A Phase-Aware Automated Parameter Search Benchmark Framework and Search Behavior Study for OpenROAD Physical Design

Junkai Wang

{2960150W@student.gla.ac.uk}

Glasgow College, UESTC, Chengdu, Sichuan, 611731, China

Abstract. Parameter settings in open-source EDA flows increasingly affect design convergence and physical results, making a reproducible and comparable automated parameter search framework necessary. Based on OpenROAD and OpenROAD-flow-scripts, this paper builds an automated parameter search benchmark that executes flows, parses logs, extracts metrics, and classifies trials into three categories: final, grt_partial, and invalid. Controlled experiments are then conducted on search space design, reward mechanism design, and trial budget. Results show that the framework can reliably execute, parse, and evaluate parameter search processes while preserving valid intermediate physical signals in complex designs. Experiments on AES further indicate that the global search space is more stable across random seeds, the narrow search space achieves better final solutions when successful, the stage-aware reward performs better for some seeds but is seed-sensitive, and increasing the trial budget from 10 to 20 helps markedly, whereas gains from 20 to 30 are limited.

Keywords: OpenROAD; Automated Parameter Search; Physical Design; Design Space Exploration; Stage-Aware Evaluation; Benchmark

1 Introduction

The physical design process in Very Large Scale Integration (VLSI) design consists of several steps, including placement, clock tree synthesis, and routing, that will gradually change a logical netlist into a manufacturable geometric pattern. The number of tunable parameters in the physical design flow keeps increasing as the process nodes are being scaled down, and the relationships between these parameters become progressively more complex. Even minor changes in placement approach, time limitations, and routing heuristics may have a strong influence on the design convergence and the ultimate outcomes of physical results. Optimization problems of this form, where the cost of a single evaluation is high and there are no explicit analytic derivatives, can no longer be addressed using traditional approaches to parameter-tuning based on human intuition and repeated trials and errors - it is no longer efficient or systematic enough to explore the design space. Therefore, the development of an automatable parameter search framework in order to create a reproducible research setup that

ICIAAI 2026, May 22-24, Singapore, Singapore

DOI 10.4108/eai.22-5-2026.2365248

can be compared to the others and is aimed at automating the process of finding the best parameters to use in each design task has great significance.

To resolve this issue, many researchers have performed multifactorial investigations. Shahriari et al. offered a methodological overview of using Bayesian optimization in high expense black-box optimization situations providing a methodological basis of parameter search in physical design [1]. The openness of open-source EDA toolchains (e.g., OpenROAD) facilitates a favorable situation of forming automated experimental systems and gathering the stage-specific measurements, as observed by Etengoff [2]. Geng et al. implemented multi-objective Bayesian optimization in tuning physical design tool parameters to confirm the possibility of the use of the data-driven search strategy to obtain the best possible PPA (Power, Performance, and Area) trade-offs [3]. Ghose et al. introduced the concept of ORFS-agent, which involves the integration of large language models (LLMs) in the OpenROAD tool invocation loop, thus demonstrating the potential of tool-using agents in chip design optimization [4]. Agnesina et al. used deep reinforcement learning to minimize placement-related parameters, demonstrating that intelligent search algorithms do have great potential in the physical design field [5]. Besides, Mutar et al. proposed an improvement to the Auto-PPA approach, i.e., the application of adaptive deep reinforcement learning to VLSI physical design optimization, which supports the further potential of reinforcement learning in parameter search applications [6]. Xu et al. devised the concept of RankTuner, a preference-based Bayesian optimization introduced to the parameter optimization of design tools; they established that automatic search performance depends not only on the optimizer itself but also heavily on how information is managed during the search process [7]. All of these studies point to the conclusion that even though automated parameter search has developed a strong methodological groundwork, it can still experience consistent problems when it comes to complex design - in particular, sparse feedback, search instability, and lack of information utilization during the intermediate design steps. The purpose of this paper is to build a suite of stage-sensitive, automated parameter-search benchmarking tools that could be used specifically in the context of OpenROAD physical design and to employ such a framework to examine search dynamics within complex designs. In cases involving complex designs, most of the trials tend to get stuck at intermediate steps- like global routing. When these trials are simply marked as failures using the return codes of the commands, the intermediate physical information generated until that step, like wirelength or overflow, is ignored completely, and the optimization engine has to deal with very sparse feedback signals. To solve this problem, the current paper separates the execution state of a trial from the presence of its metrics; by using a staged measurement tool, the paper intends to save the comprehensible intermediate physical signals, making them more interpretable and repeatable.

