli-alp/

Are the authors considering the approach procedures of Gatwick airport when applying their framework? It is not clear from me from the text.

Response

We thank the reviewer for this observation. While our manuscript does not explicitly state this, our study implicitly aligns with Gatwick Airport's approach procedures. The trajectory patterns and traffic flow in Figure 2 (a) (depicting the TMA) reflect Gatwick's standard arrival routes. To eliminate ambiguity, we will revise Section 4.1 to explicitly state that the framework assumes Gatwick's procedures.

Are the authors considering wake turbulence categories to set the $s_{i,j}$ parameter? (i.e., required time separation?); I recommend the authors to use realistic separation values and rerun the framework if needed.

Response

Yes, in our study, we consider the wake turbulence categories (WTC) to set the $s_{i,j}$ parameter. Based on the aircraft type code, we can match the WTC based on the Aircraft Database released by OpenSky Network. The $s_{i,j}$ is then assigned based on the WTC [a]. We have added explanations in Section 4.2 to address this comment.

[a] https://knowledgebase.vatsim-germany.org/books/separation/page/wake-turbulence-separation

Is any fairness considered when assigning delays to the arriving aircraft? If not, the authors might risk assigning huge delays to one individual aircraft, which might not be operationally sound.

Most of the results the authors are presenting are related to the performance of the proposed framework. I am missing some information in the results regarding the delay assigned per aircraft, the final scheduling order, etc. I strongly recommend the authors to choose one of the instances, specify which are the arriving aircraft, what is the final order after running the framework, which is the delay assigned per aircraft, etc.

Response

Thank you for these important comments regarding fairness and detailed results. We have conducted additional analysis to address these concerns in Section 5:

Regarding fairness in delay assignments, we have analyzed the transit time differences and position shifting patterns for our framework. Our analysis focuses on the maximum precipitation scenario, which represents the most challenging conditions with the highest total costs for MLP+SPO+. As shown in Table 2, MLP+SPO+ demonstrates improved fairness metrics compared to optimization with true cost:

- Lower mean transit time
- difference (18.60s vs. 43.62s)
- Reduced standard deviation in transit times (181.67s vs. 236.69s)
- Fewer position shifts per instance (13 vs. 17)

To provide concrete operational insights, consider that with 15 aircraft per instance, MLP+SPO+ achieves on average less than one position shift per aircraft. This suggests our framework maintains reasonable operational stability while optimizing for performance.

We acknowledge that neither MLP+SPO+ nor the baseline explicitly incorporates fairness parameters in their optimization objectives. This is reflected in the relatively high standard deviations in transit time differences (181.67s for MLP+SPO+ and 236.69s for true cost optimization). While our framework shows implicit improvements in fairness metrics, these results highlight an opportunity for future work

to integrate explicit fairness constraints into the optimization model. Such potential direction has been added to the conclusion section.

The maximum delay observed (572.10s for MLP+SPO+ vs. 703.54s for true cost optimization) indicates that while large delays can occur, our approach reduces their magnitude compared to the baseline optimization method.

Some typos, grammar errors, format considerations: The font size should be increased for Figure 1, Figure 2 and Figure 3 (specially legend and axis information). In addition, the authors should include the axis information for Figure 2 a and b (e.g., Latitude and Longitude (in degrees) for fIGURE 2a)

Response

Based on the reviewer's comment, we have adjusted the figure's font size and added corresponding axis labels to the figures.

Line 73: "As stated in the research [...] real-time applications": please rewrite, not grammatically correct.

Line 111: "fir" should be replaced by "for"

Line 234:"Located [...] Central London": please rewrite, the sentence seems incomplete.

Line 251: "freeze conditions are" (instead of "is")

Table 1: the plural of "aircraft" is "aircraft" (without the "s")

Table 1: "Area of interest" (without the "s")

Line 265: Since the scope of our data "is"

Line 276: we will present (without the "s")

Response

We appreciate the reviewer's thorough comments. All the above typos and grammar mistakes are revised as suggested. We also check the whole manuscript thoroughly to avoid grammar mistakes.

