AROPS: A Framework of Automated Reaction Optimization with Parallelized Scheduling

*Yixiang Ruan, Sen Lin, Yiming Mo**

ABSTRACT

With the development of automated experimental platforms and optimization algorithms, chemists can easily optimize chemical reactions in an automated and high-throughput fashion. However, the modules in existing automated experimental platforms are operated in a predefined way without orchestrating with the optimization algorithm, thus leaving room for further efficiency improvement. Here, we introduced a framework of automated reaction optimization with parallelized scheduling (AROPS) to realize the integration of optimization algorithm and module scheduling. AROPS relies on a customized Bayesian optimizer to solve multi-reactor/analyzer reaction optimization problems with three different scheduling modes to arrange tasks for various experimental modules. In addition, a mechanism based on probability of improvement (PI) for discarding unpromising on-going experiments was developed to facilitate freeing-up valuable experimental resources in parallelized optimization. We tested the performance of AROPS using a hardware emulator on three typical benchmark reactions encountered in organic synthesis, illustrating that AROPS can trade off optimization time and cost according to the chemists' preference.

INTRODUCTION

Reaction optimization is an essential routine task for both medicinal chemistry and process chemistry in the pharmaceutical industry.[1] However, optimizing a chemical transformation is challenging owing to its diverse design space, high complexity, and strong nonlinearity. It is estimated that a design space with three continuous variables (e.g., reaction time, temperature, or stoichiometry) and one categorical variable (e.g., catalysts, ligands, or additives) has more than 10^7 combinatorial configurations.[2] Therefore, reaction optimization is excessively expensive and time-consuming, which sometimes becomes the rate-determine step during the synthesis development.

One-factor-at-a-time (OFAT) optimization[3] (i.e., varying only one factor at a time) and factorial design[4] are the most preferred reaction optimization approaches due to their simplicity for post data analysis. These approaches may only give a sub-optimal outcome despite their easy implementation. Recently, with the development of the high-throughput experimentation (HTE) technology,[5–7] exhaustive search of all combinations of reaction condition variables has become possible, such that finding the global optima is feasible within a reasonable experimental budget. In addition, the experimental data generated during the screening process can be used to establish the corresponding chemical reaction database, benefiting building data-driven prediction models.[8–10] However, although HTE platform can accelerate the reaction screening speed by 2-3 orders of magnitude compared to the manual execution, the practical and accessible scale of reaction screening enabled by state-of-the-art HTE technology is still limited to thousands of reactions per day due to the cost and time constraints.

Fortunately, automatic reaction optimization (the combination of optimization algorithms and automated experimental platforms) has been proved to be able to complete optimization with reduced time and cost. Over the years, researchers have been exploring this field to expedite reaction optimization. Nelder–Mead Simplex,[11–13] stable noisy optimization by branch and fit (SNOBFIT) algorithm,[14,15] and the mixed-integer nonlinear program (MINLP) algorithm[16–19] were developed and implemented to optimize various chemical reactions using the automated flow experimental platforms. Moreover, machine learning algorithms like covariance matrix adaption−evolution strategy (CMA-ES) and deep reinforcement learning were also used to optimize microdroplets reactions.[20] However, these algorithms mentioned above all follow a sequential fashion (i.e., only generate one experimental candidate at a time after the previous one has been finished), such that it is not well suited for modern parallelized experimental platforms. On the other hand, Bayesian optimization (BO), a surrogate model-based global optimization algorithm typically used in optimizing expensive black-box functions,[21,22] is well compatible with the sequential or parallel optimization fashion (i.e., generating single/multiple candidate experiments simultaneously).[23,24] In addition to its outstanding performance in tuning hyper-parameters in machine learning models,[25] Bayesian optimization has also been successfully employed for a variety of applications across chemical reaction optimization, such as high-throughput virtual screening,[26] rational solvent selection,[27] the discovery of battery electrolytes,[28] carbon nanotube synthesis,[29] and photocatalysts selection.[30] Doyle and coworkers combined two parallel Bayesian optimizations (i.e., Kriging believer algorithm and Thompson sampling) with HTE platform to realize batched parallel reaction optimization, which can significantly shorten optimization elapsed time compared to traditional sequential optimization.[31]

Although the combination of modern parallelized experimental platforms and Bayesian optimization algorithm has been shown to improve the efficiency of reaction optimization, two new problems arise: (1) The maximum achievable throughput of each module in automated experimental platforms for processing samples is inconsistent. The module with lowest throughput acts like the shortest board in Cannikin Law that limits the overall system throughput. (2) There is no general scheduling method to ensure the efficient cooperation of multiple experimental devices in multiple stages in the optimization process.

To tackle the challenges above, we developed a framework of automated reaction optimization with parallelized scheduling (AROPS) to make more efficient use of automated synthesis screening platform to accelerate the chemical reaction optimization. AROPS is composed of two modules including Bayesian optimizer and parallelization scheduler, responsible for proposing new candidate experiments and assigning devices for executing experiments, respectively. Monte-Carlo Bayesian optimization was employed to optimize reaction, which can find high-quality acquisition function optimal solutions with fewer evaluations compared to heuristic algorithms like Kriging believer and constant liar.[33] In addition, AROPS provides three different scheduling schemes with optional experiment discarding mechanism, which can be selected according to users' preference to time cost or reagent consumption. A multi-reactor/multi-analyzer automated synthesis screening platform simulator was constructed to evaluate the optimization algorithm under various scenarios. Based on this simulator, we comprehensively evaluated AROPS' performance on three benchmarks that covered typical types of reaction mechanism, dataset, optimization objective, and design space encountered in organic synthesis.

METHOD

Bayesian optimizer. Searching for the optimal conditions of chemical reactions can be regarded as analogous to finding the optimal values of unknown objective functions:

$$\boldsymbol{x}^* = \underset{x \in \chi}{\operatorname{argmax}} f(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x}$ denotes the d-dimensional input feature vectors representing reaction conditions, $\boldsymbol{x}^*$ is the optimal conditions, χ is the d-dimensional design space (i.e., $\boldsymbol{x} \in \chi \in \mathbb{R}^d$), and f is the objective function (i.e., reaction yield or selectivity). The Bayesian optimization algorithm is composed of Gaussian process (GP) model and acquisition functions (AF). The Gaussian process is a cheap-to-evaluate approximation to f, which is obtained by regression of the existing data. In this work, GP used the Matérn52 kernel[34] (Eq. S1) to handle continuous variables considering its capability for smooth fitting of nonlinear relationships, and a categorical kernel (Eq. S2) would be added when design space includes categorical variables. The new experiment candidates are proposed by maximizing the acquisition functions, which uses expected improvement (EI) method[35] to tradeoff between exploration and exploitation. When multiple candidates are required in the parallelized platform, multi-points expected improvement (qEI) method will be used:

$$\{\boldsymbol{x}_{new}^{(k)}\}_{k=1}^{q} = \text{argmax}\, q\text{EI}\left(\{\boldsymbol{x}^{(k)}\}_{k=1}^{q}\right) = \text{argmax}\, \mathbb{E}_n \left(\text{ReLu} \left(\max_{i=1,..,q} f(\boldsymbol{x}_i) - f_n(\boldsymbol{x}^+) \right) \right) \quad (2)$$

where $\{\boldsymbol{x}_{new}^{(k)}\}_{k=1}^{q}$ is the q newly proposed reaction conditions, $\boldsymbol{x}^+$ is the current optimal condition, and $\mathbb{E}_n$ indicates that the expectation is taken under the posterior distribution at time n. Since it is challenging to derive the analytical expression of qEI function when q is large, the Monte Carlo sampling is employed instead to simplify the calculation by implementing the numerical multivariate integral approximation.[24,36] More detailed information of Bayesian optimizer in AROPS can be found in the Supporting Information.

Scheduling modes. We constructed a multi-reactor/multi-analyzer platform simulator (RA simulator) to emulate the real-world automated reaction optimization platform for experimental task allocation and elapsed time calculation. It is assumed that reagent preparation and post-reaction workup time can be ignored compared to the reaction and analysis time, and as such, the experiment workflow in RA simulator only has two stages including reaction and analysis. In AROPS, we designed three scheduling modes, namely the synchronous-reaction/batch-analysis (SRBA), synchronous-reaction/instantaneous-analysis (SRIA), and asynchronous-reaction/instantaneous-analysis (ARIA) (Figure 1).