The paper extends this groundwork and perform controlled experiments that are based on the design of the search space, the design of the reward mechanism and design of trial budget. Section 2 presents the experimental framework, stage segmentation, and evaluation methodology; Section 3 provides the experimental settings, results, and analysis; and Section 4 presents the paper summary and future work.

2 Experimental Framework and Advantages

2.1 Experimental Framework and Evaluation Approach

In order to handle the phenomenon of a sparse feedback due to the conventional binary classification of success and failure, in this paper, I propose an automatic parameter-search benchmark based on OpenROAD/ORFS. The framework uses Python scripts to connect the process of parameterization, invocation of workflow, logging and aggregation of results together to form a closed-loop execution model focused on individualized trials. As opposed to conventional methods that use only the end result code, this system is independent of the exit status of the run on the presence of physical measures; it guarantees the inclusion of physically meaningful signals produced at the intermediary phases of sophisticated designs in the later evaluation steps as well.

The framework follows the hierarchical logging paradigm of METRICS 2.1 in terms of metric extraction and it allows extracting important physical quantities, including wirelength, DRC violations and global routing overflow, at different design steps [8]. This framework further sorts the results into three types depending on the physical stage achieved by a trial and the parsability of its measures: final means that the trial has succeeded to achieve the final physical design stage and produced a full set of measures; grt_partial means that, even though the trial has not made the final stage, it produced honest and practical intermediate measures at the global routing stage; and invalid means that none of the valid physical information could be extracted out of the logs. The categorization guarantees that the so-called partial - frequently thrown away as failed attempts of complex designs - will be kept as important intermediate indicators that indicate whether the search is approaching a viable design space [8], [9].

Moreover, the framework is able to work with two reward modes as well: the final and grt_partial samples are given to the optimizer in stage_aware mode; conversely the final samples only are used in final_only mode. With this design, it is possible to directly compare performance based on whether intermediate-stage signals are added or subtracted to the overall search behavior [9].

2.2 The supportive role for subsequent experiments

Following the proposed framework, the experiments given in Chapter 4 are classified into three types of controlled comparisons: The first is a comparison of the global search space with the narrow ones that aim to explore how the design of parameter range affects search coverage ability and the end solution quality; the second is comparison of the final_only and stage_aware rewards to examine the effect of intermediate-stage signal on the search behavior; and the third is the comparison of various trial budget levels to determine the impact of varying budgets on the search success rate as well as end outcomes.

The benefit of such a design is the fact that the experiment is no longer just a simple recording of the findings but revolves around comparing three significant factors systematically: search space, evaluation signals, and trial budget. Unlike regarding OpenROAD as an object of parameter tuning, the benchmark developed in the present work can be perceived more like a reproducible, comparable and understandable experimental platform: it does not only

automate running of experiments, but also keeps intermediate-stage data of complex designs, thus offering a unitary background of following examinations of searching behavior, as well as introducing more sophisticated optimization techniques [3], [4], [7]-[9] .

3 Experiments & Results

3.1 Experiment Setup

The experiments that will help to verify the effectiveness of the designed parameter search benchmark of OpenROAD in this paper will use an open-source RTL-to-GDS physical design flow to act as a base execution platform. Additionally, unified management of parameter sampling, workflow invocation, log parsing, and aggregating of results is achieved through Python scripts. The two representative designs, which were chosen to be the benchmark items were GCD and AES. Among them, GCD is less in scale and its structure is somewhat simple, so it will be used to confirm the viability of the search framework with low complexity designs; AES is larger in scale and is more challenging in physical design, which is what makes it proper to see the interaction between parameter search, stage evaluation and experimental budget in complex designs.

The key elements of physical measurements tracked in the course of an experiment are the amount of wirelength utilized, DRC errors and the existence of a global routing overflow. The outcome of each of these so-called trials can be divided into three classes depending on the physical step achieved at the time of execution as well as whether the relevant metrics were able to be parsed: final, grt_partial, or invalid. Particularly, final means that the trial had successfully landed in the final feasible area and provided a full collection of final physical metrics; grt_partial means that the trial has not arrived at a final feasible point but still enabled the derivation of meaningful physical signals at the intermediary stages; invalid means that the trial has not produced any valid intermediate metrics and it is thus regarded as a failed sample. By such a classification scheme, the benchmark would not depend solely on the end results but would also use the data about intermediate steps in order to give a more refined description of the search process.

