SLAS Technology
2021, Vol. 26(6) 650–659

DOI: 10.1177/24726303211021790
journals.sagepub.com/home/jla
SAGE

Optimal Scheduling for Laboratory Automation of Life Science Experiments with Time Constraints

Takeshi D. Itoh[1,2*], Takaaki Horinouchi[3,4*], Hiroki Uchida[5], Koichi Takahashi[3], and Haruka Ozaki[6,7*]

Abstract

In automated laboratories consisting of multiple different types of instruments, scheduling algorithms are useful for determining the optimal allocations of instruments to minimize the time required to complete experimental procedures. However, previous studies on scheduling algorithms for laboratory automation have not emphasized the time constraints by mutual boundaries (TCMBs) among operations, which is important in procedures involving live cells or unstable biomolecules. Here, we define the "scheduling for laboratory automation in biology" (S-LAB) problem as a scheduling problem for automated laboratories in which operations with TCMBs are performed by multiple different instruments. We formulate an S-LAB problem as a mixed-integer programming (MIP) problem and propose a scheduling method using the branch-and-bound algorithm. Simulations show that our method can find the optimal schedules of S-LAB problems that minimize overall execution time while satisfying the TCMBs. Furthermore, we propose the use of our scheduling method for the simulation-based design of job definitions and laboratory configurations.

Keywords

scheduling, laboratory automation (LA), time constraint by mutual boundaries (TCMB), branch-and-bound algorithm

Introduction

Automating experimental procedures in life science is important for stabilizing quality, saving cost, and improving efficiency. For instance, thermal cyclers were invented to automate the complicated PCR procedures that had previously required manual transfer of labware between water baths.[1] Recent advances in mechanical engineering and robotics have enabled the automation of more advanced and complex experimental procedures in the life sciences through the development of specialized instruments for large-scale genome editing,[2] library preparation for next-generation sequencing (NGS),[3] cell culture,[4–6] omics measurements,[7,8] and high-throughput assays.[9–12] Automating these kinds of advanced experimental procedures reduces the cost of training human operators, dependency on experts, and the burden of repeating procedures. On the other hand, multipurpose instruments (e.g., liquid handling workstations and dual-arm humanoid robots) have also been actively developed because they offer the flexibility to carry out various kinds of experiments.[13–16] However, despite the development of these laboratory automation instruments, one generally needs to use several different types of instruments to execute even one procedure

[1]Mathematical Informatics Laboratory, Graduate School of Science and Technology, Nara Institute of Science and Technology, Ikoma, Nara, Japan
[2]Anchorage Works, Inc., Kobe, Hyogo, Japan
[3]Laboratory for Biologically Inspired Computing, RIKEN Center for Biosystems Dynamics Research, Suita, Osaka, Japan
[4]Laboratory for Multiscale Biosystem Dynamics, RIKEN Center for Biosystems Dynamics Research, Suita, Osaka, Japan
[5]Tecan Japan Co., Ltd., Kawasaki, Kanagawa, Japan
[6]Bioinformatics Laboratory, Faculty of Medicine, University of Tsukuba, Tsukuba, Ibaraki, Japan
[7]Center for Artificial Intelligence Research, University of Tsukuba, Tsukuba, Ibaraki, Japan
*These authors contributed equally to this research.

Received Jan 19, 2021, and in revised form May 5, 2021. Accepted for publication May 11, 2021.

Supplemental material is available online with this article.

Corresponding Authors:
Haruka Ozaki, Bioinformatics Laboratory, Faculty of Medicine, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8575, Japan.
Email: haruka.ozaki@md.tsukuba.ac.jp

Koichi Takahashi, Laboratory for Biologically Inspired Computing, RIKEN Center for Biosystems Dynamics Research, 6-2-3 Furuedai, Suita, Osaka 565-0874, Japan.
Email: ktakahashi@riken.jp

composed of multiple steps[17,18] because a single instrument is rarely capable of performing all the steps in the procedure.

When coordinating multiple types of instruments to execute procedures, it is often essential to reduce the execution time required to complete the procedures while avoiding resource conflicts. We can formulate this as a scheduling problem, such as the job shop problem[19,20] and its variants. In scheduling problems for automated laboratories, the procedure is divided into operations that are processed by multiple instruments in a given order.[21] The objective of the scheduling problem is to find a schedule that allocates instruments to each operation so that all procedures are completed in a short execution time without resource conflicts. Several scheduling approaches have been proposed for laboratory automation in, for example, image-based cellular assays,[22] clinical blood tests,[23] bacterial engineering,[24] protein crystallization,[25] high-throughput screening,[26] and materials research.[27] Most of these scheduling algorithms fall into two categories: rule-based algorithms and mathematical optimization algorithms. Rule-based algorithms are specifically designed for specific scheduling problems and are easy to understand for users.[23,24,27] For instance, Delaney et al. developed a dynamic scheduling method that selects the operation with the earliest prescribed time among the uncompleted operations as the next operation for execution.[24] Likewise, Burger et al. scheduled tasks so that the robot starts the oldest queued job on instruments in parallel.[27] In contrast, mathematical optimization algorithms aim to find an optimal schedule for minimizing the entire execution time. In particular, metaheuristic algorithms, a group of stochastic search algorithms that balance intensification and diversification strategies,[28] have been widely applied to scheduling problems in laboratory automation.[25,26,29] For example, Cabrera et al. applied simulated annealing to scheduling plate imaging of protein crystallization.[25] Recently, genetic algorithms have been used for scheduling complex life science workflows in laboratory infrastructure composed of distributed automation systems connected by robot transporters and human operators.[26,29] However, these scheduling methods do not guarantee the global optimality of the solution because they are based on metaheuristic algorithms,[25,26,29] partly due to the computational burden of exact algorithms for mathematical optimization problems, that are guaranteed to find the global optimal solution.

Previous work on scheduling algorithms for laboratory automation has not focused on the time constraints by mutual boundaries (TCMBs) among operations. A TCMB is the upper limit of the time difference between the start or end (boundary) of an operation and the boundary of another.[21] For example, a TCMB can be used to represent a situation such as where operation B must be started within 10 min after operation A ends. These kinds of TCMBs exist widely in life science experiments.[30–32] In particular, procedures involving live cells or unstable biomolecules (e.g., RNA or enzymes) require strict time constraints to avoid alteration, denaturation, or degradation of the samples. Therefore, scheduling methods for use in the life sciences need to ensure that solutions satisfy the TCMBs.