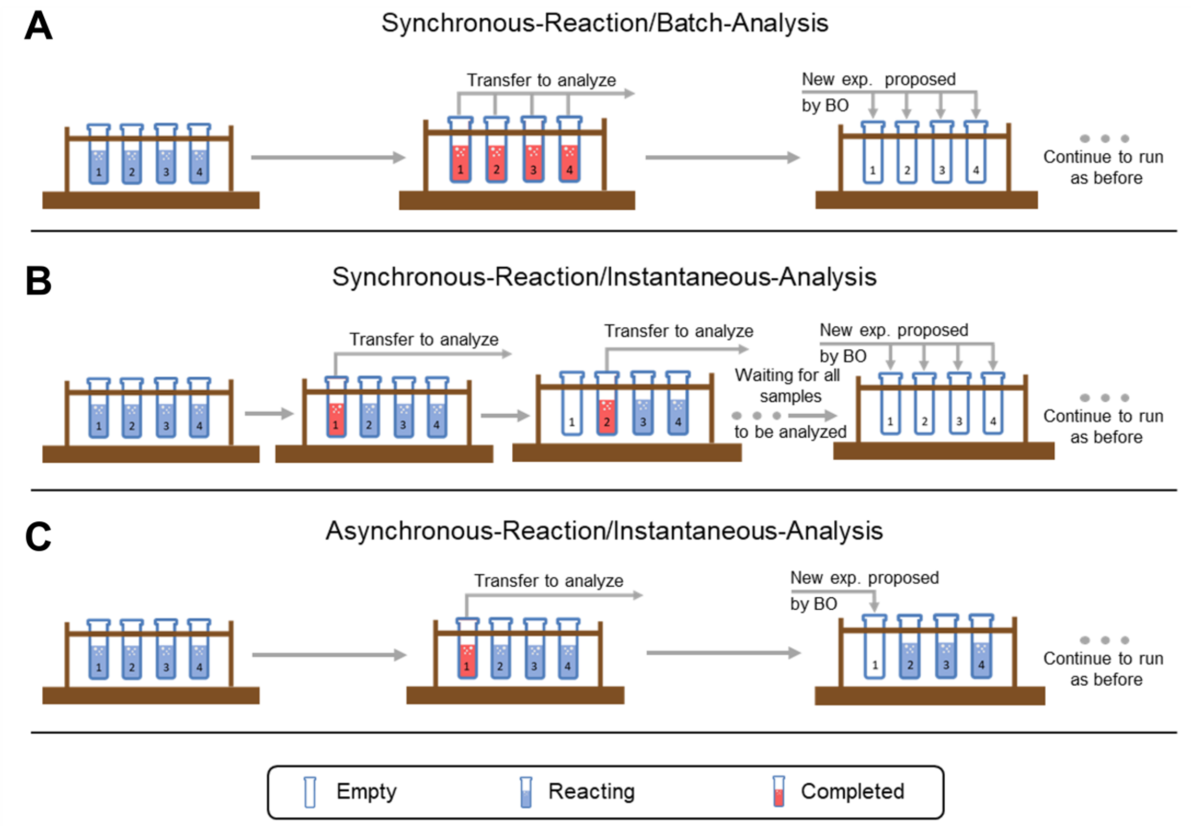

A
Synchronous-Reaction/Batch-Analysis
Transfer to analyze
New exp. proposed by BO
Continue to run as before
B
Synchronous-Reaction/Instantaneous-Analysis
Transfer to analyze
Transfer to analyze
Waiting for all samples to be analyzed
New exp. proposed by BO
Continue to run as before
C
Asynchronous-Reaction/Instantaneous-Analysis
Transfer to analyze
New exp. proposed by BO
Continue to run as before
Empty
Reacting
Completed

Figure 1. The sketch of (A) synchronous-reaction/batch-analysis, (B) synchronous-reaction/instantaneous-analysis, and (C) asynchronous-reaction/instantaneous-analysis. They are three basic scheduling modes for multi-reactor optimization.

The synchronous-reaction (SR) proposes a new batch of experiments simultaneously after previous batch has all completed, and in contrast, the asynchronous-reaction (AR) starts a new experiment whenever there is one previous sample finished. On the other hand, the batch-analysis (BA) means that a batch of samples will be sent for analysis together after they have all finished the reaction, while the instantaneous-analysis (IA) will start analyzing a sample as soon as it ends reaction. However, these three scheduling modes are the identical in the single-reactor/single-analyzer scenario. In the real-world automation hardware, the instantaneous-analysis and asynchronous-reaction require more flexibility in equipment operation than batch-analysis and synchronous-reaction.

PI stopping criterion. The probability of improvement (PI) value is a measure of the possibility that candidates could have an improvement over the current optimal value (Eq.3).

$$\mathrm{PI}(\boldsymbol{x}) = \mathbb{P}(f(\boldsymbol{x}) \geq f(\boldsymbol{x}^{+}) + \xi) = \Phi\left(\frac{\mu(\boldsymbol{x}) - f(\boldsymbol{x}^{+}) - \xi}{\sigma(\boldsymbol{x})}\right) \tag{3}$$

where $\mu(\cdot)$ is GP's mean, $\sigma(\cdot)$ is GP's standard deviation, $\Phi(\cdot)$ is the normal cumulative distribution function, and ξ is the trade-off parameter of exploitation and exploration. If $\xi = 0$, PI is purely exploitative.[22] Therefore, ξ is usually set to a small positive value (0.01 was used in this work following a classic Bayesian optimization library, scikit-optimization[37]) to consider exploration. Compared to the EI value, the PI value has a relatively fixed range between 0 and 1. Moreover, calculating the PI is akin to calculating the p-value for the hypothesis test associated with the null H_0: $\mu(\boldsymbol{x}) > f(\boldsymbol{x}^{+})$. Based on this, Lorenz et al.[38] proposed a stopping criterion by calculating PI value of each new candidate, $\boldsymbol{x}_{new}$, proposed by EI acquisition function. The Bayesian optimization algorithm will be terminated when $\mathrm{PI}(\boldsymbol{x}_{new}) < \alpha$, where α is the significance level. We have adapted this stopping criterion to make it suitable for multi-reactor optimization (the optimization process employing multiple reactors). Newly proposed experiments, whose PI values are below a certain threshold. (e.g., 10^{-2}, 10^{-4}), will be regarded as the unpromising experiments. The optimization will be terminated when the number of proposed unpromising experiments reach a threshold number (e.g., 3). The choice of the threshold tradeoffs between the number of experiments and the confidence of the obtained global optima (see Supporting Information for detailed discussion). A smaller threshold will lead to higher confidence in finding the global optimum and correspondingly more experiments, and vice versa. The Bayesian optimization with this PI stopping criterion is denoted as PISC-BO.

PI discarding mechanism. In addition, we proposed a PI discarding mechanism (real-time PI value smaller than a certain threshold) to remove unnecessary experiments that are at a specific stage (reaction not started or in progress, or waiting for analysis) in the experimental process. It is aimed at avoiding further wasting experimental resources in the multi-reactor optimization. It is worth noting that in the same optimization process, the PI threshold used for PI discarding mechanism is consistent with that for PI stopping criterion. The modified scheduling modes combining the three scheduling modes above with PI discarding mechanism are denoted as SRBA-PI, SRIA-PI, and ARIA-PI.

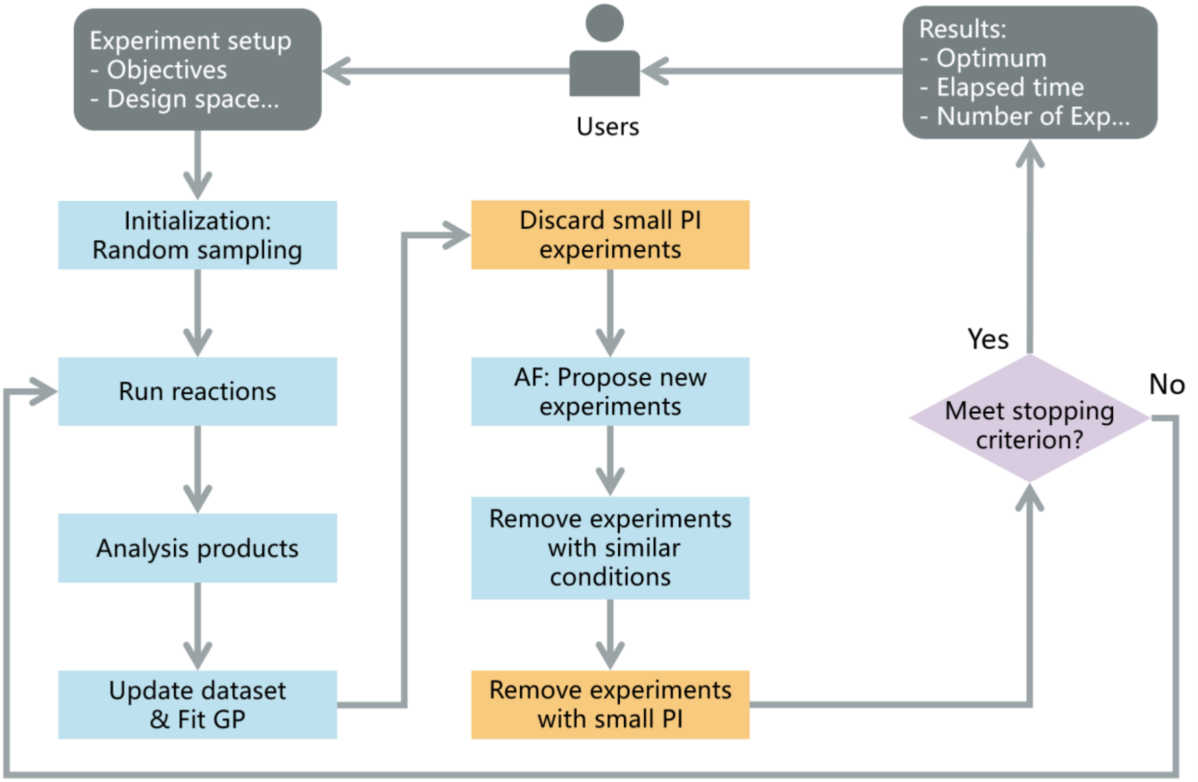


Figure 2. The flow chart of AROPS. The steps in orange are unique to scheduling modes with PI discarding mechanism, and basic scheduling modes will skip these steps.