Apart from the main benchmark runs, three controlled comparative experiment series were planned. In the second experiment- Experiment 2.1, the effect of search space design is studied where the experiment compares the use of global and narrow search spaces on the design of the AES using a fixed value of `reward_mode=stage_aware`. Experiment 2.2 looks at how the design of the reward mechanism influences things, where the two reward modes are final_only and stage_aware in the same narrow search space in AES. Experiment 2.3 studies the effect of the trial budget by varying `n_trials` (10, 20) and 30 at fixed design, search-space, and reward mechanism. Comparative experiments were conducted across three different random seeds unless otherwise stated {0, 1, 2}, to reduce the effect of the randomness of individual runs in experiments on the results of the experiment.

3.2 Main Benchmark Verification: GCD and AES

There is a significant disparity between how trial stages are distributed among GCD and AES as shown in Figure 1. By contrast, GCD has a higher propensity to create so-called´ final`

samples, where most of AES trials are stuck in what is called as` gra t_partial´. This indicates the fact that the search path to more complex designs does not often go straight into the final feasible region but rather takes a more incremental course i.e. it has some steps where an improvement is made along the way of the global routing phase before reaching the final stage. Moreover, this means that the` gra t_partial´ stage is not a useless failure but an informative intermediate physical metric that may give directional feedback on further sampling cycles.

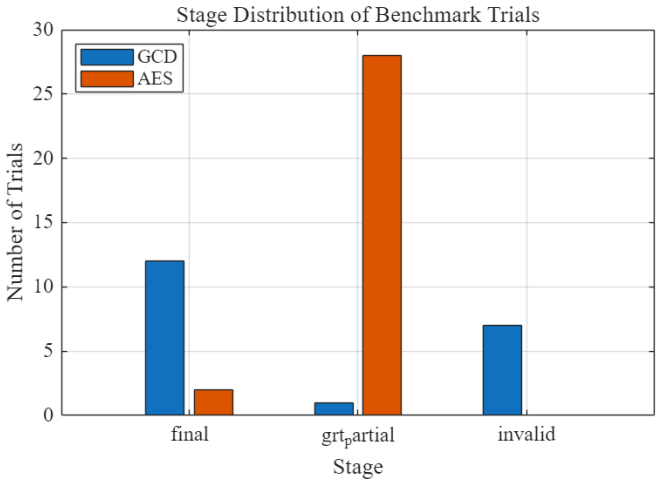

Fig. 1. Comparison of Phase Distributions in GCD and AES Benchmark Tests(Photo/Picture credit : Original)

Talking about GCD, the process of search shown in Figure 1 also indicates that it can reach a so-called final region in relatively short time and continue to provide similar physical metrics to the last stage on consecutive attempts. It confirms the fact that the benchmark developed in this paper possesses strong executability and reproducibility in case of the simple design.

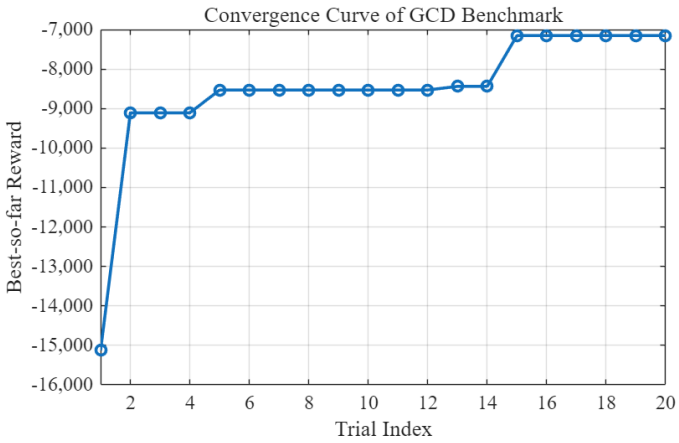

Fig. 2 . Convergence Curve of GCD Benchmark(Photo/Picture credit : Original)

The illustration in Figure 2 shows how GCD converges with the assigned experimental budget. It can be seen that a possible final result comes out at a fairly early point; later enhancements mainly occur as the slow improvement of goal measures in this final area. Such an indication means that in case of simple designs, evaluation signals have a sufficient amount of information and there is a high density of feasible solutions, which allows the optimizer to quickly concentrate on the effective parameter space within a small budget.