In this study, we defined the "scheduling for laboratory automation in biology" (S-LAB) problem as a scheduling problem in which operations with TCMBs are performed by multiple different instruments. We formulated the S-LAB problem as an instance of the mixed-integer programming (MIP) problem and proposed a scheduling method using the branch-and-bound algorithm, an exact algorithm for MIP problems. By performing a scheduling simulation, we demonstrated that our method can find optimal schedules of S-LAB problems that minimize overall execution time while satisfying the TCMBs. Furthermore, we also proposed the use of our scheduling method for designing job definition and laboratory configuration.

Materials and Methods

Definition of Terms

- Protocol: A description of how to complete a procedure. A protocol is composed of multiple experimental steps.
- Experimental step: The smallest unit in a protocol.
- Job: The counterpart to a procedure in a scheduling problem. A job is composed of a series of operations.
- Operation: The smallest unit for scheduling in a job. An operation consists of one or more experimental steps and is processed by an instrument.

S-LAB Problem

We formulated the S-LAB problem based on the job shop problem, one of many classical scheduling problems.[19,20] In the job shop problem, there are (1) multiple types of instruments and (2) jobs consisting of multiple operations processed in a certain order. Each operation has a process time and a type of instrument that processes it. The objective of the job shop problem is to find a schedule that allocates instruments to each operation so that all jobs are completed in the minimum execution time without resource conflicts.

We formulated the S-LAB problem for an automated laboratory composed of multiple instruments as follows (**Fig. 1**).

Job Definition

- There are M instruments in a laboratory, each of which has an instrument type T_m ($1 \leq m \leq M$, $1 \leq T_m \leq K$), where K is the number of different instrument types.

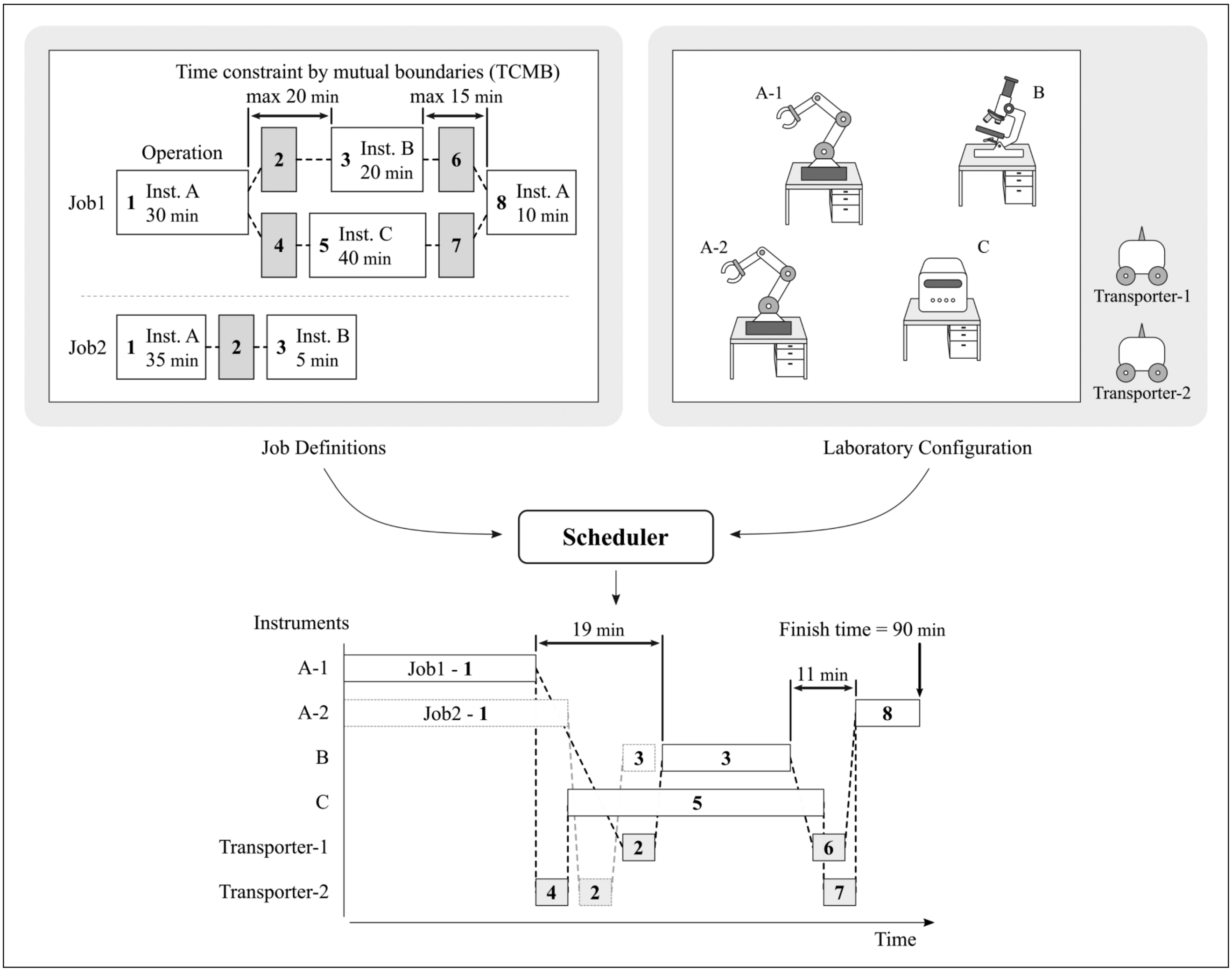

Figure 1. Overview of the S-LAB problem. **Top left:** Job definition. A job consists of one or more operations, the dependency graph among operations, and TCMBs for pairs of operations. An operation, represented by a rectangle, has an index, a compatible instrument type, and processing time. The operation dependency graph is a directed acyclic graph, in which each node represents an operation and each dotted line represents the dependency between a pair of operations, in which one must start after another ends. A TCMB specifies the maximum time difference between a pair of operation boundaries. **Top right:** Laboratory configuration. The laboratory is equipped with multiple types of instruments. Transporters transport samples among instruments. **Bottom:** The scheduling method reads the job definitions and the laboratory configuration to schedule the optimal allocation of the instruments to the operations to minimize the entire execution time while satisfying the TCMBs. The diagram shows an example of a scheduling result, where the horizontal axis represents the time and the vertical axis represents the instruments. A box represents the process time for an operation. A dotted line represents the dependency between a pair of operations. Inst., instrument.