3.2 Response to reviewer 2

In this paper, the authors propose to apply a smart predict-then-optimize (SPO) method to the arrival scheduling problem (ASP), and provide results on a case-study at London Gatwick airport. Standard approaches to this problem usually first predict the landing time of each aircraft, typically using a Machine Learning method with a sum-of-squares or absolute error loss to measure the cost of the landing time prediction error, and in a second phase solve the arrival scheduling problem using the predicted landing times.
The proposed SPO framework follows the same two steps, but the "decision loss" function used to predict the landing times incorporates the resolution of the aircraft scheduling problem. From what I understand, the rationale behind the choice of a decision loss incorporating the ASP is not to improve the landing prediction times, but rather to mitigate the impact of the landing times prediction errors on the final scheduling.
As solving the ASP only makes sense when considering a number of flights requiring to land on the same runway in a given time interval, the landing time prediction problem must be solved simultaneously for this same number of flights, which is set at 15 in the paper.

The paper is clearly in the scope of JOAS. It is well structured and easy to read, overall, despite a few points that need clarification. I think it is a really good idea to explore the use of SPO in the context of aircraft arrival scheduling, that is worth publishing. There are however a few major issues that require to be clarified or addressed. This is why I recommend that the authors resubmit a revised version of the paper taking into account the suggestions listed below.

Response

We appreciate your kind words and encouragement on our paper. We have considered all the comments and address each of them accordingly.

The first major issue is a potential data leakage. The traffic instances are made of 15 consecutive arriving flights, as described in Algorithm 1. A same day of traffic can provide a number of traffic instances simply by sliding a window of size 15 over the temporally ordered flights of the day. The training and test sets ratio is 8:2, as stated at the end of section 4.2. This would seem to indicate that the training and test samples were simply randomly drawn according to this ratio from the traffic instances produced by algorithm 1. Consequently, the test set most probably contains traffic instances from the same day as some instances in the training set, differing only by a few flights (at the begining and end of the sequence). Such data leakage usually results in optimistically biased results on the test set. I would recommend to split the training and test data by considering different days, or at least sequences of flights that do not overlap.

Response

Thank you for raising this important concern about potential data leakage. We acknowledge that our original explanation on the instance generation is raising this concern with the wrong description of the method. Our traffic instances are constructed using non-overlapping 45-minute time windows (not sliding windows) for each day. This ensures that no flight sequence overlaps with another instance within the same day. For example, when a valid instance is found (15 flights within 45 minutes):

1. The window advances by 15 flights (flights per instance), ensuring the next instance starts after the previous one.
2. If flights 1–15 form a valid instance, the next candidate group starts at flight 16.

When a group fails the time constraint:

1. The window slides by 1 flight, but valid instances are still non-overlapping.
2. If flights 1–15 are invalid, flights 2–16 are checked. If flights 2–16 are valid, the next group starts at flight 17, not 2.

This guarantees that no two valid instances share flights, even if some groups are skipped. To address the reviewer's concern, we updated the current methodology description in section 3.2 to better match our code.

The second issue is the performance of the multi-layer perceptron (MLP) on Fig.3 (right). I find it strange that MLP is unable to fit the training data better than linear regression (LR) in the two-stage approach. Have the neural network inputs been normalized? Also the scale on Fig. 3 (left) is such that it is difficult to actually compare the training curves of LR and MLP with SPO+. The same phenomenon may occur but be hidden by the difference in scale on the left and right figures.

Response

Thank you for the insightful comments on Figure 3. We agree that the underperformance of MLP-2S compared to linear regression (LR-2S) is counterintuitive and have investigated this further.

1. **Input Normalization:** We re-ran experiments with normalized input features for both MLP and LR. While normalization slightly improved MLP-2S performance (Figure 1), LR-2S and LR-SPO+ (Figure 3, 4) failed to converge under the same conditions. To ensure a fair comparison between MLP and LR across both SPO+ and two-stage (2S) approaches, we retained the original (unscaled) inputs. This choice prioritizes methodological consistency over isolated performance gains, as LR is highly sensitive to input scaling. In addition, we can observe that LR+SPO+ has a converging regret curves while LR+2S doesn't, showing the superior performance of SPO+ on reducing the decision loss during training.
2. **Figure Scale Adjustment:** We acknowledge the difficulty in comparing training curves due to differing scales in Figure 3 (left vs. right). In the revised figure, we now use a normalized loss for both 2S curve and SPO+ curve for easier comparison.
3. **Normalization as Future Work:** While normalization improves MLP-2S, its interaction with LR requires deeper study (e.g., adjusting learning rates or regularization). We will explicitly discuss this in the paper's limitations and highlight input normalization as a promising improvement for future work.