By comparison, the search algorithm of the AES model is more complex. Experimental results indicate that with many initial tests, the output achieved is a grt-partial, as opposed to a final result. This suggests that multi-stage designs do not usually enter directly into the feasible final region in one step, but rather pass through intermediate steps to slowly approach the final stage of convergence to a "final" solution

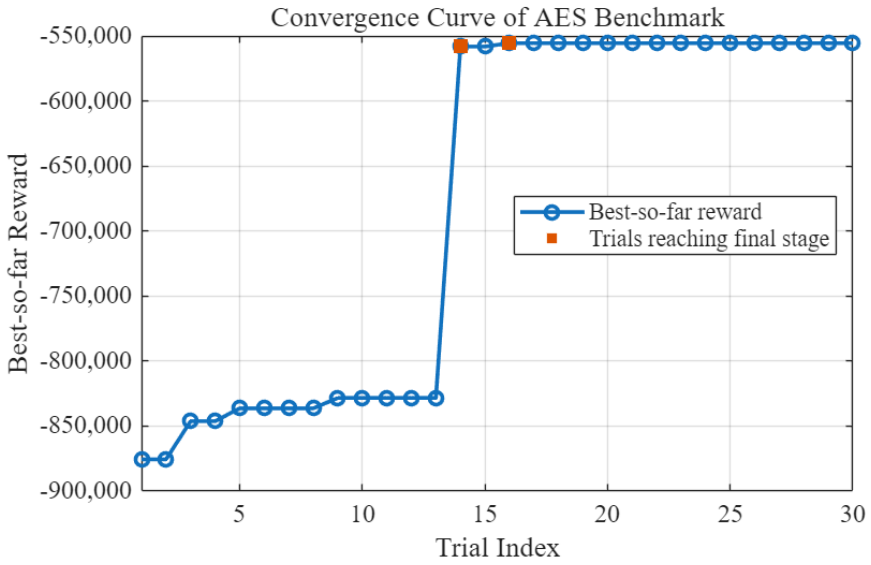

Fig. 3 . Convergence Curve of AES Benchmark(Photo/Picture credit : Original)

In Figure 3, it is depicted that the search process of AES usually consists of progressive improvement of a certain area of the partial region until a major reward jump takes place and the search moves into the last region. It implies that when it comes to the complicated designs such as AES, the binary outcome of the ultimate success or failure cannot be used to illustrate the real search process; rather, signals during the intermediate states may indicate whether the search approaches the feasible region.

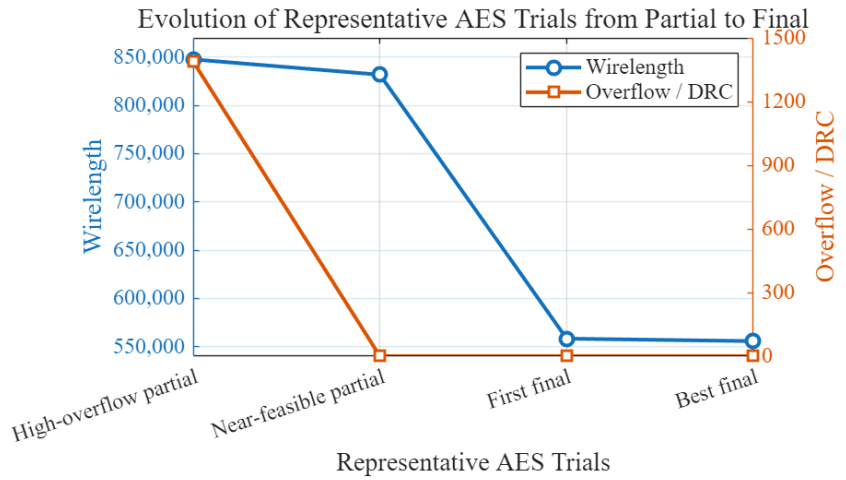

Fig. 4 . Evolution of Representative AES Trials from Partial to Final(Photo/Picture credit : Original)

As shown in Figure 4, the development of representative trials in AES according to the stage model (grt_partial -> final) continues to be illustrated. In addition to those trials which cannot achieve the end detailed routing level, they provide some measurable intermediate metrics as part of the global routing step; most notably, the metrics show similar improvement over time between various trials. Both GCD and AES are shown in their outcomes, and this shows clearly that the benchmark developed in this paper does not only confirm the effectiveness of search processes and metric analysis on simple systems but also conserves physical data of intermediate steps on complicated systems, thus offering a sound base to future controlled comparative experiment.