- There are J jobs.
- The j-th ($1 \le j \le J$) job is composed of N_j operations. In total, there are $N = \sum_{j=1}^{J} N_j$ operations.
- The a-th operation O_a ($1 \le a \le N$) has a compatible instrument type C_a ($1 \le C_a \le K$); that is, O_a needs to be processed by an instrument m with the instrument type $T_m = C_a$.
- O_a has process time τ_a, which is intrinsically determined by the combination of the nature of the operation and the instrument type C_a to process it.
- There are dependencies among operations; that is, one operation must start after another ends. The operation dependency graph P is a directed acyclic graph, in which each node represents an operation and each directed edge represents the dependency between a pair of operations:

$$P_{a,b} = \begin{cases} 1\left(\text{if } O_b \text{ must start after } O_a \text{ ends}\right), \\ 0\left(\text{otherwise}\right). \end{cases}$$

- There must be a buffer time between a pair of operations consecutively processed on the same instrument. The length of the buffer time β ($\beta > 0$) is determined by the user.

- Each instrument has parking positions through which an instrument passes samples to transporters or receives them from transporters. Here, we assume that each instrument has a sufficient number of parking positions so that samples can wait if the instrument is occupied by another operation.

Scheduling Solutions

- *A schedule* is defined by determining the start time S_a and the processor E_a for each O_a.
- E_a ($1 \leq E_a \leq M$) is the instrument that processes O_a, chosen from the set of compatible instruments that have the type $T_m = C_a$.

Constraints

- When multiple instruments are allocated to multiple operations, one instrument can process at most one operation at a time.
- The order of operations defined by P needs to be maintained.
- There are I TCMBs. The i-th ($1 \leq i \leq I$) TCMB sets the upper limit α_i of the maximum tolerable difference between the start or end time of each of a pair of operations O_a and O_b. If a TCMB is specified as the difference between the start times of two operations, then $\left|S_a - S_b\right| < \alpha_i$. If a TCMB is specified as the difference between the start time of one operation and the end time of another, then $\left|S_a - (S_b + \tau_b)\right| < \alpha_i$. If a TCMB is specified as the difference between the end time of one operation and the start time of another, then $\left|(S_a + \tau_a) - S_b\right| < \alpha_i$. Otherwise, a TCMB is specified as the difference between the end times of two operations and $\left|(S_a + \tau_a) - (S_b + \tau_b)\right| < \alpha_i$.

Objective

- Based on these definitions and constraints, a scheduling method attempts to find the *optimal schedule* that minimizes the entire execution time of the jobs, that is, the end time of the lastly processed operation

$$\max_a \left(S_a + \tau_a\right).$$

- The objective of our method is to minimize this value, as in

$$\min \max_a \left(S_a + \tau_a\right).$$

Formulation as a MIP Problem

We formulate the S-LAB problem as an instance of the MIP problem and propose a scheduling method using the branch-and-bound algorithm.[33] The branch-and-bound algorithm is an exact algorithm, which is guaranteed to find a global optimal solution, for MIP problems. To search for the optimal solution efficiently in the solution space, the branch-and-bound algorithm recursively prunes (narrows) the solution space so that the algorithm can avoid performing an exhaustive search. For details of the problem formulation as a MIP problem and the branch-and-bound algorithm, see Supplemental Material Sections 1 and 2.

Implementation

We implemented the proposed scheduling method for the S-LAB problem in the Julia language,[34,35] using JuMP.jl[36] and Cbc.jl, a Julia API for an open-source MIP solver named Cbc.[37] The code is available at https://github.com/labauto/SLab.jl.

Results

To validate the utility of the proposed method, we simulated the scheduling of several jobs having typical properties observed in real biological experiments. We formulated these experimental procedures as S-LAB problems and solved the problems using the proposed method. We set $\beta = 1.0$ min for all the simulations below. We conducted these simulations on a computer with an Intel Core i7-8700K CPU.

Scheduling Jobs with Various Time Constraints

Dealing with TCMBs is an important aspect of the S-LAB problem. Let us consider scheduling several jobs with TCMBs in a laboratory equipped with three types of instruments (**Suppl. Table S1**). **Figure 2a** shows a schematic diagram of case I-A, which consists of two serial operations in which the latter operation must start within 10 min after the former ends (**Suppl. Tables S2–S4**). **Figure 2b** shows a diagram of case I-B, which consists of three operations in which two operations are processed in parallel and the difference between their start times must be less than or equal to 1 min (**Suppl. Tables S5–S7**). **Figure 2c** shows case I-C, in which we set a 1 min TCMB between the end times of the two operations (**Suppl. Tables S8–S10**). **Figure 2d–f** shows the scheduling results for each case, demonstrating that the proposed scheduling method found the optimal schedules for the jobs and that the TCMBs in the job definitions are satisfied. The computations took 0.84, 0.81, and 0.82 s for cases I-A, I-B, and I-C, respectively.

Scheduling Jobs with Branching and Convergence

In some life science experiments, the protocols might have branching dependencies in which multiple experimental steps depend on the same preceding step or converging

