Response to Reviewer 1

"...Provide some evidence that the parameterizations relate to structural properties of failure graphs arising from real applications"

There are many application scenarios where our parameterizations are reasonable. For instance, if the failure graph depends on the timeout given by the user, then the user has some control over its density, and therefore can tune the parameters. However, if this data is fine-tuned by the user as part of an experimental evaluation, then this might be subject to the criticism of having fine-tuned the data to reach a positive outcome. Therefore, the data source should be independent from the user, and we are currently reviewing some external data obtained from the QBF setting (where we have no control over the failure rates) for inclusion in the final or journal version of the paper.

"Analyze the complexity of these problems in the more common scenarios 1, 2, and 3. Do the problems remain NP-hard (without parameterization)? What about parameterized with respect to length of the optimal scheduling?"

It is not difficult to show that the problem remains NP-hard in all 3 scenarios via a reduction from Min-Sum Set Cover (reference [6] in the paper). As for the parameterized complexity of the problem parameterized by the length of an optimal schedule, it remains W[2]-hard in all 3 scenarios via a reduction from Set Cover; this reduction is the same as the one used for the more general problem. We will add a specific remark about the above into the final version.

Response to Reviewer 2

"the paper said that it would use a spade to mark statements with omitted proofs, but they were not actually marked with spades."

We apologize for this confusion. We will fix this in the final version.

"It might be helpful to add a chart to the introduction, indicating the map from parameter to tractable/intractable"

Thank you for this suggestion. We will add the suggested chart to the introduction.

"As stated earlier in the review, this submission could be a strong accept if it gave some empirical evidence that its algorithms help with the original questions. Are these parameters actually small?"

We believe that our main contribution is an in-depth complexity classification of the problems under consideration. As for the size of the parameters in practice, please see our response to Reviewer 1.

Response to Reviewer 3

We would like to profoundly thank the reviewer for spotting the issue concerning the running times in Theorem 7, and we apologize for missing this in our proofreading. Fortunately, it is easy to fix the issue by slightly modifying the records kept in the dynamic programming table: while these currently store the exact positions of "important algorithms" w.r.t. the whole schedule of length ℓ (which is where the dependency on ℓ came from), it suffices to store their relative positions (which can be kept as a permutation of only these algorithms).

More precisely, to fix this issue, the dynamic programming table in Lemma 11 still stores records with the same structure $(\alpha_{\text{past}}, \alpha_{\text{future}}, \sigma, \delta)$ as before, with the difference being that now σ captures only the relative positions of the following algorithms (as opposed to their global positions):

1. the algorithms identified by α_{past} and α_{future},
2. the algorithms in the current bag, and
3. the algorithms that will solve the tests in the bag.

δ then only tells us whether the test will eventually be solved by an algorithm in the future or whether it is solved by an already-introduced algorithm (this is needed for processing the *join nodes*). Since in the proofs of Claims 13-18 (in the full version) we only need the relative positions of the algorithms in the bag in order to update the records and their associated costs, all the arguments in these claims seamlessly go through. The outcome is a fixed-parameter algorithm for this case with a runtime of $\mathcal{O}^*(4^{\text{tw}} \cdot \text{tw}^{\text{tw}})$.

We have already prepared a formal write-up of this adapted proof and will make the necessary updates to the final version. (The NeurIPS guidelines explicitly forbid us from linking to a modified version of the paper.)

"In proof of Proposition 1, t is used to denote test and destination of the DAG...".

We will use a different notation for the destination vertex.