3.3 The Impact of Search Space Design

The experiments in this part are designed to measure the influence design of search space boundaries have on search behavior during more complex design problems. Concentrating on the AES design, the paper sets the parameter `reward_mode` to `stage_aware` and only compare two different search spaces, i.e., `global` and `narrow`. The `global` space is a larger, initial parameter search space, which means higher coverage of the search area; on the other hand, the `narrow` space is an empirical limit space, which is built based on prior information so that it limits the areas of ineffective exploration and directs the search into areas with prospective large returns. It has already been established by studies that the arrangement of the parameter space plays a major role in shaping search efficiency and ultimate results; therefore, the search space itself must be viewed as one of the essential elements of the benchmarking protocol, not just a minor aspect of its implementation process [3].

Table 1. Comparison Results of Global vs. Narrow Search Spaces on AES

Search Space	Runs Successfully Reaching Final	Total Final Run	Avg. Trial No. of First Entry into Final	Avg. Best Final Wirelength
global	3/3	8	12.7	583356
narrow	2/3	9	11.5	548883

Table 1 illustrates that the "global" search space managed to reach the final stage in three runs (random seeds), with a success rate of 3/3; it has delivered the total number of 8 end solutions with an average of 12.7 trial number on the first entry into the final stage. The contrast is with the "narrow" search space that entered the final stage in just two of the three seeds; however, it provided a larger total of 9 final solutions, with a mean trial number of only 11.5 on the initial entry into the final stage. Moreover, its best averaged final wire length was 548883, lower than 583356 obtained by the global search space. Such findings tell us that a more extensive search space may have a larger percentage of the low-quality samples but still has better coverage ability over various seeds hence reducing the likelihood of not being able to cover the entire feasible region at all. However, experimentally reduced search space eliminates wasteful searching on the edges and helps to find better quality final solutions once the final stage is reached successfully.

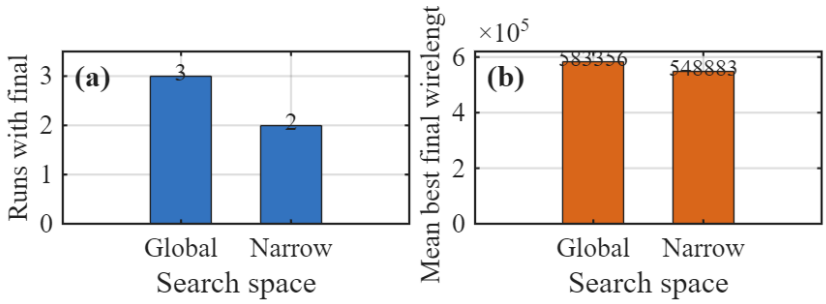

Fig. 5 . Comparison of Global vs. Narrow Search Spaces on AES(Photo/Picture credit : Original)

This contrast is illustrated more intuitively with Figure 5: the number of runs successfully reaching the final stage in global significantly outperforms narrow, which indicates its higher cross-seed stability; on the contrary, in narrow the average best final wirelength of success runs is smaller, which means that it has more ability to produce solution quality as it nears the feasible region. Therefore, Experiment 2.1 does not show a straightforward principle of either the wider being better or the narrower being better, but rather illuminates a fundamental trade-off between search space designs: global focuses on robustness whereas narrow places emphasis on solution quality.

3.4 The Impact of Reward Mechanism Design

These experiments examine how reward mechanism design affects the search process. This comparison is made between two different reward modes within the limited search space of AES: final_only and stage_aware. Final_only mode offers valuable feedback to the optimizer only if a particular trial succeeds in moving through the last detailed routing stage; it considers anything else as non-explorable failures. On the other hand, the stage_ aware mode can include interpretable intermediate measures of the physical quantity that can be obtained at each of the grt_partial stage to form part of the objective function, providing the optimiser with a more continuous feedback signal. Currently available literature suggests that the performance of both reinforcement learning-induced tool tuning and stage-aware parameter optimization techniques is very susceptible to the quality and discriminatory abilities of the

feedback signals and thus, the reward mechanism itself has a significant impact on the search behavior [5], [6], [9].