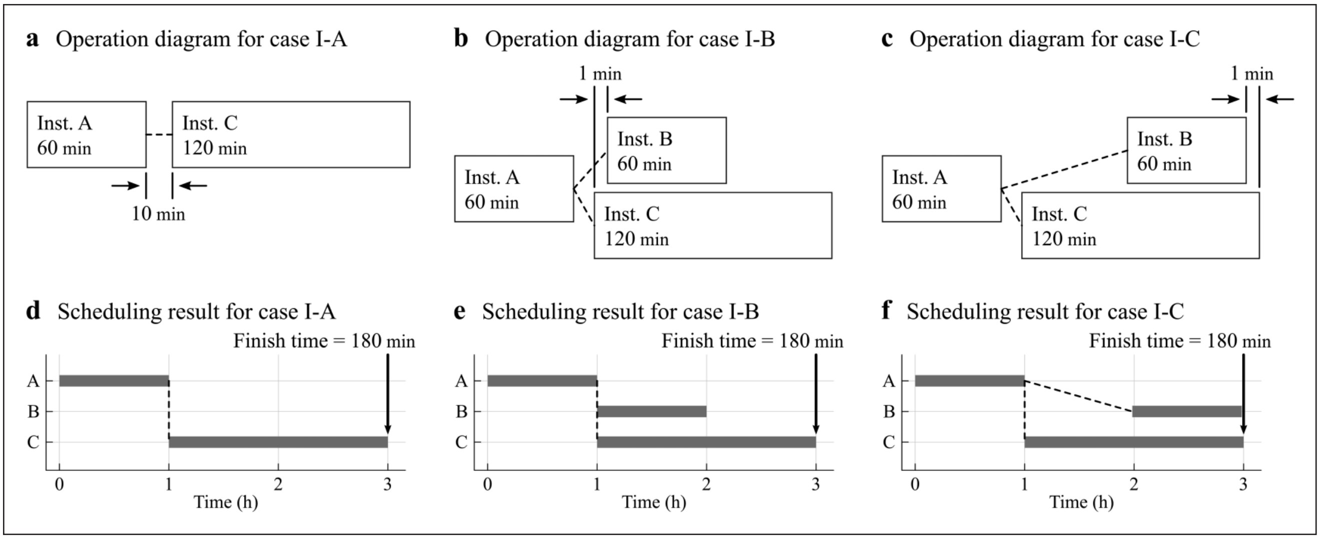

Figure 2. Scheduling with different types of time constraints: case I. (**a**) Schematic diagram of case I-A, in which there is a time constraint between the end time of one operation and the end time of another operation. (**b**) Schematic diagram of case I-B, in which there is a time constraint between the start time of an operation and the start time of another operation. Detailed descriptions of the job definitions can be found in **Supplemental Tables S1–S10**. (**c**) Schematic diagram of case I-C, in which there is a time constraint between the end time of an operation and the end time of another operation. (**d**) The scheduling result of **a**. (**e**) The scheduling result of **b**. (**f**) The scheduling result of **c**. Inst., instrument.

dependencies in which an experimental step depends on multiple preceding steps. For example, an instrument might produce two or more samples delivered to distinct instruments for succeeding experimental steps. Optimizing the schedules for these kinds of protocols, which contain branching and converging dependencies, is more difficult than scheduling serial protocols without these dependencies.[38]

We therefore tested whether our scheduling method is applicable to such protocols. We formulated a protocol based on that used by Gu et al.,[26] which is for a high-throughput screening involving coordination of multiple instruments, as an S-LAB problem (**Fig. 3a**, **Suppl. Tables S11–S14**). This job includes branching dependencies at three operations as well as converging dependencies at three operations. We set a 10 min TCMB for each consecutive pair of operations. **Figure 3b** shows the scheduling result for this job, demonstrating that our scheduling method found an optimal schedule. The job execution time was 101.0 min. The computation took 1.6 s.

Simulation-Based Job Design

Modern biological laboratories are often equipped with integrated workstations capable of processing multiple kinds of experimental steps as well as specialized instruments for particular experimental steps. In this kind of laboratory, there might be multiple options for selecting which type of instrument to use to process each operation. For example, although a workstation may be able to process several experimental steps as a single operation, it may alternatively be possible to use multiple specialized instruments to process the same steps as multiple separate operations. This kind of variation in job design may affect the execution time of a job, and it may be possible to shorten the execution time of a procedure by altering the job design.

To address this possibility, we simulated the scheduling of a procedure in two different job designs—one using only a single automated workstation (case III-A) (**Fig. 4a**) and the other using multiple instruments (case III-B) (**Fig. 4b**)—and compared their execution times. For the simulation, we formulated a protocol for automating sample preparation for deep sequencing using the Nextera rapid capture custom enrichment kit (Illumina, Inc., San Diego, CA)[39,40] by the Freedom Evo workstation (Tecan Group, Ltd., Männedorf, Switzerland.) as two S-LAB problems (**Suppl. Tables S15–S22**). Given that the protocol includes experimental steps using enzymatic reactions, we set TCMBs as 10 min for those operations to ensure that samples were not left for a long time.

We simulated a situation in which three instances of the job were executed. **Figure 4c** shows the scheduling results for case III-A. It took 851 min to complete the three jobs. **Figure 4d** shows the scheduling results for case III-B, in which we scheduled the three jobs simultaneously, whereas **Figure 4e** shows the results of the same case, but the three jobs were scheduled sequentially. When simultaneously scheduled, the jobs took 576 min, or 68% of the time in case III-A. In contrast, when sequentially scheduled, the jobs took 756 min, or 89% of the time in case III-A. The computation times were 1.0, 1.4 × 10^2, and 1.1 s for cases III-A,