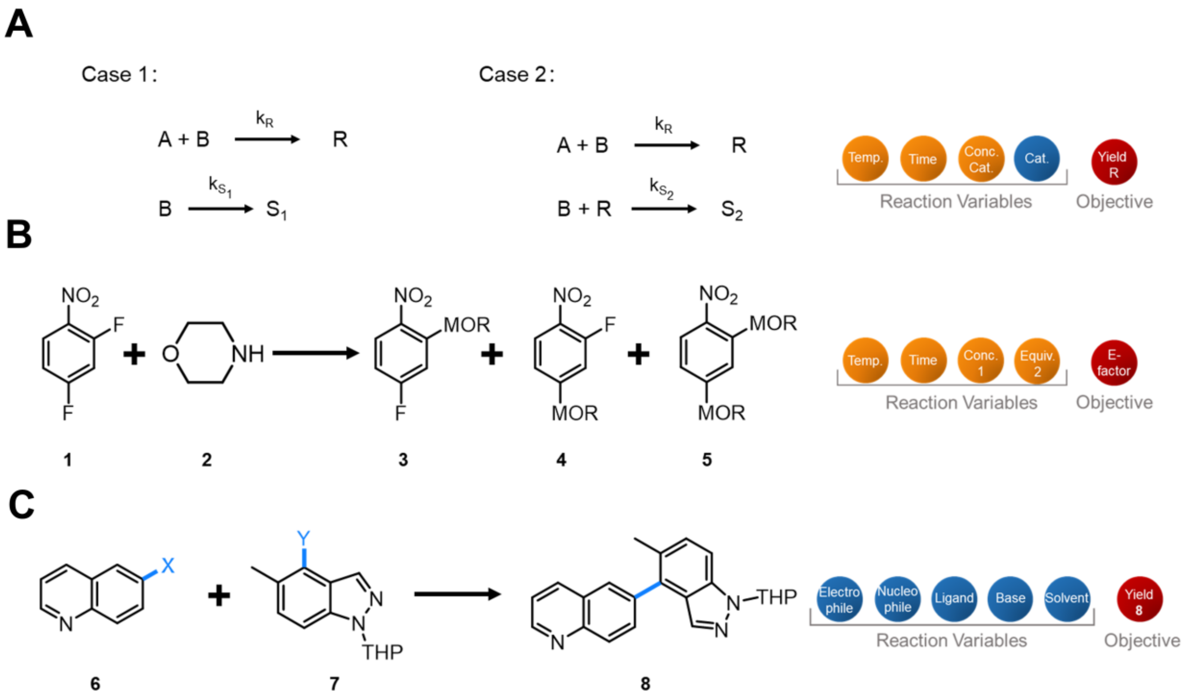
Execution of AROPS. Figure 2 illustrates how AROPS solves a chemical reaction optimization problem. Users first define the design space and prepare the corresponding reagents for the experiments. In the initial phase, random sampling is used to construct the coarse response surface. Then AROPS sequentially executes reaction, performs analysis, updates the data, fits GP model, and examines the PI values of ongoing experiments. Experiments, whose PI values are below a certain threshold, will be discarded avoiding further wasting experimental resources. Then according to the respective scheduling modes, new candidate experiments are proposed. Avoiding executing new experiments similar to previous ones can maximizing the information gain in the optimization task. Thus, the Euclidean distances of conditions' feature vectors between the newly proposed experiments and ongoing and finished experiments are calculated, and newly proposed experiments with a minimal Euclidean distance below a small threshold value will be discarded (e.g., 0.001). In addition, newly proposed experiments with PI values smaller than the threshold will also be discarded. The optimization workflow will continue to iterate until the stopping criterion is met.

Benchmarks. The systematic studies of AROPS were conducted on three different benchmarks (Scheme 1).

Benchmark A (Scheme 1A) includes two reaction kinetic models developed and used by Reizman[16] and Baumgartner.[18] Case 1 introduces a consecutive side reaction in a simple bi-molecular reaction A + B → R to produce the unwanted side product S_1. Case 2 introduces a side reaction yielding the unwanted side product S_2. The yield can be obtained by solving the ordinary differential equations (ODEs). Its design space consists of three continuous variables (i.e., reaction time, temperature, and catalyst concentration) and one categorical variable (i.e., catalyst type). The target is to find the optimal reaction condition to maximize the yield of R. The PI threshold was set to 10^{-4} for this mixed continuous/categorical variable optimization problem with multiple local optima (see Supporting Information for details of selecting the PI threshold).

Benchmark B (Scheme 1B) is a nucleophilic aromatic substitution (S_NAr) reaction containing four continuous variables: reaction time, temperature, reagent concentration and equivalence.[39] The experimental data was regressed using a Bayesian neural network (BNN),[40] and experimental noise was captured by the variance given by BNN. This benchmark was used to test the robustness of AROPS when experimental noise exists. The aim of optimization is to minimize the environment factor (E-factor), defined as the mass ratio of waste (by-products: **4** and **5**) to product (**6**). Since this optimization problem only considers continuous variables, the PI threshold was chosen to be 0.01.

Benchmark C (Scheme 1C) is a Suzuki–Miyaura C–C coupling reaction dataset reported by Pfizer,[5] which was adopted in many works because of its data integrity.[31,41,42] The design space has five categorical variables, consisting of 4 electrophiles, 3 nucleophiles, 12 ligands, 8 bases, and 4 solvents, giving a total of 4608 reaction condition combinations. The objective is maximizing the reaction yield of **8**. This benchmark is introduced to evaluate AROPS' applicability in handling design space only with categorical variables. The design space does not contain reaction time as the variable, namely every reaction completes at the same time. Thus, SRBA and SRIA scheduling modes are essentially the same, and they will be referred to as SRIA for Benchmark C. Since Benchmark C only has categorical variables, the calculated new experiments' PI values are generally more than an order of magnitude larger than those in the other benchmarks. Therefore, a large PI threshold (0.1) is appropriate for Benchmark C.



Scheme 1. Benchmark cases used for evaluating the different performance of AROPS: (A) reaction kinetic models, (B) nucleophilic aromatic substitution reaction, and (C) Suzuki-Miyaura reaction. The continuous and categorical variables represented by yellow and blue balls, respectively. See Supporting Information for the reaction design space and the optimal conditions.

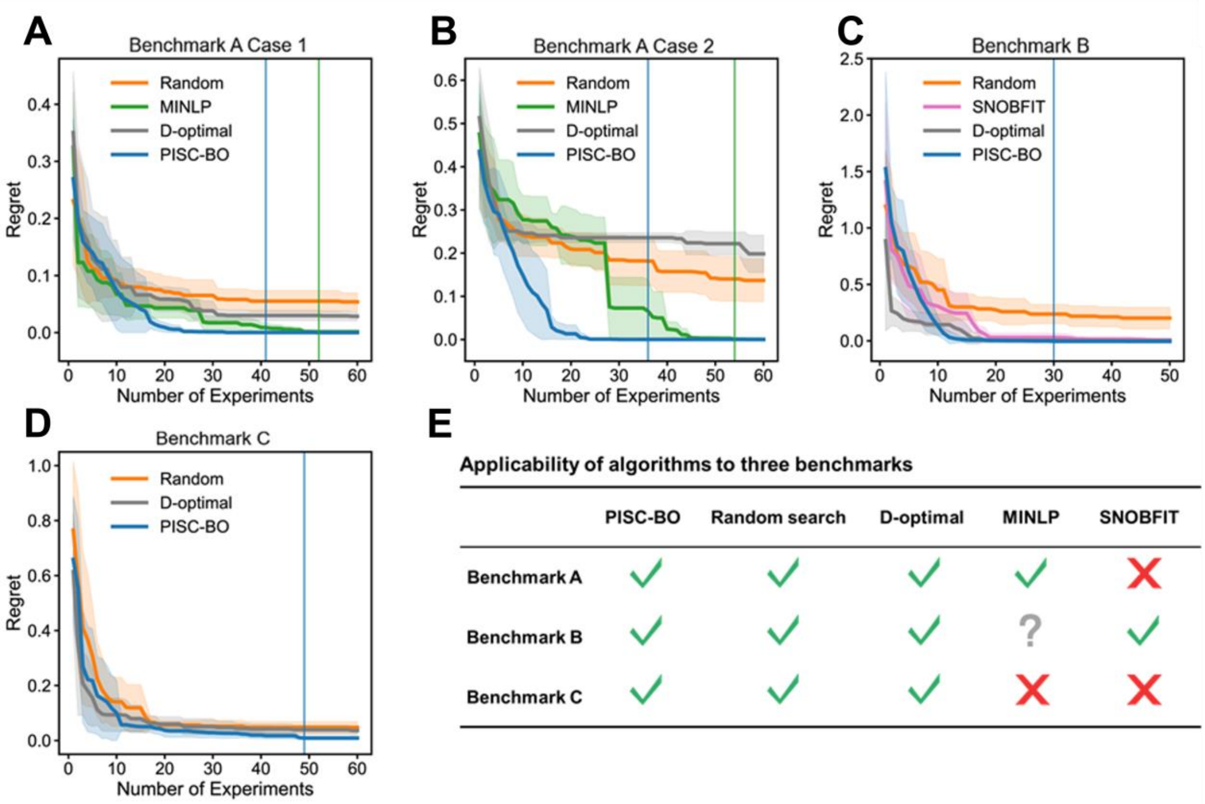
RESULTS & DISCUSSION

Comparison of optimization algorithms. We first sought to compare AROPS' PISC-BO with several black box optimization algorithms on the three benchmarks in the single-reactor optimization (SRO, executed in a sequential fashion), including random search, MINLP, D-optimal design and SNOBFIT. PISC-BO and MINLP have self-termination mechanism when the global optimum is considered to be found, while others require users to specify the number of experimental trials before stop. For fair comparison, we set the same number of experimental trials for each algorithm, but marked the terminating point when PISC-BO and MINLP decided to terminate programmatically. Applicability of these algorithms in three benchmarks is shown in Figure 3E. Random search, D-optimal design, PISC-BO can be applied to all categorical/continuous variable optimization. MINLP algorithm loses its original characteristics when applied to purely continuous or categorical variable optimization, thus evaluations on Benchmark B and Benchmark C were not carried out here. SNOBFIT can only be implemented for continuous variable optimization. Regret is the metric used for quantifying the capability of finding the global optima, which is defined as:

$$\text{Regret} = |f(\boldsymbol{x}^{*}) - f(\boldsymbol{x}^{+})| \tag{4}$$

where $f(\boldsymbol{x}^{*})$ is the true global optimum, and $f(\boldsymbol{x}^{+})$ is the current optimum found by the optimization algorithm. Lower regret indicates better optimization results.

