

Problem C: Microarchitecture Design Space Exploration

(DAMO Academy)

Q&A

Q1. Regarding the problem C, I have the following concept.

I want to build an offline dataset with objective functions like power, performance, area, congestion score with early physical estimation. I have encountered some issues with building the entire VLSI flow(Chipyard, Hammer and ASAP7, etc). Since the contest also builds the entire flow and collects the data for the contest. Could you please help to give some suggestions? Thank you very much.

A1. We will provide this dataset later soon. The data collection process is time-consuming and resource-consuming. Please consider using the provided dataset for this contest.

Q2. I have a question about Problem C since I noted that in the instruction document, some example optimizers are referenced. However, I could only find the source code of the random optimizer in the instruction doc, and there is no related code in the provided package installed via pip. May I know whether it is possible to provide the code of the other example optimizers?

Thank you.

A2. Please refer to the released case on 5/31.

Q3. Could you please help to advise if the data return to us for each test is a single ADRS value or PPA values? Thank you.

A3. Yes, the result after running a test is a ADRS value, not PPA.

Q4. We have two additional questions about problem C. Could you please help to check? Thank you.

1) In the dataset to be released in the future, for a specific design, are all the feasible configurations in the corresponding design space (along with PPA & times) provided in the dataset? Otherwise, it could happen that we may query a configuration that is feasible but not included in the dataset.

2) Is there a way to perform “early stopping” using the given codebase? Since we notice that the number of queries is provided as an constant argument when launching the script, but it could be possible that an optimization strategy wants to determine this number dynamically and early stop if needed.

A4. Below please find our responds.

1) All configurations are provided in the dataset, and you should make sure your inquiry is valid, i.e., a valid configuration within the design space.

2) You can configure the number of queries provided, i.e., you can specify the number of queries in your optimizer so that you can terminate an optimization strategy w.r.t. your "early stopping" criterion.

You can specify the number of queries through '-q' or '--num-of-queries' option when you launch your optimizer.

Q5. Could you please help to check the following questions? Thank you very much!

1). Is there a proportional or inverse proportional independent relationship between ADRS and the normalized value of P, P, and A in this problem, respectively?

2). When we try to check the relation between ADRS and the normalized PPA, we find a seemingly unexpected PPA appearing in the reference pareto frontier after filtering by get_non_dominated() (please refer to the attachment). Is it proper that these two PPAs appear simultaneously in the same pareto frontier? Will it affect ADRS if the predicted pareto frontier is worse than the reference pareto frontier?

```
tensor([[ -1.1928,  1.4688,  2.3476],
        [  1.2082,  3.4290,  1.4483],
        [  1.2082,  1.1483,  1.7981],
        [  0.3969,  3.4113,  2.2354],
        [  0.0578,  3.4290,  1.7550],
        [  0.4402, -0.3200,  2.0225],
        [ -1.2399,  3.4290,  2.3476],
        [ -1.3220,  3.4290,  2.4186],
        [ -0.5948, -0.3200,  2.2767],
        [  1.2082, -0.3200,  1.8691],
        [  0.4871,  3.4290,  1.6660],
        [ -0.5513,  3.4290,  2.1645],
        [  0.5697,  3.4290,  1.4952],
        [  1.2082,  0.7579,  1.8691],
        [  0.0487,  3.4290,  1.9870],
        [ -0.7382, -0.3200,  2.3476],
        [ -1.0906, -0.3200,  2.4186]])
```

3). Can we import get_adrs from iccad_contest.functions.problem and directly call function get_adrs() in the implement of function observe()?

A5. Below please find our responds.

1) A higher normalized performance value, a lower normalized power value, and a lower normalized area value suggest that the design is on the true Pareto frontier with a higher possibility. Suppose your optimizer can explore designs with a higher normalized performance value, lower normalized power value, and lower normalized area value as much as possible. In that case, the ADRS of your optimizer will be lower, indicating better your optimizer is. Simultaneously, it would be best to consider the overall running time of your optimizer. A lower overall running time of your optimizer is expected.

2) It is normal that these two PPA values are on the true Pareto frontier.