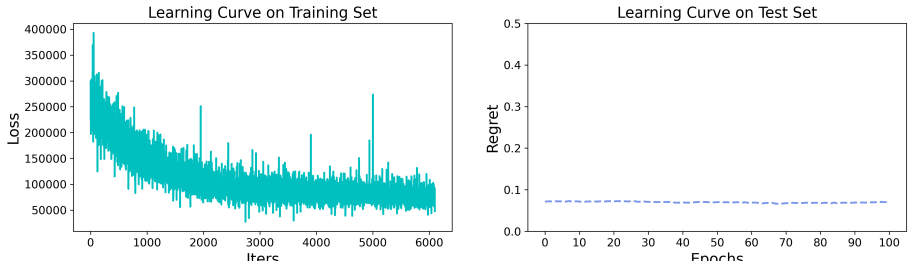

Figure 1. MLP+2S, 100 epochs, min time window

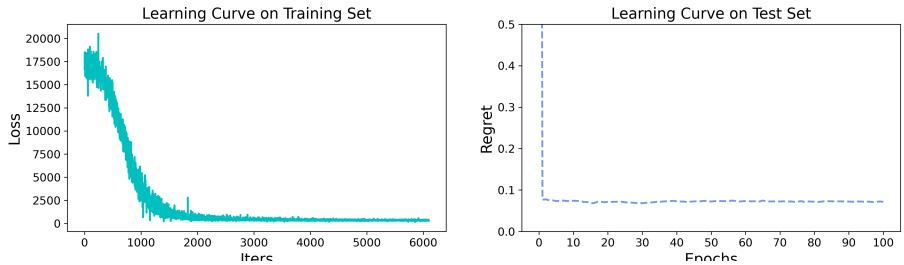

Figure 2. MLP+SPO+, 100 epochs, min time window, normalized inputs.

The third issue concerns the results on the test sets. Figure 4 compares two SPO-based method (MLP+SPO+ and LR+SPL+) with a baseline two-stage method (LR+Two-stage), showing the normalized regret values for different weather conditions. Given the scale of the normalized regret on this figure, I strongly suggest doing a statistical test (e.g. Wilcoxon pairwise test) to be more certain about the comparative performances of the different methods. Also, it would be nice to have overall results (all weather situations considered).

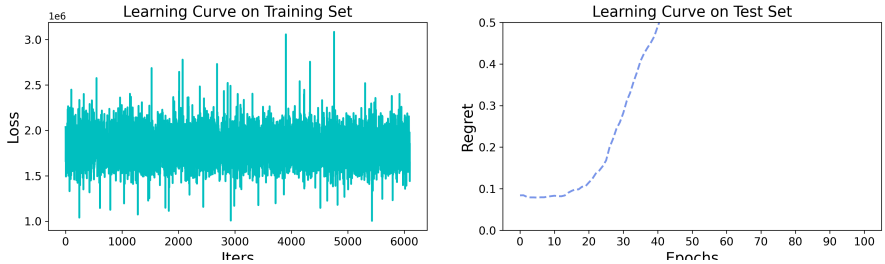

Figure 3. LR+2S, 100 epochs, min time window, normalized inputs.

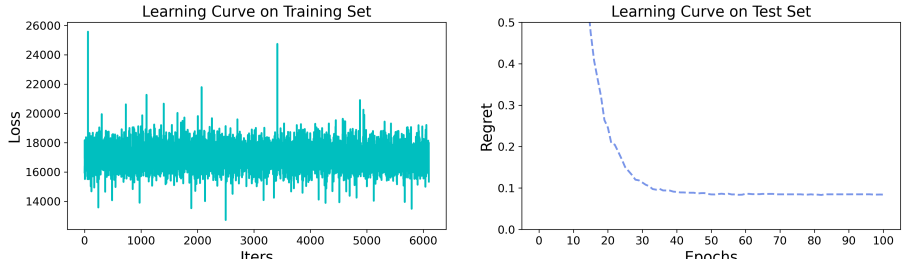

Figure 4. LR+SPO+, 100 epochs, min time window, normalized inputs.