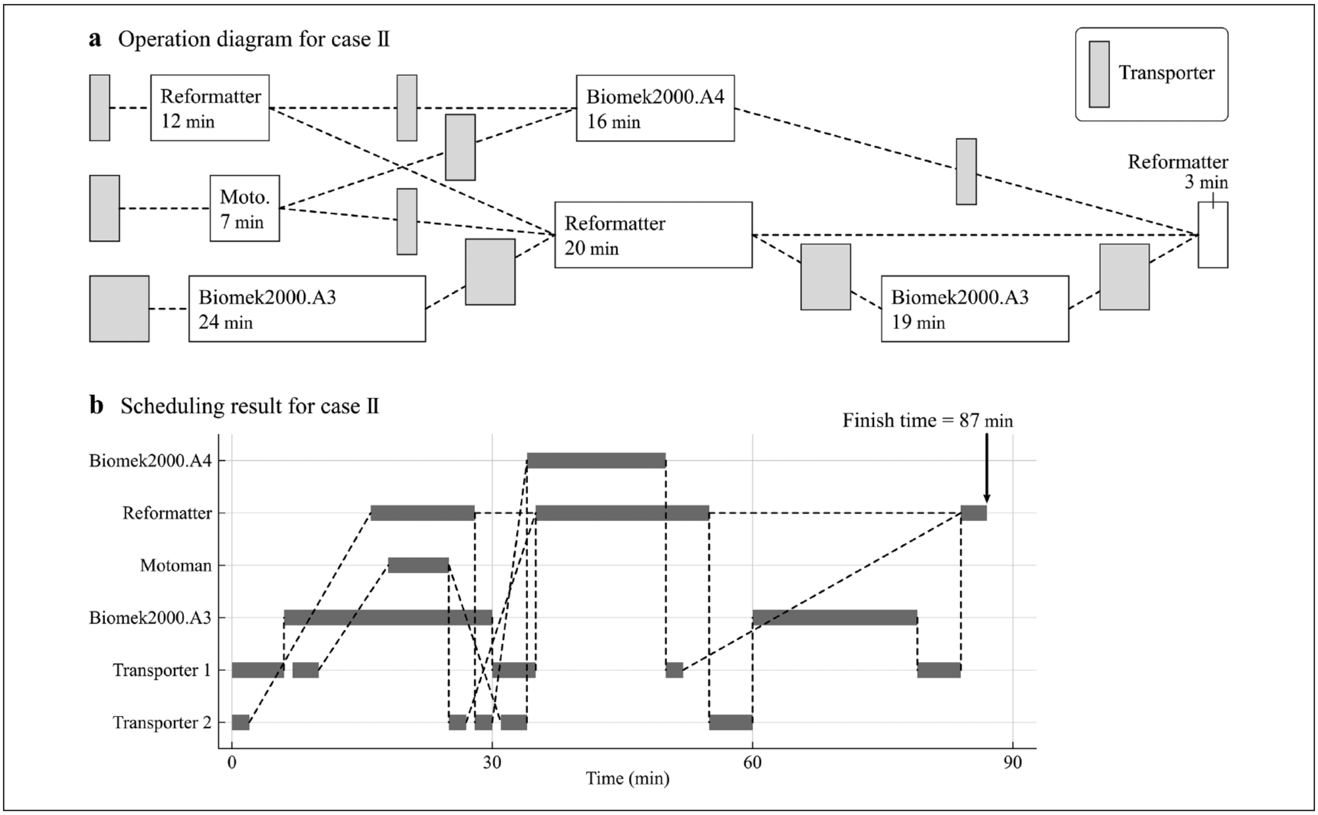

Figure 3. Scheduling of S-LAB problem with branched operation dependencies: case II. (**a**) Schematic diagram of the scheduling problem of case II. A detailed description can be found in **Supplemental Tables S11–S14**. A gray box represents a transportation operation processed by a transporter. (**b**) Scheduling result of case II. Moto., motoman.

III-B (simultaneous), and III-B (sequential), respectively. These results suggest that job designs affect the throughput, but simultaneous scheduling of multiple jobs is important to fully enjoy the benefit.

Simulation-Based Laboratory Configuration Design

The execution time of procedures can also be reduced by modifying the laboratory configuration, including the types and numbers of instruments. For example, it is reasonable to expect that jobs can be executed in less time if there are more instruments. However, the trade-off between the cost and benefits of adding new instruments needs to be evaluated in advance based on the expected throughput improvement of adding more instruments. Therefore, we next consider the use of scheduling simulations to evaluate the throughput improvement of different laboratory configurations.

To test this possibility, we scheduled the same jobs using different laboratory configurations by changing the number of instruments. We prepared a job definition composed of five operations, each of which is processed by different types of instruments A, B, C, D, and E (**Fig. 5a**). In the job definition, there is a bottleneck operation that takes two-thirds (60 min) of the entire execution time (**Suppl. Tables S23–S26**). We computed the schedule for three instances of this job in four different laboratory configurations, each of which has one, two, three, or four units of Instrument C to process the bottleneck operation. In case IV-A, which has only one unit of Instrument C, it took 122 min to finish three jobs (**Fig. 5b**). In case IV-B, which has two units of Instrument C, it took 91 min, or 75% of the time in case IV-A (**Fig. 5c**). When the number of Instrument C units was increased to three in case IV-C, the execution time further decreased to 82 min, or 67% of the time in case IV-A (**Fig. 5d**). However, the improvement gained by adding an instrument was smaller than the difference between cases IV-A and IV-B. There was no further reduction in execution time by the addition of more than three units of Instrument C (case IV-D, 82 min) (**Fig. 5e**). The scheduling method took 6.0, 5.4, 2.0, and 2.1 s for computation in cases IV-A, IV-B, IV-C, and IV-D, respectively. These results imply that one can improve the throughput of procedures by increasing the number of instruments for a bottleneck operation, but the effect of adding a single type of instrument reaches saturation at some point, possibly because other operations become bottlenecks.