Firstly, you can inspect their PPA values by printing them using higher precision, e.g., 8 significant digits. Secondly, the designs corresponding to these two PPA values indicate approximately the same performance. However, one design dissipates more power and has a smaller area, while the other one dissipates less power but has a larger area. Both of them cannot dominate each other.

3) You can leverage all functions provided by the contest platform.

Q6. We have a question about problem C and would like a clarification.

Can we pretrain a model locally using provided or self-generated data, and submit the obtained model parameter files together with other code files to the online evaluation system for further operations (such as fine-tuning for the specific dataset)? Or is this not allowed and all training that might be involved has to be done online from scratch? We think the latter seems more reasonable, otherwise it would be difficult to maintain a fair evaluation of ORT.

Thank you.

A6. Yes, you are right. It's not allowed to submit the obtained model parameter files to the online evaluation system. Any model leveraged in your optimizer should be trained online, and the training time is also corporated into the ORT automatically by the contest platform.

The submission includes:

- 1. your implemented Python scripts,**
- 2. the execution commands,**
- 3. related JSON configurations leveraged by your algorithm, if any, and**
- 4. dependency packages list for supporting your algorithm execution, if any.**

Q7. We find a PPA with higher performance, lower power, and smaller area and another one with lower performance, higher power, and larger area appear simultaneously in the reference pareto frontier.

According to the previous clarification, the former shall dominate the latter, right?

```
tensor([[ -1.1928,  1.4688,  2.3476],
        [  1.2082,  3.4290,  1.4483],
        [  1.2082,  1.1483,  1.7981],
        [  0.3969,  3.4113,  2.2354],
        [  0.0578,  3.4290,  1.7550],
        [  0.4402, -0.3200,  2.0225],
        [ -1.2399,  3.4290,  2.3476],
        [ -1.3220,  3.4290,  2.4186],
        [ -0.5948, -0.3200,  2.2767],
        [  1.2082, -0.3200,  1.8691],
        [  0.4871,  3.4290,  1.6660],
        [ -0.5513,  3.4290,  2.1645],
        [  0.5697,  3.4290,  1.4952],
        [  1.2082,  0.7579,  1.8691],
        [  0.0487,  3.4290,  1.9870],
        [ -0.7382, -0.3200,  2.3476],
        [ -1.0906, -0.3200,  2.4186]])
```

A7. We update and release the contest platform.

Please update the contest platform on your local machine.

The variable `problem.pareto_frontier` is an 'nx3' matrix, denoting 'n' points of objective values on the Pareto frontier.

Each point is a '1x3' vector, with each element representing the metric score of performance, power, and area values, respectively.

A better design has higher performance, less power, and a smaller area.

Hence, a better design has larger metric scores, i.e., the larger each element of a '1x3' vector, the better the design is.

We negate the power and area values to calculate the Pareto frontier, aligning with the definition of a better design.

Q8. We suspect that there is a bug in the provided codebase. When computing the golden Pareto frontier of the design space, the power and area values are negated (problem.py, line 110-114).

However, when computing the Pareto frontier of the explored set, these values are not negated (design_space_exploration.py, line 208-228).

A8. We update and release the contest platform. Please check the latest version. If you have any questions, comments, or bug reports, please do not hesitate to contact us.

Q9. Could you please clarify that for each of the three values of the evaluated PPA (the variable `objective_values`, or `y` of the `observe` method), is it higher is better or lower is better?

A9. Each element of the variable `objective_values` or `y` defined in the contest platform denotes normalized performance, power, and area values, respectively. The design is better if it has a higher normalized performance value and lower normalized power and area values. In conclusion, in each element of `objective_values` or `y`, if the first value should be larger, the second and the third value should be smaller, then the microarchitecture is better.

Q10. For the alpha submission of Problem C, should we submit to the TSRI machines, or there is another server for Problem C?

A10. Please submit to the TSRI machines.

Q11. Does the given dummy dataset meet the condition "A higher normalized performance value, a lower normalized power value, and a lower normalized area value suggest that the design is on the true Pareto frontier with a higher possibility" stated in A5?