Response

We thank the reviewer for the feedback. Below are our responses:

1. **Statistical Testing:** Following the reviewer's suggestion, we performed the Mann-Whitney U test (a non-parametric alternative to the Wilcoxon test for unpaired samples) to rigorously compare the normalized regret distributions between methods. These results are now integrated into the revised discussion in Section 5. For example, in the maximum wind scenario, MLP+SPO+ significantly outperforms LR+Two-Stage ($U = 130.0, p = 0.024$), supporting our claim about the benefits of end-to-end learning with expressive architectures.
2. **All Weather Situations:** The reviewer's request for "overall results" (all weather events occurring simultaneously) is insightful. However, due to the temporal scope of our dataset (June–September 2023), we observed no instances where all weather events (e.g., wind, precipitation, dangerous phenomenon, visibility) occurred concurrently.

I have also noted a few minor points listed below: - Predicted landing time (as defined on line 145) is called "transit time" on line 201 and further.

Response

The term on Line 145 is revised as suggested.

Equation 10 (line 214): shouldn't is be z*(c) instead of z(c)?

Response

Yes, this is a typo. We revise this part as suggested.

Most figures are barely readable. Please increase figure size and/or police size

Response

Based on the reviewer 1 and reviewer 2's comments, we have increased the figures' size as well as the font size in the figures.

Caption of Fig. 4 is not clear ("during the training process"..."on the test sets"). I guess the training data is used to normalize the regret computed on the test sets?

Response

We agree with you that this caption is confusing, we have updated it to better introduce the figure.

3.3 Response to reviewer 3

This paper proposed an application of the smart predict-then-optimize (SPO) framework to the Arrival Scheduling Problem (ASP) within terminal maneuvering area. The effectiveness was shown through a case study experiments targeting arrival traffic at London Gatwick airport. The experimental results demonstrated the effectiveness of the proposing framework while adapting adverse weather conditions comparing with the traditional First-Come-First-Serve (FCFS) costs. Overall, the paper is well-written with literature reviews, scientific approaches, experimental results, and discussion. It would be beneficial to discuss the variation in delay times for each aircraft and the number of position shiftings in the discussion.

Response

Thank you for your thoughtful review and positive feedback on our paper's structure, methodology, and experimental results. We appreciate your suggestion regarding the analysis of delay variations and position shifting patterns.

To address this, we have expanded our discussion to include a detailed analysis of delay distributions and sequencing changes in Section 5. As shown in Table 2, our analysis of the maximum precipitation scenario reveals that MLP+SPO+ achieves:

- A lower mean transit time difference of 18.60s compared to 43.62s for optimization with true cost
- Improved consistency in delay distribution with a standard deviation of 181.67s versus 236.69s
- More stable sequencing with an average of 13 position shifts per instance versus 17 for the baseline

With 15 aircraft per instance, this translates to less than one position shift per aircraft on average, suggesting our framework maintains operational stability while improving overall performance. While these results show implicit improvements in fairness metrics, we acknowledge that explicit fairness constraints could further enhance the framework's practical applicability.

This additional analysis strengthens our findings by demonstrating that MLP+SPO+ not only improves overall system performance but also leads to more balanced delay distributions and fewer sequence adjustments compared to traditional approaches.

Specific comments

In Section 2, the authors refer to multiple papers collectively, for example, [6, 7, 8, 9] and [13, 14, 15, 16, 17, 18], but please explain the characteristics of each individual paper.

Response

We thank the reviewer's comment. Based on our previous literature review, we elaborate the characteristics for each individual paper in Section 2.

In Section 3, how did authors balance delay times in the entire arrival traffics? How does this methodology resolve the inequality where only certain aircraft experience significant delays while other aircraft arrive at their ideal times?

Response

Thank you for the follow-up question on the inequality. As mentioned in our first response, while our current methodology does not explicitly incorporate fairness parameters, our analysis in Table 2 shows that MLP+SPO+ achieves better implicit fairness compared to the baseline, with lower mean transit time differences and reduced delay variability. However, we acknowledge this limitation and have added a discussion in our Conclusion section about potential improvements through explicit fairness constraints in future work.

Does the proposing framework suggest the ideal arrival scheduling online? If so, in the en-route airspace before entering TMA, at what point can scheduling be proposed?

Response

The proposed framework supports dynamic arrival scheduling online, but the optimal point for initiating scheduling in en-route airspace depends on predictability of transit times and the operational range defined for the system.