For Benchmark A (Figure 3A-B), neither the random search algorithm nor D-optimal can find the optimal conditions within the given number of experiments, indicating that the grid-like algorithms can only be used for coarse reaction screening. Both PISC-BO and MINLP algorithm can find the optimal reaction conditions, and PISC-BO requires fewer experimental trials than MINLP. For Benchmark B (Figure 3C), D-optimal, SNOBFIT, and PISC-BO all can get ideal optimization results. Since the optimal conditions lie at the boundaries of all variables, which makes the optimization relatively easy for the grid-like D-optimal algorithm. SNOBFIT demonstrates its ability to optimize continuous reaction variables, although it requires more experimental trials than PISC-BO and D-optimal. For Benchmark C (Figure 3D), since there are multiple reaction conditions giving high yields in the design space, the performance of all the algorithms is similar, and the PISC-BO still outperforms the others. In general, PISC-BO outperforms other algorithms significantly by reaching a lower regret value with fewer experimental trials, and it is applicable to the optimization of all types of variable combinations. The PI stopping criterion assists PISC-BO to terminate at an appropriate stage (soon after finding the global optima) instead of manually specifying the number of experiments.



Applicability of algorithms to three benchmarks

	PISC-BO	Random search	D-optimal	MINLP	SNOBFIT
Benchmark A	✓	✓	✓	✓	✗
Benchmark B	✓	✓	✓	?	✓
Benchmark C	✓	✓	✓	✗	✗

Figure 3. Performance comparison of various optimization algorithms (PISC-BO, MINLP, Random search, SNOBFIT, and D-optimal design). The cumulative minimal regret gained by optimization algorithms on (A) Benchmark A Case 1, (B) Benchmark A Case 2, (C) Benchmark B, and (D) Benchmark C. The blue and green vertical lines represent the number of experiments required for PISC-BO and MINLP to terminate, respectively. The aggregated results are from 10 random initializations. (E) The applicability of optimization algorithms to three benchmarks.

Comparison between scheduling modes. In the multi-reactor parallel screening platform, the scheduling of different modules will have a great impact on the optimization efficiency. Among the three scheduling modes mentioned above, SRBA is the most widely used scheduling mode thanks to its simplicity, while SRIA and ARIA require more flexibility in hardware scheduling. Therefore, SRBA was used as a baseline to evaluate the performance of SRIA and ARIA.

As shown in Figure 4, the performance of three scheduling modes was compared in terms of the required optimization time and the number of executed experiments (or reagent consumption). Compared to the single-reactor optimization, the multi-reactor optimization can significantly save elapsed optimization time without increasing optimization regret, but it consumes more experimental trials. Except for Case 2 of Benchmark A, the elapsed optimization time shows a trend as SRBA>SRIA>ARIA. On the other hand, ARIA requires more experiments to complete the optimization compared to SRBA and SRIA. The results of different scheduling modes for Benchmark B and C are similar to those in Case 1 of Benchmark A (Figure S1 and Figure S5).

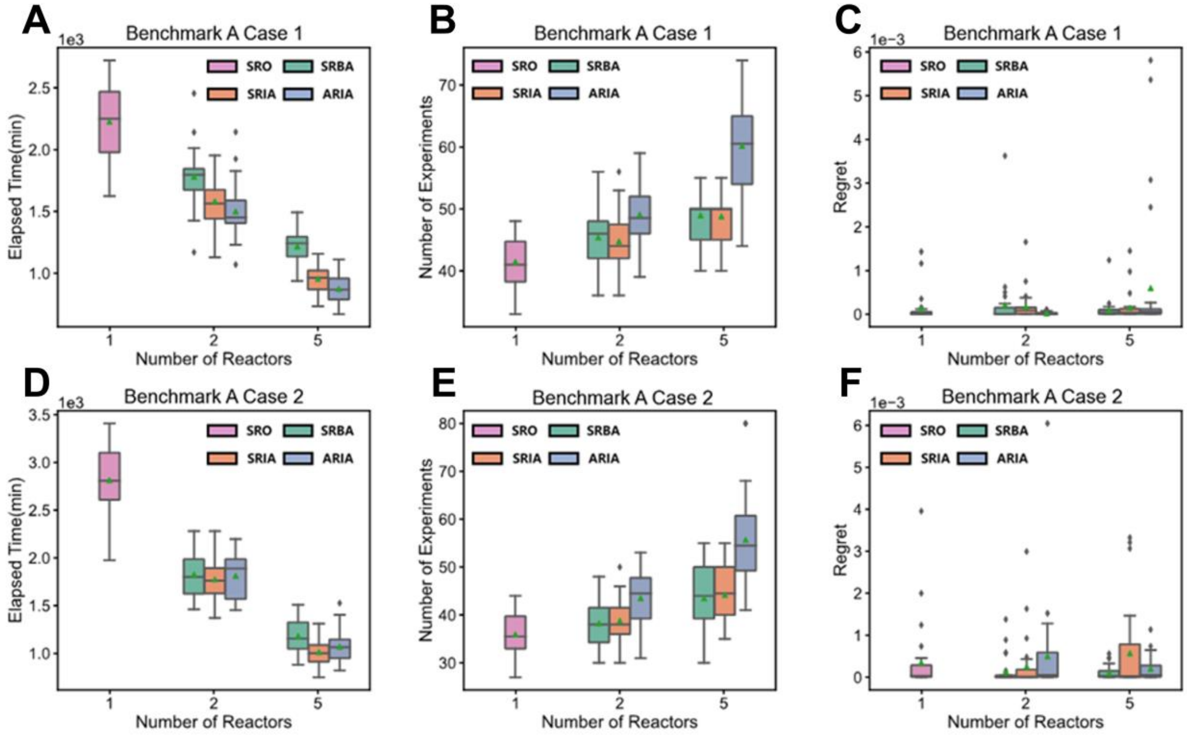


Figure 4. AROPS' performances on Benchmark A with various scheduling modes and different number of reactors. For Benchmark A Case 1 (A-C), the elapsed time (A), required number of experiments (B), and optimization regret (C) are plotted against number of parallelized reactors. For Benchmark A Case 2 (D-F), the elapsed time (D), required number of experiments (E), and optimization regret (F) are plotted against number of parallelized reactors. The results were obtained and averaged using 30 simulated runs with random initialization. The results for other benchmark cases are included in Supporting Information.

Figure 5 illustrates the optimization process examples of the three basic scheduling modes. The utilization ratio of the devices shows a trend as ARIA>SRIA>SRBA, which is the same as the trend of their optimization efficiency. In cases of multi-reactor versus single-reactor or asynchronous-reaction versus synchronous-reaction, the increase in the number of experiments required is due to the fact that multi-reactor or asynchronous-reaction mode has a lower number of evaluated experiments when the same number of experiments are proposed (Figure 6). This is equivalent to sacrificing the quality of the proposed experiments in order to pursue the maximal utilization ratio of the hardware.