A11. The dummy dataset meets the condition "A higher normalized performance value, a lower normalized power value, and a lower normalized area value suggest that the design is on the true Pareto frontier with a higher possibility".

Q12. Is the component "time of the VLSI flow" of the dummy case reasonable, i.e., the evaluation of each microarchitecture embedding costs about 30,000+ seconds?

A12. It is reasonable. We leverage a series of commercial electronic design automation (EDA) tools to evaluate and verify each microarchitecture. It does consume much time.

Q13. In the final calculation of ADRS, any two PPAs employed to determine the same Pareto frontier shall never dominate each other. Otherwise, it will affect the calculated result of ADRS. Is it right?

A13. If two objective values fall into the same Pareto frontier, they cannot dominate each other.

Q14. As the number of predicted PPAs grows with our suggestion, ADRS between the predicted Pareto frontier and the golden Pareto frontier will only keep unchanged or decrease but never increase, right?

A14. If your optimizer is reasonable and works well, ADRS will decrease monotonically as your suggestion increases.

Q15. We notice that the dummy design space consists of several sub-embeddings such as ISU and IFU. Does every combination of these sub-embeddings' candidate options always compose a valid microarchitecture? Will the design space of the real dataset also consist of sub-embeddings?

A15. Every combination of these sub-embeddings' candidate options composes a valid microarchitecture. The real dataset consists of these sub-embeddings.

Q16. Following up our previous Q&A, in fact we have observed some abnormality when using the provided codebase of Problem C. As a simple example, we modified the random_search optimizer as shown in the bottom fig, which essentially suggests the configurations in the design space sequentially without randomness. We found that the reported ADRS firstly decreases and then increases (set -q > 100). According to the definition, ADRS should not increase when adding new points to an existing set. Hence we suspect that there might be something wrong within the codebase.

Could you please kindly look into this issue?

```
class RandomOptimizer(AbstractOptimizer):
    primary_import = "iccad_contest"

    def __init__(self, design_space):
        """
        build a wrapper class for an optimizer.

        parameters
        -----
        design_space: <class "MicroarchitectureDesignSpace">
        """
        AbstractOptimizer.__init__(self, design_space)
        self.n_suggestions = 1
        self.suggest_count = 0

    def suggest(self):
        """
        get a suggestion from the optimizer.

        returns
        -----
        next_guess: <list> of <list>
            list of `self.n_suggestions` suggestion(s).
            each suggestion is a microarchitecture embedding.
        """
        self.suggest_count += 1
        x_guess = [self.suggest_count]

        return [
            self.design_space.vec_to_microarchitecture_embedding(
                self.design_space.idx_to_vec(_x_guess)
            ) for _x_guess in x_guess
        ]
```

A16. Thanks for your question. However, we cannot reproduce this error.

We updated the platform yesterday, can you please use the latest version (0.0.3) and let us know if this issue still exists. Thanks.

Q17. We have a question about problem C. Could you please help to check?

We notice that the number of queries (i.e., -q) should be provided when launching the script. It is determined before we learn about the design space, which seems inappropriate. Is there any method we could use to change the number of queries during the optimization? We would appreciate it if any information could be provided. Thank you.

A17. Thanks for your suggestions. We have updated the contest platform. Please update it with the command: pip3 install iccad-contest==0.0.4

You can apply the early stopping criterion conveniently with the latest platform.

Q18. I ran the example code of Problem C. I get Nan in the array. Is this normal behavior?