In our experiments, we used a 50NM radius around the airport as the scheduling boundary. This range balances two factors:

1. Predictability: Closer to the Terminal Maneuvering Area (TMA), trajectory uncertainties (e.g., speed adjustments, weather, vectoring) are reduced, enabling more reliable transit time estimates.
2. Flexibility: Scheduling too late (e.g., within 20NM) risks insufficient time for conflict resolution or speed adjustments.

We have also tested 80NM and it can also work well under this framework. For en-route scheduling beyond the 80NM, further investigation is needed to determine the maximum viable range where predictions remain accurate enough for scheduling. This could involve machine learning for trajectory forecasting or hybrid strategies that refine schedules incrementally as aircraft approach the TMA. Such study could be an interesting potential topic to investigate.

Conventionally, it is well-known that the FCFS cost is higher than the cost for optimized sequences. However, current arrival manager employs FCFS because FCFS is manageable for air traffic controllers, and it is fair to airlines. The effectiveness of the proposed method will likely require further discussion from the perspectives of ease of air traffic control and fairness to airlines.

Response

Thank you for the insightful suggestion. We have updated our Conclusion based on your comment.

4. Review - round 2

4.1 Reviewer

In my first review of the paper, I had raised three major issues that required to be addressed, in my view: 1) potentially overlapping traffic instances, possibly inducing some data leakage between the training and test set, 2) poor performance of the multi-layer perceptron (MLP), when compared with linear regression (LR), in the two-stage approach, making me wonder if the inputs had been normalized, and 3) the need of statistical tests to support the results, and a suggestion to present overall results.

The revised paper has improved in many ways. Concerning issue 1), the text describing the algorithm producing traffic instances now makes it clear that there is no data leakage. There is however a typo line 211 (Delta T should be less than or equal to Delta T max).

Concerning issue 3) the authors have performed a Mann-Whitney U test to compare the different methods, showing that their SPO+ framework improves the results with statistical significance in the maximum wind scenario, and possibly also (with no statistical significance) in the maximum dangerous phenomenon scenario. For all other scenarios there is no significant improvement, as honestly reported by the authors. I had suggested to present overall results, considering the union of all data (all weather conditions confounded). The authors have wrongfully interpreted my suggestion as considering traffic instances exhibiting all weather conditions simultaneously, and there is none. I would have been interested in the result of the statistical test on the union of all data subsets.

Concerning issue 2) the authors' answer makes it clear that the inputs to the MLP had not been normalized in the initial paper. The MLP inputs are still not normalized in the revised version. The authors chose to use un-normalized inputs for all methods, on the grounds that a) according to the authors, the same inputs (either normalized, or not) should be used for all methods to allow for fair comparison b) on the few trials they made, LR fails to converge with normalized inputs, in both combinations LR-2S (the baseline two-stage aproach) and LR-SPO+

In my opinion, this answer poses a number of issues. Normalizing the inputs of a neural network is necessary to make the gradient descent work correctly. With a sigmoid or tanh activation function, this avoids to saturate the activations on a plateau of the activation function (with gradient 0). With a ReLu activation function, the objective is to stay around the non-linear part of the activation function. Let us imagine that we use raw inputs with a range of values such that the activations are all constant (leftmost part of ReLu). The gradient of the output error is then 0, and we can do any number of gradient descent iterations without learning anything. If the inputs are such that the activations are far away on the rightmost part of ReLu, we might have to do a huge number of iterations before being able to cancel some of the activations and be able to learn something from the data. So the MLP inputs should absolutely be normalized.

On the contrary, linear regression with ordinary least squares is in theory insensitive to input normalization. It should give exactly the same results with or without input normalization. The fact that the two-stage method LR-2S fails to converge seems related to the fact that a gradient descent was used to learn the linear model used in the first stage of the method, probably with a bad choice of learning parameters. Why not use an ordinary least squares (OLS) method instead of gradient descent for the linear regression in LR-2S where the linear regression is not coupled with the second stage? The OLS method solves an over-determined linear system and would give the exact values of the linear model weights. This would provide a reliable baseline approach, without having to tune any learning parameter.

My idea of a fair comparison of the methods is that each method should be tuned to its best before

comparing the results on the test set. I would suggest to use un-normalized inputs (or normalized, indifferently) and ordinary least squares regression for the baseline LR-2S, and normalized inputs for the other methods using gradient descent. More extensive parameter tuning should also be performed for these other methods, in order to make each of them work as best as possible.