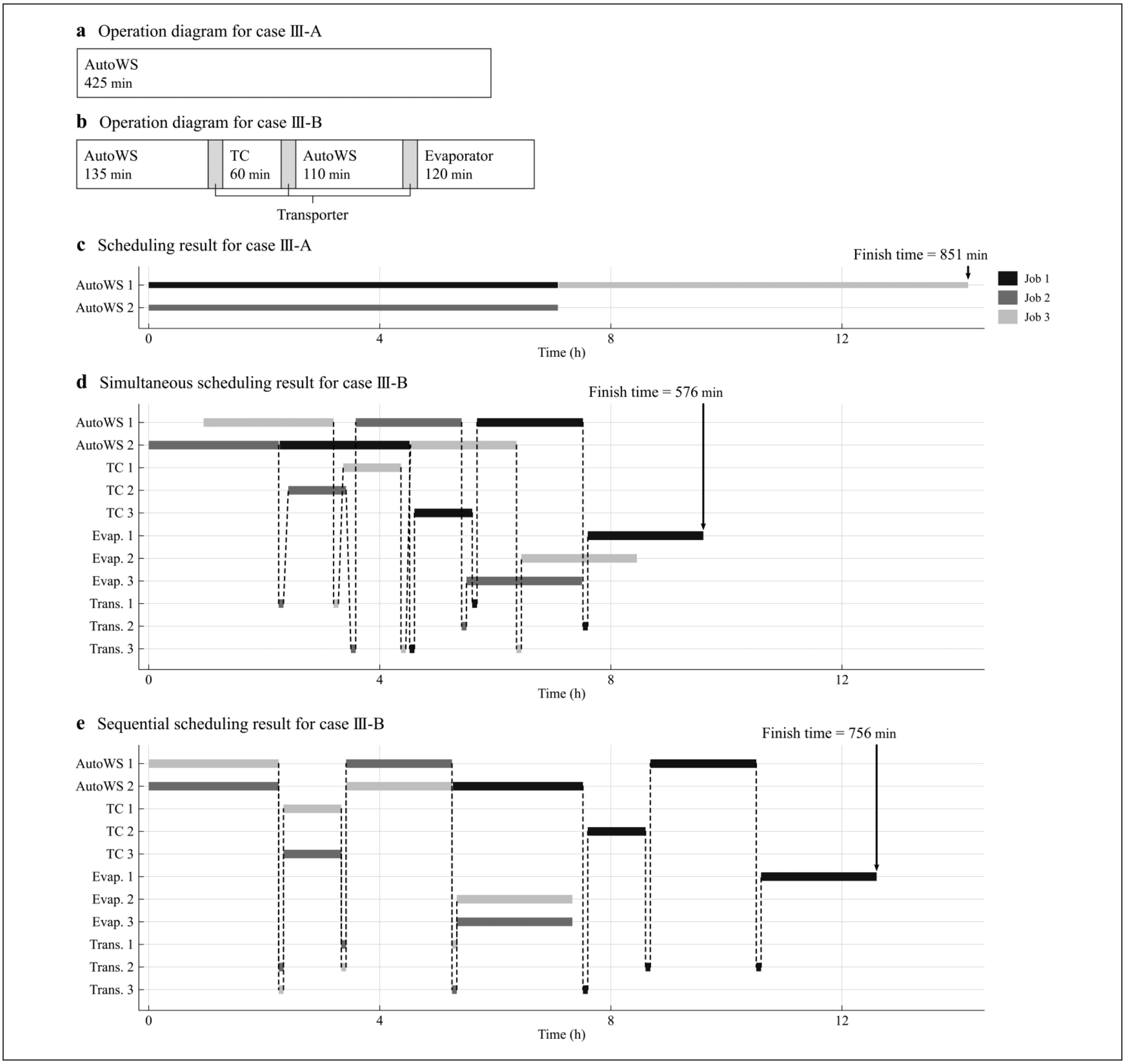

Figure 4. Simulation-based job design through scheduling: case III. (**a,b**) Schematic diagrams of the scheduling problems of case III-A (**a**) and case III-B (**b**). A detailed description can be found in **Supplemental Tables S15–S22**. (**c**) Scheduling result of case III-A, in which the three jobs are scheduled simultaneously. Instruments with unused instrument types are omitted from the *y* axis. (**d**) Scheduling result of case III-B, in which the three jobs are scheduled simultaneously. (**e**) Scheduling result of case III-B, in which the three jobs are scheduled sequentially. AutoWS, automated workstation; Evap., evaporator; TC, thermal cycler; Trans., transporter.

Discussion

In this study, we formulated the S-LAB problem as an instance of the MIP problem. S-LAB problems are scheduling problems for life science experiments using multiple types of instruments with TCMBs among operations. We proposed a scheduling method for determining optimal schedules for S-LAB problems based on the branch-and-bound algorithm (**Fig. 1**). Taking TCMBs into account is essential in life science experiments, particularly in those handling living cells or unstable biomolecules,[30–32] because simply optimizing schedules for throughput could lead to faster but poorer results. To evaluate the proposed method, we conducted scheduling simulations using both simple (**Fig. 2**) and complex (**Fig. 3**) job definitions, which have the typical characteristics of real-life science experiments. Through these simulations, we demonstrated that our method

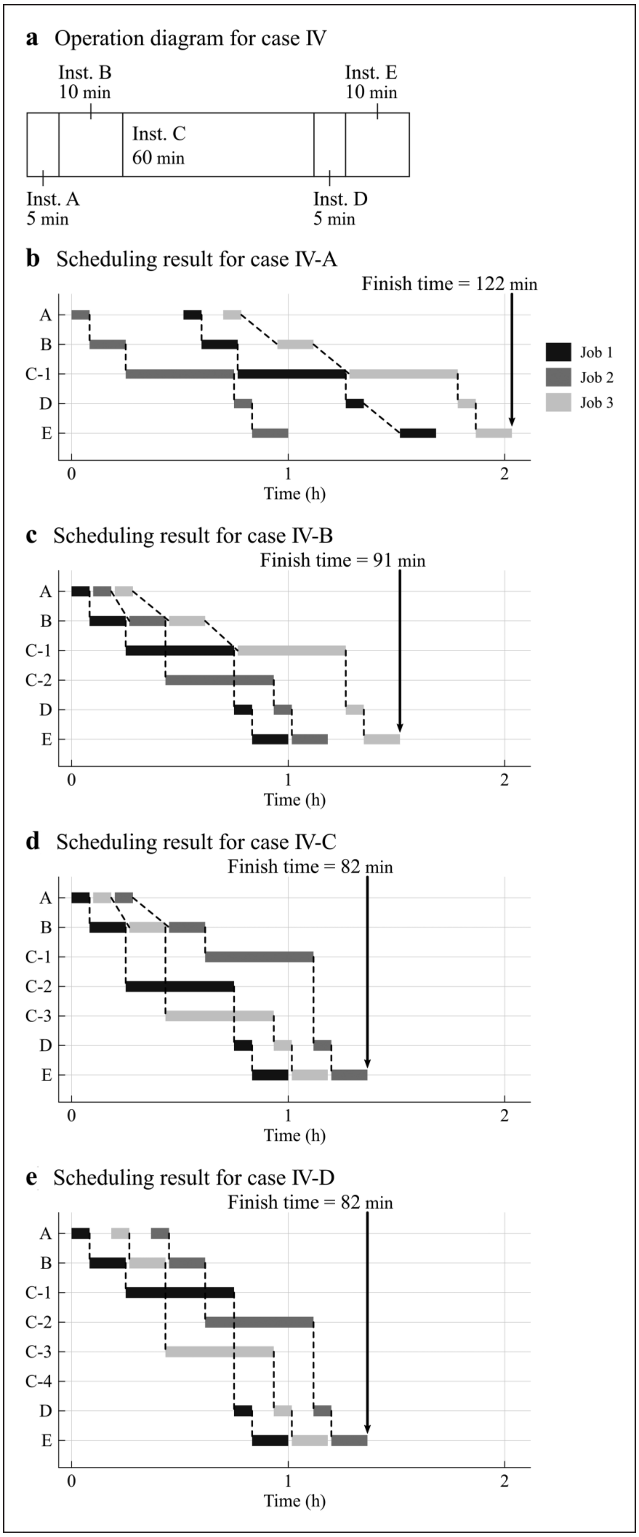

Figure 5. Laboratory configuration design by simulation: case IV. (**a**) Schematic diagram of case IV. A detailed description can be found in **Supplemental Tables S23–S26**. (**b–e**) Scheduling results for lab facilities equipped with different numbers of instrument C: (**b**) one, (**c**) two, (**d**) three, and (**e**) four. Inst., instrument.

was able to determine an optimal schedule that can carry out the entire procedure in the minimum execution time while satisfying TCMBs. The computational cost of the algorithm is also small enough for use in the real world.

We further applied this method to evaluate the effect on the throughput of different job designs of a procedure (**Fig. 4**) or increasing the number of instruments to parallel bottleneck operations (**Fig. 5**). Until now, studies on scheduling methods for laboratory automation have only investigated scheduling problems with fixed job definitions in a fixed laboratory configuration and have not discussed alternative designs for job definitions or laboratory configurations.[25,26,29] However, these kinds of design have been increasingly more important to improve the throughput of automated laboratories coordinating multiple types of instruments. For example, recent studies have discussed integrated laboratory automation systems in which various instruments are connected to each other by transporters.[26,27,29] Therefore, our proposed approach to use scheduling simulations for designing job definitions and laboratory configurations would become more essential in the future.

Previous studies have employed metaheuristic algorithms for scheduling,[25,26,29] partly due to the lower computational cost compared with exact algorithms. We showed that the branch-and-bound algorithm, an exact algorithm for MIP problems, was able to determine an optimal schedule for the S-LAB problems at a reasonably low computational cost. However, because we did not exhaustively evaluate various kinds of S-LAB problems, it might be possible that our method cannot determine an optimal solution for other S-LAB problems with a small computational time, which is particularly important for simulating multiple job designs and laboratory configurations. In such a case, the calculation might need to be stopped in the middle to obtain a tentative solution.