Table 2. Reward Mechanism Comparison Table

Reward Mode	Runs Successfully Reaching Final Stage	Total Final Runs	Avg. Trial No. of First Entry into Final Stage	Avg. Best Final Wirelength
final-only	2/3	9	4.5	551894
stage-aware	2/3	9	11.5	548883

Both reward schemes have the same metrics concerning the 2/3 value of the metric, i.e., the number of runs successfully reaching the final stage as well as the total number of final-stage runs (9), as illustrated in Table 2, yet they had different search trajectories and end outcomes. With `final_only` scheme, the mean trial count that had been first entered to the final stage was 4.5, and the mean optimal final wire length was 551,894. Conversely, with the `stage_aware` scheme, the mean trial count of the initial entrance into the final stage was 11.5; however, its mean optimal final wire length was 548,883- marginally better than that of `final_only`. This indicates that the reward structure is not only changing when the optimizer changes steps to the final step but also how it explores the feasible area. Although `stage_aware` does not always achieve the final stage in a shorter time, in some runs it provides better final solutions, implying that information about intermediate stages conveyed by physical signals cannot be regarded useless noise, but can contain important structural insights to direct the search.

This seed-sensitive nature of the discrepancy is even more evident in Figure 6. Breaking down the results based on random seed shows that in seeds 0 and 1, the stage_aware method produces a greater amount of final trials compared to final_only; conversely, in seed 2, the outcomes are reversed where final_only shows better performance. Therefore, Experiment 2.2 does not lead to the conclusion that stage_aware is always better than final_only; what is concluded is that the reward mechanism has an intense effect on the search path-and whereas physical cues at intermediary stages may improve the quality of search in some occasions, the advantage is inherently seed sensitive.

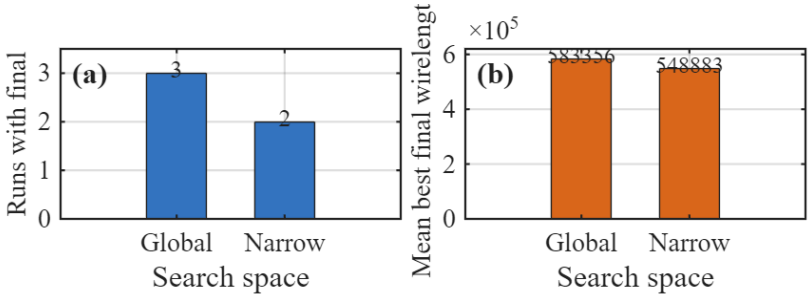

Fig. 6 Comparison of reward mechanisms between final-only and stage-aware approaches under AES narrow search space(Photo/Picture credit : Original)

3.5 Impact of the Trial Budget

The experiment given in this section will determine how the trial budget affects search performance. The experimental setup used in this paper is fixed, namely, it uses an AES strategy, a global search space, and a stage-aware reward mode to ensure that the effect of different trial budgets (n_trials = 10, 20, 30) on the search of a parameter can be compared. The key issue explored in this experiment is: Can the success rate of the search and the quality of the end solution be improved in the case of computationally intensive EDA tuning conditions when the numbers of trials are increased? Moreover, does this enhancement have substantial diminishing marginal returns with higher budgets? Physical design parameter tuning has become a paradigmatic instance of a costly, black-box optimization problem where sample efficiency is a key consideration in establishing the usefulness of a given approach [1]; likewise, studies related to preference-based Bayesian optimization emphasize the need to maximize effective informational gain with restricted budgets [7].

Table 3. Trial Budget Comparison Table

Trial Budget	Runs Successfully Reaching Final	Total Final Run	Avg. Trial No. of First Entry into Final	Avg. Best Final Wirelength
10	0/3	0	-	-
20	3/3	5	12.7	583356
30	3/3	8	12.7	583356

There is a high value in the trial budget on the search success rates, as illustrated by Table 3. A trial budget of 10 (only) did not enable any of the three random seeds to attain the "final" stage, which means that whenever a budget was too low, the system stopped searching without having enough useful information. Increasing the budget to 20 trials enabled all three seeds to achieve the final stage (3/3 final solutions), and the cumulative number of final solutions increased to five. In addition, the mean trial number at which the last stage would be first achieved was 12.7 and the mean best final wirelength was 583,356. It shows that with an increase in the budget between 10 and 20 the performance leaped and the importance of the trial budget at this level of performance should be emphasized to make sure that it can help in crossing the feasibility boundary.