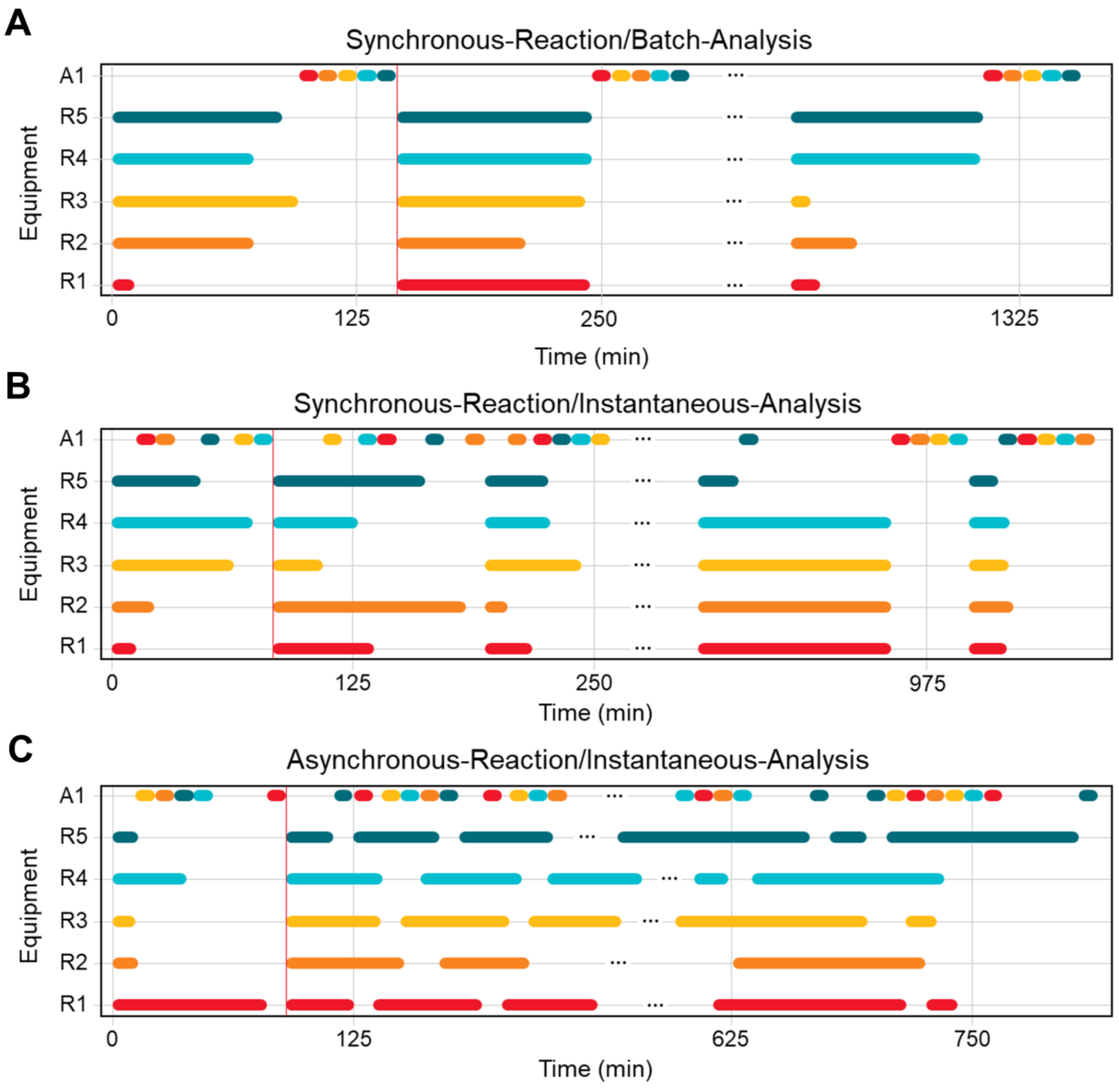


Figure 5. The Gantt charts of three scheduling modes: (A) synchronous-reaction/batch-analysis, (B) synchronous-reaction/instantaneous-analysis, and (C) asynchronous-reaction/instantaneous-analysis. These Gantt charts are plotted for five reactors (R1-R5) and one analyzer (A1) configuration on Benchmark A Case 1. The red vertical lines represent the end of the optimization initialization phase.

As analyzed above, ARIA shows speed advantage on all the benchmarks over the other two scheduling modes thanks to its more efficient usage of the experimental platform. However, ARIA consumes slightly more time than SRIA on Case 2 of Benchmark A (Figure 4D). We speculated that this is because the optimal reaction time is obtained at the boundary of the specified range. In summary, it is reasonable to choose SRIA if the design space contains the reaction time as a variable. For the design space without the reaction time variable, it depends on the user's trade-off between optimization time and cost. Choose ARIA when short optimization time is desired, choose SRIA when small number of experiments is preferred.

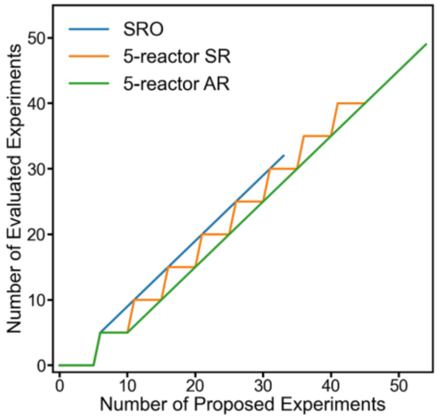


Figure 6. The number of proposed experiments versus the number of evaluated experiments diagram of SRO, 5-reactor synchronous-reaction, and 5-reactor asynchronous-reaction in the optimization process.

PI discarding mechanism. The probability of improvement (PI) discarding mechanism is employed to abandon unnecessary experiments (usually considered to be too explorative or exploitative) in order to free up experimental resources for more promising or informative experiments, and, eventually, save optimization time and cost. This process is similar to the early-stopping strategy in the neural network hyper-parameter optimization, that is, the training of poor performance models is terminated in advance to save elapsed time and computational resources.[43]

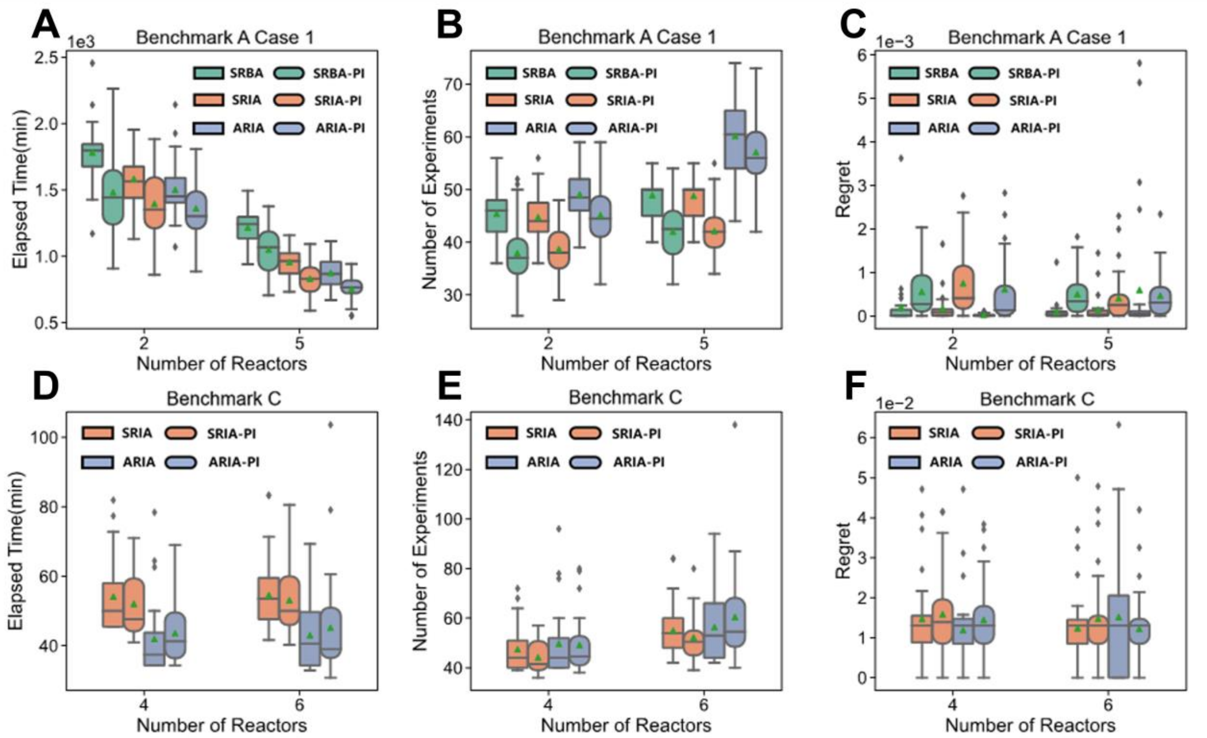


Figure 7. The optimization performances comparison of scheduling modes with and without PI discarding mechanism. For Benchmark A Case 1 (A-C), the elapsed time (A), required number of experiments (B), and optimization regret (C) are plotted against number of parallelized reactors. For Benchmark C (D-F), the elapsed time (D), required number of experiments (E), and optimization regret (F) are plotted against number of parallelized reactors. The results were obtained and averaged using 30 simulated runs with random initialization.

For Benchmark A and B, the modified scheduling modes with PI discarding mechanism typically have 10-20% reduction in time and number of experiments comparing to the corresponding pristine scheduling modes (Figure 7A-B, Figure S2, and Figure S5). Although implementing PI discarding mechanism resulted in slightly increased optimization regret, but still within a negligible range for organic synthesis optimization (Figure 7C). However, for Benchmark C, PI discarding mechanism did not show significant benefits in improving optimizing speed and cost savings (Figure 7D-E). This may arise from that GP has poor prediction accuracy when model inputs only have categorical variables, leading to large inaccuracy in the calculated PI values[31,44] since its derived formula requires the corresponding predicted target values. Thus, some key experiments may be removed during the discarding step because of the inaccurate PI values.

Reactor parallelization. With the rapid development of chemistry HTE technologies, automated experimental platforms can execute up to thousands of experiments simultaneously.[5–7] However, in the process of reaction optimization, due to the limited analysis throughput, the excessive reaction parallelization may not be beneficial to improving overall efficiency in finding the optimal conditions. Therefore, choosing a suitable number of reactors according to the analysis throughput is vital for finding the best balanced configuration in the multi-reactor optimization.

We sought to understand the performance of AROPS under different reactor parallelism and analysis time on Case 1 of Benchmark A. The analysis time was set to 3, 10, and 20 minutes, covering a broad range relative to the optimal reaction time (20 min). The benefits of saving optimization time diminishes as the number of parallelized reactors increases (Figure 8A). This is caused by that the analyzer is too busy to process the finished reaction samples, resulting in the reactors being idle (operating time ratio decreases, Figure 8B).