The error is shown in the picture. Is there any package conflict I should be aware of? Thank you!

```
(base) jovyan@19ddc1f79a50:~/ICCAD/ProblemC$ python3 random-search-optimizer.py -o output -u "00ef538e88634ddd9810d034b748c24d" -q 1
[INFO]: no --solution-settings is specified.
[INFO]: use UUID 00ef538e88634ddd9810d034b748c24d to reproduce the experiment results.
[INFO]: read the sheet Microarchitecture Design Space of excel from /opt/conda/lib/python3.8/site-packages/iccad_contest/contest-dataset/design-space.xlsx
[INFO]: read the sheet Components of excel from /opt/conda/lib/python3.8/site-packages/iccad_contest/contest-dataset/design-space.xlsx
[INFO]: loading from /opt/conda/lib/python3.8/site-packages/iccad_contest/contest-dataset/contest.csv
[INFO]: 1-th query, next microarchitecture embeddings: [[4, 1, 1, 1, 10, 1, 1, 12, 1, 1, 12, 10, 12, 20, 34, 62, 58, 12, 12, 64, 4, 1, 32, 64, 4, 0, 2, 1, 8, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan]] 'suggest' time: 0.00046181678771972656.
[WARN]: failed in evaluating microarchitecture embeddings. set the evaluation of the limit.
[INFO]: evaluate 1-th candidate in a batch, microarchitecture embedding: [4, 1, 1, 1, 10, 1, 1, 12, 1, 1, 12, 10, 12, 20, 34, 62, 58, 12, 12, 64, 4, 1, 32, 64, 4, 0, 2, 1, 8, nan, nan, nan, nan, nan, nan, nan, nan, nan, nan], norm. performance: 0, norm. power: inf, norm. area: inf, time of the VLSI flow: inf.
[INFO]: summary for 1-th query, current ADRS: inf cost: inf.
[INFO]: summary for the solution, the best ADRS: inf cost: inf.
[INFO]: save the experiment result. please refer them to /home/jovyan/ICCAD/ProblemC/output/summary/00ef538e88634ddd9810d034b748c24d.rpt.
```

A18. We cannot reproduce your console results when we execute the same command.

We provide three suggestions:

- 1. Verify that "random-search-optimizer.py" is never modified.**
- 2. Update the platform to the latest: pip3 install iccad_contest==0.0.4**
- 3. Please try numpy == 1.22.4 and pandas==1.4.2**

In the future, we will provide more information about our evaluation machines and the unified submission scripts.

Q19. We have a question about problem C. Could you please help to check? Thank you.

We notice an abnormal phenomenon that the ADRS will increase during the iterations. The log file is attached. We would appreciate it if this issue could be checked!

```
108 [INFO]: summary for 9-th query, current ADRS: 0.6875565805855919 cost: 28
109 [INFO]: 10-th query, next microarchitecture embeddings: [[4, 1, 1, 1, 6,
110 [INFO]: evaluate 1-th candidate in a batch, microarchitecture embedding:
111 [INFO]: evaluate 2-th candidate in a batch, microarchitecture embedding:
112 [INFO]: evaluate 3-th candidate in a batch, microarchitecture embedding:
113 [INFO]: summary for 10-th query, current ADRS: 0.6875565805855919 cost: 2
114 [INFO]: 11-th query, next microarchitecture embeddings: [[4, 1, 1, 1, 6,
115 [INFO]: evaluate 1-th candidate in a batch, microarchitecture embedding:
116 [INFO]: evaluate 2-th candidate in a batch, microarchitecture embedding:
117 [INFO]: evaluate 3-th candidate in a batch, microarchitecture embedding:
118 [INFO]: summary for 11-th query, current ADRS: 0.695191478028017 cost: 29
119 [INFO]: 12-th query, next microarchitecture embeddings: [[4, 1, 1, 1, 8,
120 [INFO]: evaluate 1-th candidate in a batch, microarchitecture embedding:
121 [INFO]: evaluate 2-th candidate in a batch, microarchitecture embedding:
122 [INFO]: evaluate 3-th candidate in a batch, microarchitecture embedding:
123 [INFO]: summary for 12-th query, current ADRS: 0.695191478028017 cost: 36
```

A19. Thank you for the report. We have updated the contest platform. Please update the contest platform to the latest version with the command "pip3 install iccad-contest" or "pip3 install iccad-contest==0.0.5".

Q20. We have a question about problem C. We notice that in the latest release of the codebase (==0.0.5), the computation of ADRS has changed and is now different from that specified in the problem description document (Equation 1).

1. We wonder if this is the case? And if so, is it possible that you could provide some description of the new method of calculating ADRS?