Figure 7 can also be used to demonstrate this particular trend. The runs__with_final metric changed by a factor of 0/3 at 10 trials but rose to 3/3 at 20 trials and remained constant at 30 trials; the total final count however, increased by a factor of 0 to 5 and then increased again to 8. Together with the information provided by Table 3, it is noted that when the budget is raised to 12.7 and 583,356 respectively, the average trial number at which the "final" phase was firstly realized did not change, and the average best final wire-length did not get better. This suggests that the additional investment made in the 20→ 30 bracket merely produced more "final" samples and not necessarily contributed to the increased pace at which "final" was initially achieved or the standard of the optimum solution. To put it differently, the experiments offered in this paper illustrate the principle of diminishing marginal returns in terms of the trial budget: the 10→20 span is an obvious region of improvement, and the 20→30 span is more like an increase in sample size, not a significant improvement in solution quality.

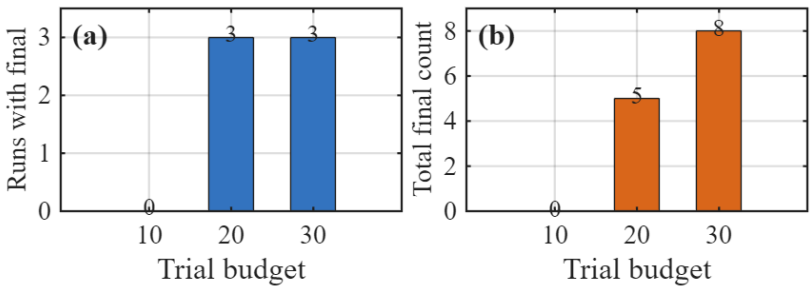

Fig. 7 Comparison of AES search results under different trial budgets(Photo/Picture credit : Original)

3.6 Discussion

Combining the findings of four experiments the paper can conclude that the result of the parameter search based on the OpenROAD is not entirely defined by the optimizer alone but that there exist three types of factors affecting the outcome in general: search space design, the reward mechanism design, and the trial budget. Firstly, Experiment 2.1 proves that the design of the search space is a trade-off between robustness and solution quality as the larger search space improves stability with various random seeds, but smaller empirically restricted search spaces make it easier to reach higher quality end result in successful runs. Secondly, Experiment 2.2 suggests that the reward mechanism has a strong effect on search behaviour; although intermediate-stage physical signals may have useful information in some runs, the value of such signals is sensitive to the particular random seed involved. Lastly, Experiment 2.3 shows that the trial budget has a strong effect on the rate of finding searches in complicated designs but the returns of raising the budget mostly fall between the low-to-moderate budget levels, and additional growth of the budget leads to a quick decline of the marginal return.

The findings highlight that automated physical design search is a non-trivial question that will not be entirely solved just by swapping the optimizer. On the contrary, in order to get more consistent and better quality search outcomes, a systemic approach needs to be taken to a variety of design factors based on the benchmark protocol, specifically, the manner in which parameter ranges are determined, which intermediary-stage messages need to be considered in the evaluation, and what is the optimal way of distributing the computational budget. Research has also shown that when searching over the large scale of parameters, depending on a single optimization method is rarely successful to achieve meaningful increases in the efficiency of tuning, whereas the arrangement of the search space and the development of the search process may play a significant role in determining the ultimate solution achieved [10]. The main point is that the optimizer becomes only a part of the entire system and the search space, the reward mechanism, and the budget configuration form together the holistic features of the search behaviour.

Regarding the contributions to research, this paper does not focus solely on implementing the working OpenROAD parameter search framework but, more importantly, on building an OpenROAD that would enable the researcher to study the very act of search. With primary benchmark validation carried out on GCD and AES, combined with three controlled

comparative experiments, it is shown that searching behavior in complex physical design problems has unique traits of stage-dependency, conditionality, and sensitivity to budget. The present work offers both an experimental basis and a methodological benchmark to subsequent efforts, including the development of more sophisticated optimization methods, the creation of more reasonable search space designs, or investigating stronger evaluation methods sensitive to stages.