In order to compare the performance under different number of reactors, we introduced the time-cost trade-off index (TCT index, Eq.5).

$$TCT\ index = \frac{t_q - t_1}{t_1} - k\frac{N_q - N_1}{N_1} \tag{5}$$

where t_q and N_q are the time and number of experiments consumed by AROPS employing q reactors to complete the optimization tasks, respectively. k is the user-defined trade-off parameter ($k > 0$). It is a linear combination of the time reduction (the first term) and the number of experiments rise (the second term) compared to the single-reactor optimization scenario. The larger TCT index is, the smaller time-cost consumption of optimization will be. k depends on the users' preference to time and cost (set to 1 here). A large k value corresponds to users' preference to cost over time, and vice versa.

In the case of SRIA-PI, Figure 8C shows that 5-reactor optimization is the most suitable regardless of the analysis time. Figure S3C indicates that in the case of ARIA-PI, 5-reactor configuration is the best only when the analysis time is 3 min, while 2-reactor configuration gives the best trade-off index with 10 min or 20 min analysis time. This difference should be caused by that the number of experiments required for ARIA-PI grows faster with the number of reactors compared to SRIA-PI. Thus, choosing a suitable parallelized configuration according to the experimental platform throughput and time-cost preference is essential to operate HTE platform in an efficient and affordable manner.

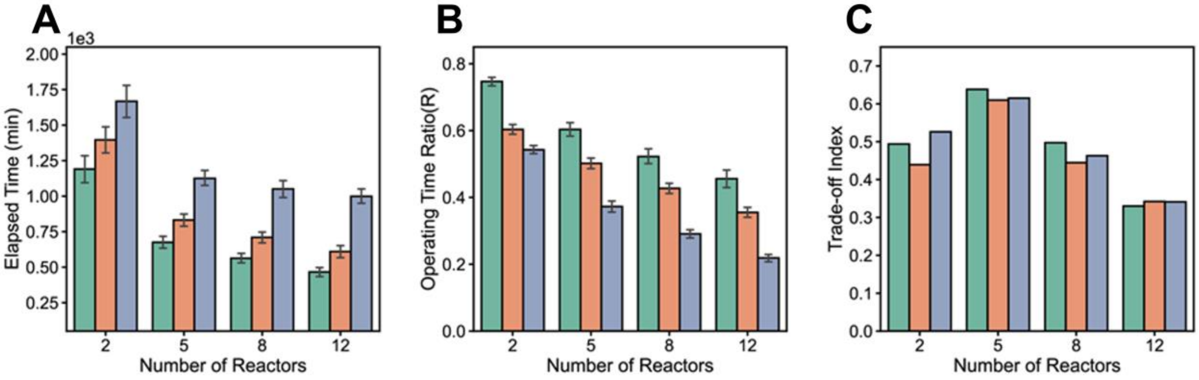


Figure 8. The optimization performances of AROPS (SRIA-PI) under different number of reactors and analysis time (3,10 and 20 minutes) on Benchmark A Case 1. The elapsed time of the optimization processes (A), average operating time ratio of all the reactors (B), and time-cost trade-off index (C) are plotted against number of parallelized reactors. The results were obtained and averaged using 30 simulated runs with random initialization.

Analyzer parallelization. According to the analysis above, under the scenario of 12-reactor optimization with 20 min analysis time on Case 1 of Benchmark A, the average operating time ratio of all reactors is only 22% (SRIA-PI, Figure 8B). The congestion of samples at analysis module is the speed bottleneck, and thus, it would be beneficial to increase the number of analyzers to improve the efficiency. In fact, researchers have equipped the automated experimental platform with multiple analyzers to improve the efficiency of reaction screening.[6] When increasing number of analyzers to 2, 4, and 6, the optimization time was reduced by 34%, 46%, and 49%, respectively (SRIA-PI, Figure 9A). In addition, the reactor operation time ratio increased when analyzer operation time ratio decreased correspondingly (Figure 9B). However, the benefits of time savings significantly diminish when number of analyzers increases from 4 to 6.

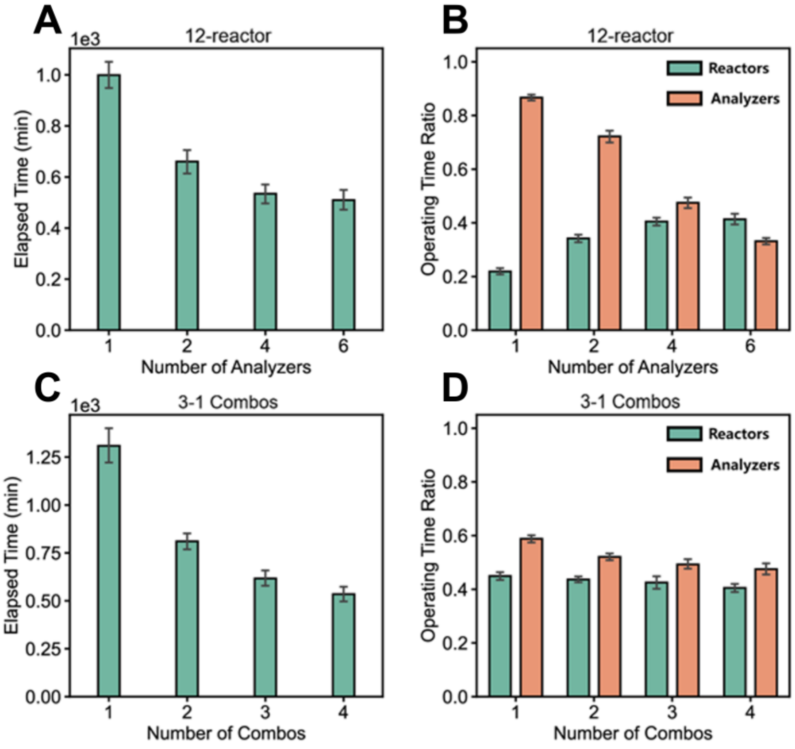


Figure 9. The optimization performance of AROPS (SRIA-PI) with multiple analyzers on Benchmark A Case 1. The elapsed time (A) and average operating time ratio of the reactors and analyzers (B) for 12-reactor configuration are plotted against number of parallelized analyzers. The elapsed time (C) and average operating time ratio of the reactors and analyzers (D) are plotted against different numbers of 3-1 combos. The results were obtained and averaged using 30 simulated runs with random initialization.

To generalize the results above, a certain number of reactors and one analyzer can be considered as a X-1 combo (X represents the number of reactors), such that each combo has a predefined optimization throughput and increasing the number of combos will improve system's overall throughput while keeping a balanced reactor/analyzer ratio. For example, for 12-reactor and 4-

analyzer configuration, it can be regarded as four 3-1 combos. As shown in Figure 9C, increasing the number of 3-1 combos will significantly reduce the time required to find the optimal condition. However, despite the variation in number of combos, the average operating time ratio of the reactors and analyzer remains almost unchanged and stays at a high value (Figure 9D), indicating that 3-1 combo is a proper pairing of reactors and analyzers.

The effect of experimental noise. Despite that the automated experimental platforms can reduce human-induced variations and significantly improve repeatability, some minor experimental noise is sometimes unavoidable (e.g., inaccurate sampling and unstable LC UV-Vis detector). This noise generated can have a potential influence on the optimization process, which is usually manifested in the inability to find the optima and difficulty in optimization termination.

In order to verify AROPS' anti-noise performance, it was first tested on Benchmark B with BNN output. PI discarding mechanism can still effectively save optimization time and cost (Figure S5A-B), and PISC-BO can still successfully find the minimal E-factor (Figure S5C). However, the noise level of Benchmark B is fixed, and it is difficult to quantitatively study the impact of different noise levels. Thus, we introduced Gaussian noises with different levels (Eq.6-7) to the objective function of Benchmark A and C.

$$\hat{f}(\mathbf{x}) = f(\mathbf{x}) + f(\boldsymbol{x}^*)\varepsilon \tag{6}$$

$$\varepsilon \sim \mathcal{N}(\mu, \sigma) \tag{7}$$

where $\hat{f}(\mathbf{x})$ is the objective function with experimental noise, and ε is the experimental noise, which conforms to the Gaussian distribution. The mean μ is 0, and the standard deviation σ is taken as 0, 0.005, 0.01 or 0.05 in this work. A larger σ corresponds to a higher the noise level. It can be found that the addition of noise has little effect on finding the optimal reaction condition (Table S9), but it increases the required number of experiments (i.e., increasing the difficulty in reaching the PI termination criterion, Figure 10). Due to the existence of noise, the model may give better results than the actual optimal value, resulting in the scenario that the PI value is still too high to terminate the optimization after finding the actual optimal conditions. Compared to Benchmark C, the addition of noise has a greater impact on the convergence of Benchmark A, probably caused by the fact that Benchmark A has multiple local optima close to the global optima. Overall, AROPS can effectively handle low-level experimental noise.

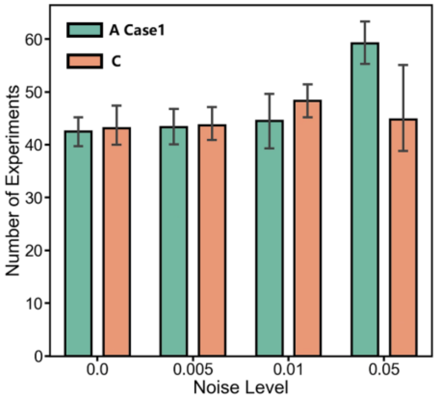


Figure 10. The convergence performances of 5-reactor optimization in AROPS (SRIA-PI) under different levels' Gaussian noise. The aggregated results are from 10 random initializations.