2. Also, according to previous QA, this change seems to aim at fixing the bug that ADRS may occasionally increase with new observations. However, we also observed increase of ADRS even using the new computation method (get_adrs_wrt_polytope). Sorry that we cannot provide a brief usecase to reproduce it.

A20. Please have the answers as follows.

1. The new calculation method is aligned with the problem description document. The function f specified in equation 1 in the document is to be determined. It could be a Euclidean distance or other forms.

The computation of ADRS in 0.0.5 differs from the former version since we think the latest calculation method could be fairer. The function f in the newest version is determined as the distance between the predicted Pareto frontier and the golden Pareto frontier and their enveloped line segments.

We will provide some descriptions of the new ADRS calculating method.

2. If you could provide a detailed example with get_adrs_wrt_polytope, we would be grateful to verify the case.

Q21. We notice that the ADRS will increase during the iterations, but ADRS should not increase when adding new points to an existing set. Could you please help to check it? Thank you.

```
[INFO]: Problem C @ ICCAD'22 contest platform version: 0.0.5
[INFO]: no --resolution-settings is specified.
[INFO]: use UUID f6d051a5dcf94f1692072b5c187a53ea to reproduce the experiment results.
[INFO]: read the sheet Microarchitecture Design Space of excel from D:\anaconda3\envs\my-python37\lib\site-packages\iccad_contest\contest-dataset\design-space.xlsx
[INFO]: read the sheet Components of excel from D:\anaconda3\envs\my-python37\lib\site-packages\iccad_contest\contest-dataset\design-space.xlsx
[INFO]: loading from D:\anaconda3\envs\my-python37\lib\site-packages\iccad_contest\contest-dataset\contest.csv
[INFO]: 1-th query, next microarchitecture embeddings: [[4, 1, 1, 1, 6, 1, 1, 6, 6, 14, 32, 42, 38, 6, 6, 64, 1, 1, 16, 64, 4, 8, 2, 1, 32], [4, 1, 1, 1, 6, 1, 1, 6, 1, 1, 6, 6, 6, 14, 32, 42, 38, 8, 8, 64, 1, 1, 16, 64, 4, 1, 1, 16, 64, 4, 1, 2, 1, 8], [4, 1, 1, 1, 10, 1, 1, 12, 1, 1, 12, 10, 12, 20, 32, 52, 48, 12, 12, 44, 1, 1, 16, 64, 4, 8, 2, 1, 8], [4, 1, 1, 1, 6, 1, 1, 6, 1, 1, 6, 1, 1, 6, 10, 12, 20, 32, 62, 58, 12, 12, 44, 1, 1, 16, 64, 4, 8, 2, 1, 32], [4, 1, 1, 1, 6, 1, 1, 6, 1, 1, 6, 6, 6, 14, 38, 42, 38, 8, 8, 64, 1, 1, 16, 64, 4, 1, 2, 1, 8]] 'suggest' time: 0:08.
[INFO]: evaluate 1-th candidate in a batch, microarchitecture embedding: [4, 1, 1, 1, 6, 1, 1, 6, 1, 1, 6, 6, 6, 14, 32, 42, 38, 6, 6, 64, 1, 1, 16, 64, 4, 8, 2, 1, 32], norm. performance: -1.239940521992794, norm. power: -3.4289647165957824, norm. area: -2.3476322244859853, time of the VLSI flow: 31399.21737.