4 Conclusion

Following the OpenROAD open-source physical design flow, the paper presents an automated benchmarking system to search parameters in VLSI physical design. The framework uses Python-based invocations of processes, parses logs, segments stages, and designs rewards based on the state of the current step in the process. Moreover, it studies the search behavior in complex designs under three important aspects, including search space, reward mechanisms, and trial budget. As shown, this framework does not just deliver closed-loop automation, including the generation of parameters, the execution of the flow, extraction of metrics, and score of the results, but it does not lose authentic intermediate physical measurements in the `grt_partial` stage, because the existence of the exit status is divorced of the metric availability, which eliminates the easy classification of all the trials that do not have an exit code of zero as unsuccessful ones. Experiment 2.1 implies that the global search space completed the last stage with all 3 seeds while the narrow search space completed 2 out of 3 seeds indicating that the former has better stability across different seeds. Yet, of the successful runs, the 、narrow search space produced a lower average best final wirelength (548,883 versus 583,356), and thus has a larger solution quality potential. Experiment 2.2 shows that both of the reward modes: stage_ aware and final _ only saw two out of the three runs make it to the final stage with a total of nine successful final results. Still, the stage _ aware mode performed marginally better in terms of average best final wirelength compared with final _ only (548,883 versus 551,894), which had different seed-sensitive properties. Experiment 2.3 shows that when the trial budget is 10, none of the 3 runs made it to the end; however, the 20-trial and 30-trial budgets each resulted in a 3-out-of-3 success-rate. It is worth noting that both the 20-trial budget and the 30-trial budget had the same average best final wirelength at 583,356, implying that the additional benefits of increasing the trial budget to 30 would be greatly reduced. All in all, the paper has established that the effectiveness of automated physical design search does not only depend on the optimizer alone, but also on the synergistic design of the search space, evaluation signals, and computational budget. Other future works may include more complete multi-objective optimization approaches and the extension of the framework itself.

References

[1] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, “Taking the Human Out of the Loop: A Review of Bayesian Optimization,” Proc. IEEE, vol. 104, no. 1, pp. 148–175, Jan. 2016.
[2] A. Etengoff, “Empowering innovation: OpenROAD and the future of open-source EDA,” Electrical Engineering News and Products, Jan. 27, 2025.

[3] H. Geng, T. Chen, Y. Ma, and B. Yu, “PTPT: Physical Design Tool Parameter Tuning via Multi-Objective Bayesian Optimization,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 42, no. 1, pp. 178–189, Jan. 2023.
[4] A. Ghose, A. B. Kahng, S. Kundu, and Z. Wang, “ORFS-agent: Tool-Using Agents for Chip Design Optimization,” in Proc. ACM/IEEE Int. Symp. Machine Learning for CAD (MLCAD), 2025, pp. 1–13.
[5] A. Agnesina, K. Chang, and S. K. Lim, “Parameter Optimization of VLSI Placement Through Deep Reinforcement Learning,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 42, no. 4, pp. 1295–1308, Apr. 2023.
[6] H. A. Mutar, I. R. N. AlRubeei, O. H. Yahya, N. A. Hussien, H. T. H. S. AlRikabi, and A. H. M. Alaidi, “Auto-PPA: An Adaptive Deep RL Agent for VLSI Physical Design Optimization,” Journal of VLSI Circuits and Systems, vol. 8, no. 1, pp. 9–19, 2026.
[7] P. Xu, S. Zheng, Y. Ye, C. Bai, S. Xu, H. Geng, T.-Y. Ho, and B. Yu, “RankTuner: When Design Tool Parameter Tuning Meets Preference Bayesian Optimization,” in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), 2024, pp. 122:1–122:7.
[8] J. Jung, A. B. Kahng, S. Kim, and R. Varadarajan, “METRICS2.1 and Flow Tuning in the IEEE CEDA Robust Design Flow and OpenROAD,” in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), 2021, pp. 1–9.
[9] X. Li, D. Luo, P. Xu, Z. Yu, Q. Sun, T. Chen, B. Yu, and H. Geng, “EDA Flow Matters: Stage-Aware Parameter Optimization of Tool Chain,” in Proc. Design, Automation and Test in Europe (DATE), Verona, Italy, Apr. 20–22, 2026.
[10] S. Zheng, H. Geng, C. Bai, B. Yu, and M. D. F. Wong, “Boosting VLSI Design Flow Parameter Tuning with Random Embedding and Multi-objective Trust-region Bayesian Optimization,” ACM Transactions on Design Automation of Electronic Systems, vol. 28, no. 5, pp. 1–23, 2023.