CONCLUSION

In summary, we developed a framework of automated reaction optimization with parallelized scheduling (AROPS) to facilitate reaction optimization. The state-of-the-art Bayesian optimizer (PISC-BO) implemented in AROPS can accommodate multi-reactor/multi-analyzer optimization. We evaluated AROPS' optimization performance on three different benchmarks, covering major types of reaction mechanisms, objectives, and design space encountered in organic synthesis. This work provides guidelines for configurating the multi-reactor optimization (including the scheduling mode and the reactor/analyzer parallelization) based on time-cost preference and experimental hardware. The application of PI discarding mechanism can reduce the elapsed time and costs in the multi-reactor optimization on most occasions. In addition, AROPS can tolerate low-to-medium experimental noise.

Nevertheless, there are also some areas requiring further exploration and development in addition to this proposed framework:

(1) For facilitating the result analysis in this work, we simplified the reaction screening process into two stages, reaction and analysis. However, in practice, this process generally includes sample preparation, reaction, workup, analysis, cleaning, and etc. The scheduling of these steps should also be considered and orchestrated to improve the execution efficiency.

(2) The current performance of using Gaussian process and various molecular descriptors to predict reaction outcomes for reaction conditions that contain only categorical variables is suboptimal, resulting in failure of the PI discarding mechanism for Benchmark C. Therefore, it is beneficial to develop more advanced molecular descriptors and surrogate models that can be applied to Bayesian reaction optimization.

(3) At present, the threshold value in PI discarding mechanism and PI stopping criterion needs to be manually specified. Improper threshold selection may lead to poor reaction optimization results or prolonged optimization time. It is desired to develop a mechanism to programmatically determine a stopping and experiment discarding criterion without human participation. Attempts have been made in other areas to develop such a mechanism, but a universal approach is still lacking.[45–47]

DATA AND SOFTWARE AVAILABILITY

AROPS algorithm and simulation experimental data is publicly available at https://github.com/Ruan-Yixiang/AROPS.

ASSOCIATED CONTENT

Supporting Information

The following files are available free of charge. The supporting description of AROPS and the benchmarks as well as the supporting figures of results section (PDF).

AUTHOR INFORMATION

Corresponding Author

Yiming Mo − College of Chemical and Biological Engineering, Zhejiang University, Hangzhou 310027, China; Hangzhou Global Scientific and Technological Innovation Center, Zhejiang University, Hangzhou 311215, China; Email: yimingmo@zju.edu.cn

Authors

Yixiang Ruan − College of Chemical and Biological Engineering, Zhejiang University, Hangzhou 310027, China; Hangzhou Global Scientific and Technological Innovation Center, Zhejiang University, Hangzhou 311215, China

Sen Lin − Shanghai ChemLex Technology Co, Ltd Shanghai 201210, China

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENT

The authors thank Yan Liu, Yiling Wang, Daisy, Jiayou Zhang, and Dr. Qifeng Yang for helpful discussions.

REFERENCES

(1) Campos, K. R.; Coleman, P. J.; Alvarez, J. C.; Dreher, S. D.; Garbaccio, R. M.; Terrett, N. K.; Tillyer, R. D.; Truppo, M. D.; Parmee, E. R. The Importance of Synthetic Chemistry in the Pharmaceutical Industry. *Science* **2019**, *363* (6424), eaat0805. https://doi.org/10.1126/science.aat0805.

(2) Eyke, N. S.; Koscher, B. A.; Jensen, K. F. Toward Machine Learning-Enhanced High-Throughput Experimentation. *Trends Chem.* **2021**, *3* (2), 120–132. https://doi.org/10.1016/j.trechm.2020.12.001.

(3) Taylor, C. J.; Baker, A.; Chapman, M. R.; Reynolds, W. R.; Jolley, K. E.; Clemens, G.; Smith, G. E.; Blacker, A. J.; Chamberlain, T. W.; Christie, S. D. R.; Taylor, B. A.; Bourne, R. A. Flow Chemistry for Process Optimisation Using Design of Experiments. *J. Flow Chem.* **2021**, *11* (1), 75–86. https://doi.org/10.1007/s41981-020-00135-0.

(4) Surowiec, I.; Vikström, L.; Hector, G.; Johansson, E.; Vikström, C.; Trygg, J. Generalized Subset Designs in Analytical Chemistry. *Anal. Chem.* **2017**, *89* (12), 6491–6497. https://doi.org/10.1021/acs.analchem.7b00506.

(5) Perera, D.; Tucker, J. W.; Brahmbhatt, S.; Helal, C. J.; Chong, A.; Farrell, W.; Richardson, P.; Sach, N. W. A Platform for Automated Nanomole-Scale Reaction Screening and Micromole-Scale Synthesis in Flow. *Science* **2018**, *359* (6374), 429–434. https://doi.org/10.1126/science.aap9112.

(6) Buitrago Santanilla, A.; Regalado, E. L.; Pereira, T.; Shevlin, M.; Bateman, K.; Campeau, L.-C.; Schneeweis, J.; Berritt, S.; Shi, Z.-C.; Nantermet, P.; Liu, Y.; Helmy, R.; Welch, C. J.; Vachal, P.; Davies, I. W.; Cernak, T.; Dreher, S. D. Nanomole-Scale High-Throughput Chemistry for the Synthesis of Complex Molecules. *Science* **2015**, *347* (6217), 49–53. https://doi.org/10.1126/science.1259203.

(7) Mdluli, V.; Diluzio, S.; Lewis, J.; Kowalewski, J. F.; Connell, T. U.; Yaron, D.; Kowalewski, T.; Bernhard, S. High-Throughput Synthesis and Screening of Iridium(III) Photocatalysts for the Fast and Chemoselective Dehalogenation of Aryl Bromides. *ACS Catal.* **2020**, *10* (13), 6977–6987. https://doi.org/10.1021/acscatal.0c02247.

(8) Haywood, A. L.; Redshaw, J.; Hanson-Heine, M. W. D.; Taylor, A.; Brown, A.; Mason, A. M.; Gärtner, T.; Hirst, J. D. Kernel Methods for Predicting Yields of Chemical Reactions. *J. Chem. Inf. Model.* **2022**, *62* (9), 2077–2092. https://doi.org/10.1021/acs.jcim.1c00699.

(9) Żurański, A. M.; Martinez Alvarado, J. I.; Shields, B. J.; Doyle, A. G. Predicting Reaction Yields via Supervised Learning. *Acc. Chem. Res.* **2021**, acs.accounts.0c00770. https://doi.org/10.1021/acs.accounts.0c00770.

(10) Ahneman, D. T.; Estrada, J. G.; Lin, S.; Dreher, S. D.; Doyle, A. G. Predicting Reaction Performance in C–N Cross-Coupling Using Machine Learning. *Science* **2018**, *360* (6385), 186–190. https://doi.org/10.1126/science.aar5169.

(11) McMullen, J. P.; Stone, M. T.; Buchwald, S. L.; Jensen, K. F. An Integrated Microreactor System for Self-Optimization of a Heck Reaction: From Micro- to Mesoscale Flow Systems. *Angew. Chem. Int. Ed.* **2010**, *49* (39), 7076–7080. https://doi.org/10.1002/anie.201002590.

(12) McMullen, J. P.; Jensen, K. F. Integrated Microreactors for Reaction Automation: New Approaches to Reaction Development. *Annu. Rev. Anal. Chem.* **2010**, *3* (1), 19–42. https://doi.org/10.1146/annurev.anchem.111808.073718.

(13) Fath, V.; Kockmann, N.; Otto, J.; Röder, T. Self-Optimising Processes and Real-Time-Optimisation of Organic Syntheses in a Microreactor System Using Nelder–Mead and Design

of Experiments. *React. Chem. Eng.* **2020**, *5* (7), 1281–1299. https://doi.org/10.1039/D0RE00081G.

(14) Parrott, A. J.; Bourne, R. A.; Akien, G. R.; Irvine, D. J.; Poliakoff, M. Self-Optimizing Continuous Reactions in Supercritical Carbon Dioxide. *Angew. Chem. Int. Ed.* **2011**, *50* (16), 3788–3792. https://doi.org/10.1002/anie.201100412.

(15) Coley, C. W.; Thomas, D. A.; Lummiss, J. A. M.; Jaworski, J. N.; Breen, C. P.; Schultz, V.; Hart, T.; Fishman, J. S.; Rogers, L.; Gao, H.; Hicklin, R. W.; Plehiers, P. P.; Byington, J.; Piotti, J. S.; Green, W. H.; Hart, A. J.; Jamison, T. F.; Jensen, K. F. A Robotic Platform for Flow Synthesis of Organic Compounds Informed by AI Planning. *Science* **2019**, *365* (6453), eaax1566. https://doi.org/10.1126/science.aax1566.

(16) Reizman, B. J.; Wang, Y.-M.; Buchwald, S. L.; Jensen, K. F. Suzuki–Miyaura Cross-Coupling Optimization Enabled by Automated Feedback. *React. Chem. Eng.* **2016**, *1* (6), 658–666. https://doi.org/10.1039/C6RE00153J.

(17) Hsieh, H.-W.; Coley, C. W.; Baumgartner, L. M.; Jensen, K. F.; Robinson, R. I. Photoredox Iridium–Nickel Dual-Catalyzed Decarboxylative Arylation Cross-Coupling: From Batch to Continuous Flow via Self-Optimizing Segmented Flow Reactor. *Org. Process Res. Dev.* **2018**, *22* (4), 542–550. https://doi.org/10.1021/acs.oprd.8b00018.