There are several possible directions for improving the proposed method. The first is support for more flexible time constraints. For example, in Autoprotocol, a descriptive language for experimental protocols, one can specify not only a lower and upper bound, but also a flexible cost function on the time difference between two operations.[41] Although allowing this kind of flexible constraint makes the optimization problem more difficult, it may decrease the number of cases in which S-LAB problems have no feasible solutions. The second is support for different optimization criteria. In this study, we used the throughput as the only optimization criterion for scheduling, but there can be different criteria, like quality of results or cost of experiments. In the future, it would be useful to incorporate such different multiple criteria in S-LAB scheduling. The third is dynamic scheduling. Instrument malfunctions and human errors can interrupt procedures or limit the availability of resources, forcing the reallocation of instruments to pending operations. A dynamic scheduling method has been proposed to deal with this kind of situation[42] and might be integrated with our method. Fourth, there is the problem of deciding which type of instrument to use to process an operation. In this study, the compatible instrument type for each operation was predetermined in the job definition. However, there are cases in which the same operation can be processed by different types of instruments. In the future, it would be useful to automatically determine which type of instrument to use for each operation by considering the availability of resources and the performance of each instrument type. Fifth, the method for evaluating the effect on the throughput of increasing the number of instruments can be improved. In this study, we evaluated the execution time of a fixed amount of jobs and showed that the effect saturates at some point (**Fig. 5**). However, it is important to evaluate the amount of jobs processed per unit time in the future because, when more instruments are available, more jobs can generally be executed by exploiting the increased resources. We note that this corresponds to the contrast between Amdahl's law[43] and Gustafson's law[44] in parallel computing: when evaluating the throughput improvement by multiple processors, the former uses the execution time of a fixed amount of workload and the latter uses the amount of workload processed per unit time.

By using the proposed scheduling method considering TCMBs, automated laboratories can improve the efficiencies of a wider range of life science experiments involving the use of multiple different types of instruments. Likewise, the idea of simulation-based evaluation using the scheduling method is expected to contribute to the design of job definitions and laboratory configurations in the future. In conclusion, this study could help the realization of laboratory automation in the life sciences by enabling sophisticated coordination of multiple different types of automation instruments.

Acknowledgments

We thank T. Mitsuyama and K. Kai for helpful discussions and G. N. Kanda for assistance in project administration.

Declaration of Conflicting Interests

The authors declared the following potential conflicts of interest with respect to the research, authorship, and/or publication of this article: H.U. is an employee of Tecan Japan Co., Ltd., which might benefit financially from increased scientific use of the Freedom EVO NGS workstation. T.D.I., T.H., K.T., and H.O. have filed for a patent on related technology.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the JST-Mirai Program (grant number JPMJMI18G4).

ORCID iDs

Takeshi D. Itoh https://orcid.org/0000-0002-4021-1481
Haruka Ozaki https://orcid.org/0000-0002-1606-2762

References

1. Weier, H. U.; Gray, J. W. A Programmable System to Perform the Polymerase Chain Reaction. *DNA* **1988**, *7*, 441–447.
2. Wang, Y.; Liu, Y.; Liu, J.; et al. MACBETH: Multiplex Automated *Corynebacterium glutamicum* Base Editing Method. *Metab. Eng*. **2018**, *47*, 200–210.
3. Hess, J. F.; Kohl, T. A.; Kotrová, M.; et al. Library Preparation for Next Generation Sequencing: A Review of Automation Strategies. *Biotechnol. Adv*. **2020**, *41*, 107537.
4. Nishimura, A.; Nakajima, R.; Takagi, R.; et al. Fabrication of Tissue-Engineered Cell Sheets by Automated Cell Culture Equipment. *J. Tissue Eng. Regen. Med*. **2019**, *13*, 2246–2255.
5. Matsumoto, E.; Koide, N.; Hanzawa, H.; et al. Fabricating Retinal Pigment Epithelial Cell Sheets Derived from Human Induced Pluripotent Stem Cells in an Automated Closed Culture System for Regenerative Medicine. *PLoS One* **2019**, *14*, e0212369.
6. dos Santos, F. F.; Andrade, P. Z.; da Silva, C. L.; et al. Bioreactor Design for Clinical-Grade Expansion of Stem Cells. *Biotechnol. J*. **2013**, *8*, 644–654.
7. Lopez, M. F.; Kristal, B. S.; Chernokalskaya, E.; et al. High-Throughput Profiling of the Mitochondrial Proteome Using Affinity Fractionation and Automation. *Appl. Theor. Electrophor*. **2000**, *21*, 3427–3440.
8. Jiang, X.; Feng, S.; Tian, R.; et al. Automation of Nanoflow Liquid Chromatography-Tandem Mass Spectrometry for Proteome Analysis by Using a Strong Cation Exchange Trap Column. *Proteomics* **2007**, *7*, 528–539.
9. Yoshimoto, N.; Kida, A.; Jie, X.; et al. An Automated System for High-Throughput Single Cell-Based Breeding. *Sci. Rep*. **2013**, *3*, **1191**.
10. Huang, D.; Ou, B.; Hampsch-Woodill, M.; et al. High-Throughput Assay of Oxygen Radical Absorbance Capacity (ORAC) Using a Multichannel Liquid Handling System Coupled with a Microplate Fluorescence Reader in 96-Well Format. *J. Agric. Food Chem*. **2002**, *50*, 4437–4444.
11. Burns, C. G.; Milan, D. J.; Grande, E. J.; et al. High-Throughput Assay for Small Molecules That Modulate Zebrafish Embryonic Heart Rate. *Nat. Chem. Biol*. **2005**, *1*, 263–264.
12. Supply, P.; Lesjean, S.; Savine, E.; et al. Automated High-Throughput Genotyping for Study of Global Epidemiology of *Mycobacterium tuberculosis* Based on Mycobacterial Interspersed Repetitive Units. *J. Clin. Microbiol*. **2001**, *39*, 3563–3571.
13. Konagaya, S.; Ando, T.; Yamauchi, T.; et al. Long-Term Maintenance of Human Induced Pluripotent Stem Cells by Automated Cell Culture System. *Sci. Rep*. **2015**, *5*, 16647.
14. Ochiai, K.; Motozawa, N.; Terada, M.; et al. A Variable Scheduling Maintenance Culture Platform for Mammalian Cells. *SLAS Technol*. **2020**, *26*, 209–217.
15. Yachie, N.; Robotic Biology Consortium; Natsume T. Robotic Crowd Biology with Maholo LabDroids. *Nat. Biotechnol*. **2017**, *35*, 310–312.