In light of the above comments, and considering that this is already the second review of this paper, I recommend to reject the paper in its current form. I am well aware of the amount of work done and all the efforts already put by the authors in this publication, and I still think that this work might be worth publishing. I encourage the authors to take the above suggestions into account, re-run their experiments, and resubmit a fresh version of the paper if the results are conclusive.

If they do so, I have another comment to further improve their publication, concerning Fig. 5 and its discussion (lines 349 to 352). It is not clear to me how could MLP-SPO+ results could be better than the optimal theoretical cost, which is supposed to represent the "theoretical optimal performance under ideal conditions". If this "optimal true cost" is actually the theoretical minimum of the cost function, no method should be able to find strictly lower values than this minimum (otherwise it is not a minimum). Or is is that the methods do not minimize exactly the same cost? Some explanation is needed here.

5. Response - round 2

5.1 Response to reviewer

In my first review of the paper, I had raised three major issues that required to be addressed, in my view: 1) potentially overlapping traffic instances, possibly inducing some data leakage between the training and test set, 2) poor performance of the multi-layer perceptron (MLP), when compared with linear regression (LR), in the two-stage approach, making me wonder if the inputs had been normalized, and 3) the need of statistical tests to support the results, and a suggestion to present overall results.

The revised paper has improved in many ways. Concerning issue 1), the text describing the algorithm producing traffic instances now makes it clear that there is no data leakage. There is however a typo line 211 (Delta T should be less than or equal to Delta T max).

Concerning issue 3) the authors have performed a Mann-Whitney U test to compare the different methods, showing that their SPO+ framework improves the results with statistical significance in the maximum wind scenario, and possibly also (with no statistical significance) in the maximum dangerous phenomenon scenario. For all other scenarios there is no significant improvement, as honestly reported by the authors. I had suggested to present overall results, considering the union of all data (all weather conditions confounded). The authors have wrongfully interpreted my suggestion as considering traffic instances exhibiting all weather conditions simultaneously, and there is none. I would have been interested in the result of the statistical test on the union of all data subsets.

Response

We sincerely thank Reviewer 2 for their continued engagement with our manuscript and for the detailed feedback provided in this second round. We appreciate the acknowledgment that the revised paper has improved, particularly regarding the clarification of data leakage (Issue 1) and the inclusion of statistical tests (Issue 3). We are grateful for the identification of the typo on line 211 and corrected this in the final version. Regarding the suggestion to present statistical test results on the union of all data, we thank the reviewer for this clarification. We initially misinterpreted the suggestion as looking for instances exhibiting all weather conditions simultaneously, which indeed don't exist in our dataset. To address

the reviewer's concern, we further conduct the statistical test on a union dataset and the result supports our conclusion from the last version.

Concerning issue 2) the authors' answer makes it clear that the inputs to the MLP had not been normalized in the initial paper. The MLP inputs are still not normalized in the revised version. The authors chose to use un-normalized inputs for all methods, on the grounds that a) according to the authors, the same inputs (either normalized, or not) should be used for all methods to allow for fair comparison b) on the few trials they made, LR fails to converge with normalized inputs, in both combinations LR-2S (the baseline two-stage aproach) and LR-SPO+

In my opinion, this answer poses a number of issues. Normalizing the inputs of a neural network is necessary to make the gradient descent work correctly. With a sigmoid or tanh activation function, this avoids to saturate the activations on a plateau of the activation function (with gradient 0). With a ReLu activation function, the objective is to stay around the non-linear part of the activation function. Let us imagine that we use raw inputs with a range of values such that the activations are all constant (leftmost part of ReLu). The gradient of the output error is then 0, and we can do any number of gradient descent iterations without learning anything. If the inputs are such that the activations are far away on the rightmost part of ReLu, we might have to do a huge number of iterations before being able to cancel some of the activations and be able to learn something from the data. So the MLP inputs should absolutely be normalized.

On the contrary, linear regression with ordinary least squares is in theory insensitive to input normalization. It should give exactly the same results with or without input normalization. The fact that the two-stage method LR-2S fails to converge seems related to the fact that a gradient descent was used to learn the linear model used in the first stage of the method, probably with a bad choice of learning parameters. Why not use an ordinary least squares (OLS) method instead of gradient descent for the linear regression in LR-2S where the linear regression is not coupled with the second stage? The OLS method solves an over-determined linear system and would give the exact values of the linear model weights. This would provide a reliable baseline approach, without having to tune any learning parameter.