(18) Baumgartner, L. M.; Coley, C. W.; Reizman, B. J.; Gao, K. W.; Jensen, K. F. Optimum Catalyst Selection over Continuous and Discrete Process Variables with a Single Droplet Microfluidic Reaction Platform. *React. Chem. Eng.* **2018**, *3* (3), 301–311. https://doi.org/10.1039/C8RE00032H.

(19) Baumgartner, L. M.; Dennis, J. M.; White, N. A.; Buchwald, S. L.; Jensen, K. F. Use of a Droplet Platform To Optimize Pd-Catalyzed C–N Coupling Reactions Promoted by Organic Bases. *Org. Process Res. Dev.* **2019**, *23* (8), 1594–1601. https://doi.org/10.1021/acs.oprd.9b00236.

(20) Zhou, Z.; Li, X.; Zare, R. N. Optimizing Chemical Reactions with Deep Reinforcement Learning. *ACS Cent. Sci.* **2017**, *3* (12), 1337–1344. https://doi.org/10.1021/acscentsci.7b00492.

(21) Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; de Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE* **2016**, *104* (1), 148–175. https://doi.org/10.1109/JPROC.2015.2494218.

(22) Brochu, E.; Cora, V. M.; de Freitas, N. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. *ArXiv10122599 Cs* **2010**.

(23) Häse, F.; Roch, L. M.; Kreisbeck, C.; Aspuru-Guzik, A. Phoenics: A Bayesian Optimizer for Chemistry. *ACS Cent. Sci.* **2018**, *4* (9), 1134–1145. https://doi.org/10.1021/acscentsci.8b00307.

(24) Ginsbourger, D.; Riche, R. L.; Carraro, L. A Multi-Points Criterion for Deterministic Parallel Global Optimization Based on Gaussian Processes. **2008**.

(25) Snoek, J.; Larochelle, H.; Adams, R. P. Practical Bayesian Optimization of Machine Learning Algorithms. *Adv. Neural Inf. Process. Syst.* **2012**, *25*.

(26) Griffiths, R.-R.; Hernández-Lobato, J. M. Constrained Bayesian Optimization for Automatic Chemical Design Using Variational Autoencoders. *Chem. Sci.* **2020**, *11* (2), 577–586. https://doi.org/10.1039/C9SC04026A.

(27) Amar, Y.; Schweidtmann, A. M.; Deutsch, P.; Cao, L.; Lapkin, A. Machine Learning and Molecular Descriptors Enable Rational Solvent Selection in Asymmetric Catalysis. *Chem. Sci.* **2019**, *10* (27), 6697–6706. https://doi.org/10.1039/C9SC01844A.

(28) Dave, A.; Mitchell, J.; Kandasamy, K.; Wang, H.; Burke, S.; Paria, B.; Póczos, B.; Whitacre, J.; Viswanathan, V. Autonomous Discovery of Battery Electrolytes with Robotic Experimentation and Machine Learning. *Cell Rep. Phys. Sci.* **2020**, *1* (12), 100264. https://doi.org/10.1016/j.xcrp.2020.100264.

(29) Chang, J.; Nikolaev, P.; Carpena-Núñez, J.; Rao, R.; Decker, K.; Islam, A. E.; Kim, J.; Pitt, M. A.; Myung, J. I.; Maruyama, B. Efficient Closed-Loop Maximization of Carbon Nanotube Growth Rate Using Bayesian Optimization. *Sci. Rep.* **2020**, *10* (1), 9040. https://doi.org/10.1038/s41598-020-64397-3.

(30) Burger, B.; Maffettone, P. M.; Gusev, V. V.; Aitchison, C. M.; Bai, Y.; Wang, X.; Li, X.; Alston, B. M.; Li, B.; Clowes, R.; Rankin, N.; Harris, B.; Sprick, R. S.; Cooper, A. I. A Mobile Robotic Chemist. *Nature* **2020**, *583* (7815), 237–241. https://doi.org/10.1038/s41586-020-2442-2.

(31) Shields, B. J.; Stevens, J.; Li, J.; Parasram, M.; Damani, F.; Alvarado, J. I. M.; Janey, J. M.; Adams, R. P.; Doyle, A. G. Bayesian Reaction Optimization as a Tool for Chemical Synthesis. *Nature* **2021**, *590* (7844), 89–96. https://doi.org/10.1038/s41586-021-03213-y.

(32) Welch, C. J. High Throughput Analysis Enables High Throughput Experimentation in Pharmaceutical Process Research. *React. Chem. Eng.* **2019**, *4* (11), 1895–1911. https://doi.org/10.1039/C9RE00234K.

(33) Wang, J.; Clark, S. C.; Liu, E.; Frazier, P. I. Parallel Bayesian Global Optimization of Expensive Functions. *Oper. Res.* **2020**, *68* (6), 1850–1865. https://doi.org/10.1287/opre.2019.1966.

(34) Minasny, B.; McBratney, Alex. B. The Matérn Function as a General Model for Soil Variograms. *Geoderma* **2005**, *128* (3–4), 192–207. https://doi.org/10.1016/j.geoderma.2005.04.003.

(35) Mockus, J. On the Bayes Methods for Seeking the Extremal Point. *IFAC Proc. Vol.* **1975**, *8* (1), 428–431. https://doi.org/10.1016/S1474-6670(17)67769-3.

(36) Balandat, M.; Karrer, B.; Jiang, D. R.; Daulton, S.; Letham, B.; Wilson, A. G.; Bakshy, E. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. *ArXiv191006403 Cs Math Stat* **2020**.

(37) *scikit-optimize: sequential model-based optimization in Python — scikit-optimize 0.8.1 documentation*. https://scikit-optimize.github.io/stable/ (accessed 2022-07-30).

(38) Lorenz, R.; Monti, R. P.; Violante, I. R.; Faisal, A. A.; Anagnostopoulos, C.; Leech, R.; Montana, G. Stopping Criteria for Boosting Automatic Experimental Design Using Real-Time FMRI with Bayesian Optimization. *ArXiv151107827 Q-Bio Stat* **2016**.

(39) Schweidtmann, A. M.; Clayton, A. D.; Holmes, N.; Bradford, E.; Bourne, R. A.; Lapkin, A. A. Machine Learning Meets Continuous Flow Chemistry: Automated Optimization towards the Pareto Front of Multiple Objectives. *Chem. Eng. J.* **2018**, *352*, 277–282. https://doi.org/10.1016/j.cej.2018.07.031.

(40) Häse, F.; Aldeghi, M.; Hickman, R. J.; Roch, L. M.; Christensen, M.; Liles, E.; Hein, J. E.; Aspuru-Guzik, A. Olympus: A Benchmarking Framework for Noisy Optimization and Experiment Planning. *Mach. Learn. Sci. Technol.* **2021**, *2* (3), 035021. https://doi.org/10.1088/2632-2153/abedc8.

(41) Gong, Y.; Xue, D.; Chuai, G.; Yu, J.; Liu, Q. DeepReac+: Deep Active Learning for Quantitative Modeling of Organic Chemical Reactions. *Chem. Sci.* **2021**, *12* (43), 14459–14472. https://doi.org/10.1039/D1SC02087K.

(42) Granda, J. M.; Donina, L.; Dragone, V.; Long, D.-L.; Cronin, L. Controlling an Organic Synthesis Robot with Machine Learning to Search for New Reactivity. *Nature* **2018**, *559* (7714), 377–381. https://doi.org/10.1038/s41586-018-0307-8.

(43) Domhan, T.; Springenberg, J. T.; Hutter, F. Speeding up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves. *Int. Conf. Artif. Intell. AAAI Press* **2015**. https://doi.org/10.5555/2832581.2832731.

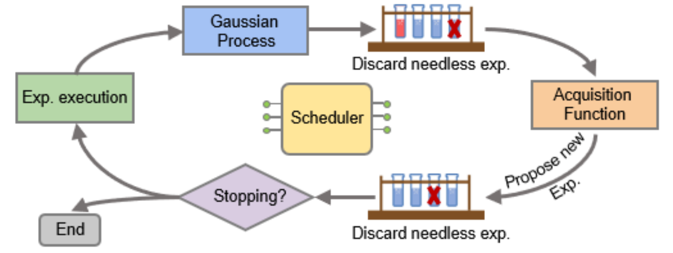
(44) Pomberger, A.; Pedrina McCarthy, A. A.; Khan, A.; Sung, S.; Taylor, C. J.; Gaunt, M. J.; Colwell, L.; Walz, D.; Lapkin, A. A. The Effect of Chemical Representation on Active Machine Learning towards Closed-Loop Optimization. *React. Chem. Eng.* **2022**, 10.1039.D2RE00008C. https://doi.org/10.1039/D2RE00008C.

(45) Tan, J.; Nayman, N.; Wang, M. CobBO: Coordinate Backoff Bayesian Optimization with Two-Stage Kernels. arXiv April 19, 2022.

(46) McLeod, M.; Osborne, M. A.; Roberts, S. J. Optimization, Fast and Slow: Optimally Switching between Local and Bayesian Optimization. arXiv May 22, 2018.

(47) Kim, B.; Shin, M. Bayesian Optimization of MOSFET Devices Using Effective Stopping Condition. *IEEE Access* **2021**, *9*, 108480–108494. https://doi.org/10.1109/ACCESS.2021.3101812.

For Table of Contents Use Only

	AROPS: A Framework of Automated Reaction Optimization with Parallelized Scheduling Yixiang Ruan, Sen Lin, and Yiming Mo*	 Gaussian Process Discard needless exp. Exp. execution Scheduler Acquisition Function Propose new Exp. Stopping? Discard needless exp. End