[INFO]: evaluate 2-th candidate in a batch, microarchitecture embedding: [4, 1, 1, 1, 6, 1, 1, 6, 6, 6, 14, 32, 42, 38, 8, 8, 64, 1, 1, 16, 64, 4, 1, 2, 1, 8], norm. performance: -0.5512598464749289, norm. power: -3.4289647165957824, norm. area: -2.1644551632698867, time of the VLSI flow: 38287.43083.
[INFO]: evaluate 3-th candidate in a batch, microarchitecture embedding: [4, 1, 1, 1, 10, 1, 1, 12, 1, 1, 12, 10, 12, 20, 32, 52, 48, 12, 12, 44, 1, 1, 16, 64, 4, 8, 2, 1, 8], norm. performance: 1.2881893752451849, norm. power: -3.4289647165957824, norm. area: 0.84238942272548825, time of the VLSI flow: 33886.98367.
[INFO]: evaluate 4-th candidate in a batch, microarchitecture embedding: [4, 1, 1, 1, 6, 1, 1, 6, 1, 1, 6, 1, 1, 6, 10, 12, 20, 32, 62, 58, 12, 12, 44, 1, 1, 16, 64, 4, 8, 2, 1, 32], norm. performance: 1.2881893752451849, norm. power: 0.3288828674848886, norm. area: -0.818611318598714383, time of the VLSI flow: 35555.94679.
[INFO]: evaluate 5-th candidate in a batch, microarchitecture embedding: [4, 1, 1, 1, 6, 1, 1, 6, 1, 1, 6, 6, 6, 14, 38, 42, 38, 8, 8, 64, 1, 1, 16, 64, 4, 1, 2, 1, 8], norm. performance: -0.88592458522424963, norm. power: 0.3288828674848886, norm. area: -2.235414474998778, time of the VLSI flow: 38883.87983.
[[[-1.239940521992794, -3.4289647165957824, -2.3476322244859853], [-0.5512598464749289, -3.4289647165957824, -2.1644551632698867], [1.2881893752451849, -3.4289647165957824, 0.84238942272548825], [1.2881893752451849, 0.3288828674848886, -0.818611318598714383], [-0.88592458522424963, 0.3288828674848886, -2.235414474998778]]]
[INFO]: summary for 1-th query, current ADRS: 0.585927537318837 cost: 161850.25689088882.
[INFO]: 2-th query, next microarchitecture embeddings: [[4, 1, 1, 1, 6, 1, 1, 6, 1, 1, 6, 6, 6, 14, 38, 42, 38, 8, 8, 64, 1, 1, 16, 64, 4, 8, 2, 1, 32]] 'suggest' time: 0:0.
[INFO]: evaluate 1-th candidate in a batch, microarchitecture embedding: [4, 1, 1, 1, 6, 1, 1, 6, 1, 1, 6, 6, 6, 14, 38, 42, 38, 8, 8, 64, 1, 1, 16, 64, 4, 8, 2, 1, 32], norm. performance: 0.3969468845927896, norm. power: -3.4112633288979723, norm. area: -2.235414474998778, time of the VLSI flow: 31499.65833.
[[[0.3969468845927896, -3.4112633288979723, -2.235414474998778]]]
[INFO]: summary for 2-th query, current ADRS: 0.6782681847548777 cost: 192549.91524808883.
[INFO]: summary for the solution, the best ADRS: 0.6782681847548777 cost: 192549.91524808883.
[INFO]: save the experiment result. please refer them to E:\iccad_C\output\summary\ofdd51a5dcf94f1692072b5c187a53ea.rpt.
```