My idea of a fair comparison of the methods is that each method should be tuned to its best before comparing the results on the test set. I would suggest to use un-normalized inputs (or normalized, indifferently) and ordinary least squares regression for the baseline LR-2S, and normalized inputs for the other methods using gradient descent. More extensive parameter tuning should also be performed for these other methods, in order to make each of them work as best as possible.

In light of the above comments, and considering that this is already the second review of this paper, I recommend to reject the paper in its current form. I am well aware of the amount of work done and all the efforts already put by the authors in this publication, and I still think that this work might be worth publishing. I encourage the authors to take the above suggestions into account, re-run their experiments, and resubmit a fresh version of the paper if the results are conclusive.

Response

We understand the reviewer's concerns regarding the use of unnormalized inputs, particularly for the Multi-Layer Perceptron (MLP), and the choice of gradient descent (GD) for the Linear Regression (LR) baseline instead of Ordinary Least Squares (OLS). We would like to elaborate on the rationale behind our methodological decisions:

Rationale for Consistent Input Handling (Unnormalized):

Our study's primary objective was to investigate and compare the effectiveness of the SPO+ loss function against a standard two-stage prediction loss, within different predictive architectures (MLP and LR). To isolate the impact of the loss function itself, we made a deliberate methodological choice to maintain consistency in the input data processing across the methods being directly compared (i.e., MLP+SPO+ vs. MLP+2S, and LR+SPO+ vs. LR+2S). Using normalized inputs for one method (e.g., MLP) and unnormalized for another (e.g., LR), or vice-versa, would introduce another confounding variable, making it harder to attribute performance differences solely to the loss function.

Empirical Observation with LR: As noted in our previous response and acknowledged by the reviewer, our preliminary trials using normalized inputs with our GD implementation for LR models (both LR+2S and LR+SPO+) led to convergence issues under the hyperparameter settings explored. While we acknowledge that further extensive tuning might resolve this for LR+GD, this empirical finding reinforced our decision to proceed with unnormalized inputs for all methods to ensure all reported methods converged and could be fairly compared under the conditions of our study.

MLP Performance: We acknowledge the reviewer's valid points about the benefits of normalization for MLP training with GD. Despite using unnormalized inputs, our MLP architecture still demonstrated effective learning capabilities, and crucially, the MLP+SPO+ approach showed statistically significant improvement over MLP+2S in the maximum wind scenario ($U = 130.0$, $p = 0.024$) and suggested improvement in the maximum dangerous phenomenon scenario ($U = 147.0$, $p = 0.066$). This indicates that while normalization might have improved absolute MLP performance, its absence did not prevent us from demonstrating the relative benefits of the SPO+ approach.

Rationale for Optimization Algorithm Choice (GD vs. OLS):

We agree with the reviewer that OLS is the standard, exact, and tuning-free method for optimizing a standalone least-squares linear regression model, as typically used in the first stage of an LR+2S baseline. However, our adoption of the SPO+ framework fundamentally alters the optimization landscape for the SPO-based methods.

Necessity of GD for SPO+: The core SPO+ framework, whether used with LR or MLP, relies on minimizing the non-standard SPO+ loss. As established in the foundational work by Elmachtoub & Grigas [3] (Section 5), optimizing models under the SPO+ loss generally requires iterative, gradient-based methods, as there is no closed-form solution akin to OLS. They explicitly discuss and utilize (sub)gradient-based approaches (like SGD, see their Appendix C) for linear predictors ($f(x) = Bx$) within the SPO+ framework. Therefore, our use of GD for LR+SPO+ is not an arbitrary choice but a direct implementation approach consistent with the original SPO methodology.

Consistency in LR Comparison: Since LR+SPO+ necessitates a gradient-based approach, we chose to also use GD for the baseline LR+2S method within our study. This decision was made to maintain consistency in the optimization algorithm strategy when directly comparing LR+SPO+ and LR+2S. This approach allows us to better isolate the impact of the SPO+ loss function versus the standard two-stage loss within the linear regression architecture, avoiding a comparison where methods differ simultaneously in both the loss function and the fundamental optimization technique (iterative GD vs. closed-form OLS). In addition, our approach aligns with recent developments in end-to-end predict-then-optimize frameworks. Tang & Khalil [4] follow a similar workflow in their implementation of the PyEPO library (detailed implementation is in Table 3), which has become a standard tool for this research area.