16. Kanda, G. N.; Tsuzuki, T.; Terada, M.; et al. Robotic Search for Optimal Cell Culture in Regenerative Medicine. *bioRxiv* **2020**. DOI: 10.1101/2020.11.25.392936.
17. Lehmann, R.; Severitt, J. C.; Roddelkopf, T.; et al. Biomek Cell Workstation: A Variable System for Automated Cell Cultivation. *J. Lab. Autom*. **2016**, *21*, 439–450.
18. Vorberg, E.; Fleischer, H.; Junginger, S.; et al. A Highly Flexible, Automated System Providing Reliable Sample Preparation in Element- and Structure-Specific Measurements. *J. Lab. Autom*. **2016**, *21*, 682–692.
19. Błażewicz, J.; Domschke, W.; Pesch, E. The Job Shop Scheduling Problem: Conventional and New Solution Techniques. *Eur. J. Oper. Res*. **1996**, *93*, 1–33.
20. Brucker, P. The Job-Shop Problem: Old and New Challenges. In 3rd Multidisciplinary International Conference on Scheduling: Theory and Applications, Paris, France, Aug 28–31, 2007; pp 15–22.
21. Schäfer, R. Concepts for Dynamic Scheduling in the Laboratory. *J. Assoc. Lab. Autom*. **2004**, *9*, 382–397.
22. Elliott, C.; Vijayakumar, V.; Zink, W.; et al. National Instruments LabVIEW: A Programming Environment for Laboratory Automation and Measurement. *J. Assoc. Lab. Autom*. **2007**, *12*, 17–24.
23. Shin, S. H.; Choi, B. J.; Ryew, S. M.; et al. Development of an Improved Scheduling Algorithm for Lab Test Operations on a Small-Size Bio Robot Platform. *J. Assoc. Lab. Autom*. **2010**, *15*, 15–24.
24. Delaney, N. F.; Rojas Echenique, J. I.; Marx, C. J. Clarity: An Open-Source Manager for Laboratory Automation. *J. Lab. Autom*. **2013**, *18*, 171–177.
25. Cabrera, C.; Fine-Morris, M.; Pokross, M.; et al. Dynamically Optimizing Experiment Schedules of a Laboratory Robot System with Simulated Annealing. *J. Lab. Autom*. **2014**, *19*, 517–527.
26. Gu, X.; Neubert, S.; Stoll, N.; et al. Intelligent Scheduling Method for Life Science Automation Systems. In 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Daegu, Korea, Nov 27–29, 2016; pp 156–161.
27. Burger, B.; Maffettone, P. M.; Gusev, V. V.; et al. A Mobile Robotic Chemist. *Nature* **2020**, *583*, 237–241.
28. Oliva, D.; Abd Elaziz, M.; Hinojosa, S. Metaheuristic Optimization. In *Metaheuristic Algorithms for Image Segmentation: Theory and Applications*; Oliva, D., Abd Elaziz, M., Hinojosa, S., Eds. Springer International Publishing: Cham, Switzerland, 2019; pp 13–26.
29. Neubert, S.; Gu, X.; Göde, B.; et al. Workflow Management System for the Integration of Mobile Robots in Future Labs of Life Sciences. *Chem. Ing. Tech*. **2019**, *91*, 294–304.
30. Gallego Romero, I.; Pai, A. A.; Tung, J.; et al. RNA-Seq: Impact of RNA Degradation on Transcript Quantification. *BMC Biol*. **2014**, *12*, 42.
31. Canelas, A. B.; ten Pierick, A.; Ras, C.; et al. Quantitative Evaluation of Intracellular Metabolite Extraction Techniques for Yeast Metabolomics. *Anal. Chem*. **2009**, *81*, 7379–7389.
32. Michl, J.; Park, K. C.; Swietach, P. Evidence-Based Guidelines for Controlling pH in Mammalian Live-Cell Culture Systems. *Commun. Biol*. **2019**, *2*, 144.
33. Dakin, R. J. A Tree-Search Algorithm for Mixed Integer Programming Problems. *Comput. J*. **1965**, *8*, 250–255.
34. Bezanson, J.; Edelman, A.; Karpinski, S.; et al. Julia: A Fresh Approach to Numerical Computing. *SIAM Rev*. 2017, 59, 65–98.
35. The Julia Project. The Julia Programming Language. https://julialang.org (accessed March 27, 2021).
36. Dunning, I.; Huchette, J.; Lubin, M. JuMP: A Modeling Language for Mathematical Optimization. *SIAM Rev*. **2017**, *59*, 295–320.
37. Forrest, J. J.; Vigerske, S.; Santos, H. G.; et al. Coin-or/Cbc. Version 2.10.5. https://github.com/coin-or/Cbc (accessed June 2, 2021).
38. Chen, C.-M.; King, C.-T.; Chen, Y.-Y. Branch Merging for Scheduling Concurrent Executions of Branch Operations. *IEE Proc. Comput. Dig. Tech*. **1996**, *143*, 369–375.
39. Illumina, Inc. *Nextera Rapid Capture Enrichment Reference Guide*. https://support.illumina.com/content/dam/illumina-support/documents/documentation/chemistry_documentation/samplepreps_nextera/nexterarapidcapture/nextera-rapid-capture-enrichment-guide-15037436-01.pdf (accessed June 2, 2021).
40. Tecan Group, Ltd. *User Guide for the Illumina Nextera Rapid Capture Library Preparation on the Freedom EVO NGS Workstation*. https://jp.illumina.com/content/dam/illumina-marketing/documents/products/appnotes/appnote-nrc-enrichment-automation.pdf (accessed June 2, 2021).
41. Bates, M.; Berliner, A. J.; Lachoff, J.; et al. Wet Lab Accelerator: A Web-Based Application Democratizing Laboratory Automation for Synthetic Biology. *ACS Synth. Biol*. **2017**, *6*, 167–171.
42. Zhang, C.; Yao, X.; Tan, W.; et al. Proactive Scheduling for Job-Shop Based on Abnormal Event Monitoring of Workpieces and Remaining Useful Life Prediction of Tools in Wisdom Manufacturing Workshop. *Sensors* **2019**, *19*, 5254.
43. Amdahl, G. M. Validity of the Single Processor Approach to Achieving Large Scale. In AFIPS Spring Joint Computer Conference, Atlantic City, NJ, April 18–20, 1967; pp 483–485.
44. Gustafson, J. L. Reevaluating Amdahl's Law. *Commun. ACM* **1988**, *31*, 532–533.