A21. We have checked your example answers attached to your email.

The objective values you have visited in the first query are [

- [-1.239940521992794, -3.4289647165957824, -2.3476322244859853],**
- [-0.5512598464749289, -3.4289647165957824, -2.1644551632698867],**

[1.2081893752451849, -3.4289647165957824, 0.04238962272548825],
[1.2081893752451849, 0.3200028674040006, -0.010611318598714383],
[-0.08592658522424963, 0.3200028674040006, -2.2354143841192777]],
and the new objective values you have visited in the second query are [[0.3969468845927896,
-3.4112633288979723, -2.235414474998778]].

We use the example dummy dataset to verify the ADRS for the case.

After the first query, the ADRS is 2.758398106245379, and when the number of visited objective values is enlarged by 1 via the second query, the new ADRS is still 2.758398106245379.

The golden Pareto frontier for the dummy dataset is [

[-1.19281101226806640625, 1.46876895427703857422, 2.34763240814208984375],
[1.20818936824798583984, 3.42896461486816406250, 1.44832527637481689453],
[1.20818936824798583984, 1.14828062057495117188, 1.79813814163208007812],
[0.39694687724113464355, 3.41126322746276855469, 2.23541450500488281250],
[0.05784664675593376160, 3.42896461486816406250, 1.75495266914367675781],
[0.44015416502952575684, -0.32000285387039184570, 2.02253794670104980469],
[-1.23994052410125732422, 3.42896461486816406250, 2.34763216972351074219],
[-1.32202494144439697266, 3.42896461486816406250, 2.41859102249145507812],
[-0.59481227397918701172, -0.32000285387039184570, 2.27667307853698730469],
[1.20818936824798583984, -0.32000285387039184570, 1.86909747123718261719],
[0.48706403374671936035, 3.42896461486816406250, 1.66599953174591064453],
[-0.55125987529754638672, 3.42896461486816406250, 2.16445517539978027344],
[0.56974458694458007812, 3.42896461486816406250, 1.49518275260925292969],
[1.20818936824798583984, 0.75791847705841064453, 1.86909735202789306641],
[0.04871569573879241943, 3.42896461486816406250, 1.98698627948760986328],
[-0.73817759752273559570, -0.32000285387039184570, 2.34763216972351074219],
[-1.09058201313018798828, -0.32000285387039184570, 2.41859102249145507812]
].

We are wondering if you do not use the dummy dataset or if you do not use `get_adrs_wrt_polytope` to calculate the ADRS.

Q22. (As mentioned QA20) Please find the use case that leads to increase of ADRS for your reference. You can run the script by specifying `-q 200`. The increase will happen at around the 70-th and 170-th query.

A22. Thank you for your information. After the discussion internally, we replace the ADRS with the Pareto hypervolume difference, the latter of which will be fairer.

The lower the Pareto hypervolume difference and ORT is, the better the optimizer is.

Please update the contest platform via `"pip3 install iccad-contest=0.0.6"`.

In our updated problem description, you can get the information on the Pareto hypervolume difference.

Q23. We have a question about problem C. In the new version, it seems that the hypervolume is always increasing instead of decreasing. We'd appreciate it if you could check it. Thank you.

A23. In normal cases, the Pareto hypervolume should be increased since the optimizer find a better microarchitecture (higher performance, lower power, and area). Hence, the Pareto hypervolume difference decreases, and a better optimizer has a lower Pareto hypervolume difference and lower ORT.

Q24. We have a question for problem C.

For the hidden datasets, is it true that all the possible combinations of the indices (not the embedding features) are in the design space (the indices of the dummy dataset are shown in the fig)? Also, is this excel file shown in the fig provided for each of the hidden dataset, so that we can rely on the provided APIs in the codebase to access the indices no matter which dataset is used?

	A	B	C	D	E	F	G	H	I	J	K
1	Design	Fetch	Decoder	ISU	IFU	ROB	PRF	LSU	I-Cache/MMU	D-Cache/MMU	Total
2	sub-design-1	1	1	1, 2, 3	1, 2, 3	1, 2, 3, 4	1, 2, 3	1, 2, 3	1, 2, 3, 4	1, 2, 3, 4	5184
3											

A24. For the hidden dataset, all the possible combinations of indices are indeed in the design space. You can rely on the provided APIs in the codebase to access the indices. All designs obey the same index rules.

Q25. Following the Q24,

For example, suppose that a full config embedding is represented by 3 indices. Each index maps to a vector of sub-config embedding. Suppose that the first index has 2 possible sub-configs, and the second and third index have 3 and 4. Then, is it true that in the hidden datasets, all of the $2*3*4=24$ configs is in the dataset/design space (which is the case of the dummy dataset)? Or, is it possible that a dataset/design space could contain less than 24 configs?

A25. All 24 configurations are indeed in the design space.

Q26. We have another question for problem C.

In the README file it is stated that “The design space is extracted from RISC-V BOOM”. Does it mean that for all the hidden datasets/design spaces, the physical meaning of each dimension is the same as the dummy dataset, i.e., as specified in the given excel file, and the difference between datasets/design spaces is the feasible value of each dimension? Or, is it possible that the dimensions in different hidden datasets/design spaces might have different physical meanings?

A26. It means that for all the hidden datasets/design spaces, the physical meaning of each dimension is the same as the dummy dataset.