On "Fair Comparison":

We recognize the reviewer's perspective that a "fair comparison" could involve tuning each method to its individual optimum (e.g., normalized inputs and tuned GD for MLP/LR+SPO+, potentially OLS for LR+2S). This is a valid approach for comparing the absolute best achievable performance of different methods.

However, our research objective was to isolate the impact of the novel loss function while maintaining

consistency in other aspects (input processing, optimization strategy where possible). This approach provides valuable insights into the specific benefits conferred by the SPO+ loss itself, holding other factors constant. As mentioned above, the work of Tang & Khalil [4] perform similar comparison in their implementation of the PyEPO library, a standard tool for this research area.

We concede that exploring optimally tuned versions of each method represents an important avenue for future work. We propose to add a paragraph in the Discussion/Limitations section acknowledging this point, clarifying our rationale for consistency, and highlighting that the reported results compare the methods under the specific (consistent) conditions chosen, rather than comparing their individually optimized peaks.

Conclusion on Issue 2:

While we deeply respect the reviewer's expertise and perspective on best practices for MLP normalization and LR baseline implementation, we believe our chosen methodology, prioritizing consistency to isolate the effect of the SPO+ loss function, is valid for the specific research questions addressed in this paper. Our results demonstrate the value of the SPO+ approach even under these consistent, though perhaps not individually optimal, conditions. We believe that re-running all experiments with individually optimized methods would constitute a substantially different study. Instead, we propose to enhance the manuscript by explicitly discussing the rationale for our choices and acknowledging the limitations and alternative approaches in the last section.

If they do so, I have another comment to further improve their publication, concerning Fig. 5 and its discussion (lines 349 to 352). It is not clear to me how could MLP-SPO+ results could be better than the optimal theoretical cost, which is supposed to represent the "theoretical optimal performance under ideal conditions". If this "optimal true cost" is actually the theoretical minimum of the cost function, no method should be able to find strictly lower values than this minimum (otherwise it is not a minimum). Or is is that the methods do not minimize exactly the same cost? Some explanation is needed here.

Response

We thank the reviewer for pointing out the potential confusion regarding Fig. 5 and the "optimal theoretical cost." We agree this requires clarification. The "optimal true cost" of 2412.4 represents the minimum achievable average cost on the test set when optimizing schedules using the true landing times as inputs, rather than predictions. However, it's important to understand that this doesn't necessarily represent a theoretical lower bound for all possible scheduling approaches.

The SPO+ method's apparent outperformance (achieving 1364.3) occurs because it optimizes for the end-to-end task rather than simply trying to optimize the sequence with true landing times. By learning the relationship between input features and final costs directly, SPO+ can discover scheduling patterns that work better in practice than those derived from optimizing with true landing times. This is possible because the true landing time optimization doesn't account for complex operational dynamics and uncertainty that the learned models implicitly handle.

We have revised the text accompanying Figure 5 (Lines 349-352) to clarify precisely what the "optimal true cost" represents and explain why SPO+ can sometimes achieve better performance. The comparison showcases the advantage of end-to-end learning approaches like SPO+ over traditional predict-then-optimize methods in complex operational environments with uncertainty.

References

[1] Rakesh Prakash, Rajesh Piplani, and Jitamitra Desai. "An optimal data-splitting algorithm for aircraft scheduling on a single runway to maximize throughput". In: *Transportation Research Part C: Emerging Technologies* 95 (2018), pp. 570–581.

[2] Sana Ikli, Catherine Mancel, Marcel Mongeau, Xavier Olive, and Emmanuel Rachelson. "The aircraft runway scheduling problem: A survey". In: *Computers & Operations Research* 132 (2021), p. 105336.

[3] Adam N Elmachtoub and Paul Grigas. "Smart "predict, then optimize"". In: *Management Science* 68.1 (2022), pp. 9–26.

[4] Bo Tang and Elias B Khalil. "PyEPO: a PyTorch-based end-to-end predict-then-optimize library for linear and integer programming". In: *Mathematical Programming Computation* (July 2024). ISSN: 1867-2957. DOI: 10.1007/s12532-024-00255-x.