Q27. Is it normal that we can't see the documents we submitted on the new server <http://47.93.191.38/> after submitting them?

A27. It is normal that you cannot see your submitted files on the server, but you can submit them to replace the old ones. We only accept the last submission for the ranking, and you can see the submission deadline for the following ranking on the page: <http://47.93.191.38/>.

Q28. My problem is: can we use the function 'load_contest_dataset()' in Code 'dataset.py' to directly access all PPA values of the real data you'll use to evaluate our optimizers? If not, how can we access the PPA values of all real data?

A28. You can use the function 'load_contest_dataset()' to directly access all PPA values of the real data.

Q29. There is a noticeable gap in ORT for all teams between Alpha Test and the 7/25 released result. Is the time cost for evaluating each suggestion changed?

A29. The ORT calculation is not changed, except that the inner data set's runtime cost for each microarchitecture differs from the Alpha submission.

Q30. The previous QA12 confirmed a time consumption of 30,000+ seconds for the evaluation of each microarchitecture. Is it just for the dummy dataset? Could you please release the time consumption for the evaluation of each microarchitecture in the real dataset?

A30. The time consumption of 30,000+ seconds is for the dummy dataset. We will update the dummy dataset's runtime cost of each microarchitecture in the contest platform's next version to match the real dataset.

Q31. Can we get the time cost of each query ?

A31. The contest platform has provided the time cost of each query for your reference in the generated report, but it cannot return the runtime cost to your optimizer.

Q32. Thanks for your effort in making the evaluation platform online. From the returned results of online evaluation, surprisingly we found that our algorithm's performance is quite different from that when we develop on the provided dummy dataset, so we think we need to rely heavily on the online platform for future developments of our algorithm. However, currently the platform is updated at a frequency of every two days, which from our perspective is quite slow and we are not able to make any improvements before the result is returned. Therefore, I wonder if it is possible to increase the update frequency of the platform, or even better, make every submission evaluated and shown on the ranking board (from the Q&A it seems that only the last submission within two days will be evaluated now). These will make contestants more active, and possibly enable better results and algorithms being produced in this contest.

A32. Thanks for your reminder. We will improve the evaluation frequency so you can conveniently tune your optimizer.

Q33. For the current setting of the online ranking board, we also have a question: will all the evaluated results (one for every two days) shown on the ranking, or only the best submission for each team is shown? As stated above, we think swift and comprehensive feedback of evaluations are very important for all contestants participating in this problem.

A33. The ranking list is generated according to your last submission before the submission released deadline.

Q34. Also, is there a way to let the contestants know if program error happens during online evaluation using the hidden dataset? (e.g., show the exception logs) Since by the current codebase, the program will not be terminated and instead the suggested points will be randomly selected.

A34. If errors exist during the online evaluation using the hidden dataset, the ranking list will give N/A and related reasons.

Q35. Could you please help to check the following question? Thank you.

Whether the data set of the website test is the same every time, and if so, whether it is the same as the final competition test data set.

A35. The data set of the website test may not be the same every time. The data set will be updated according to the ICCAD Contest Platform. If the design space is updated, the data set will be updated accordingly.

Q36. Since the metric has again changed from HV difference to HV, I wonder if it is possible to let us know the HV of the golden Pareto frontier for the hidden dataset used in the online evaluation, so that we can compare the latest evaluation results with the previous results reported in HV difference?

A36. The metric is updated for a more fair comparison. The golden Pareto frontier of the hidden data set cannot be released

Q37. The dataset provided by platform 0.0.7 has three sub designs. Are they three independent case for optimization and evaluation?

A37. They are not independent cases for the optimization and evaluation, i.e., the exploration is conducted on the whole data set.

Q38. Shall we handle sub-designs for real dataset?

A38. The real data set for all sub-designs are unreleased, but you can use the online submission and ranking platform to measure your solution. The provided dummy data set helps to implement your optimization flow.

Q39. We output the golden_hypervolume Provided by the platform (iccad-contest=0.0.7) and find it equal to 6.352477814164454, is this correct?

A39. It is correct in the dummy data set. The golden Pareto hypervolume may change since the design space is enlarged.