micromachines

MDPI

Article

SlimPort: Port-Driven High-Level Synthesis for Continuous-Flow Microfluidic Biochips

Youlin Pan [1,2,3], Yanbo Xu [1,2,3], Ziyang Chen [1,2,3], Xing Huang [4] and Genggeng Liu [1,2,3,*]

[1] College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China; panyoulin@163.com (Y.P.); pollystevens@163.com (Y.X.); czyeeric@126.com (Z.C.)
[2] Engineering Research Center of Big Data Intelligence, Ministry of Education, Fuzhou 350116, China
[3] Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou 350116, China
[4] School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China; xing.huang@nwpu.edu.cn
* Correspondence: liugenggeng@fzu.edu.cn

Abstract: Continuous-flow microfluidic biochips (CFMBs) automatically execute various bioassays by precisely controlling the transport of fluid samples, which is driven by pressure delivered through fluidic ports. High-level synthesis, as an important stage in the design flow of CFMBs, generates binding and scheduling solutions whose quality directly affects the efficiency of the execution of bioassays. Existing high-level synthesis methods perform numerous transport tasks concurrently to increase efficiency. However, fluidic ports cannot be shared between concurrently executing transport tasks, resulting in a large number of fluidic ports introduced by existing methods. Increasing the number of fluidic ports undermines the integration, reduces the reliability, and increases the manufacturing cost. In this paper, we propose a port-driven high-level synthesis method based on integer linear programming (ILP) called SlimPort, integrating the optimization of fluidic port number into high-level synthesis, which has never been considered in prior work. Meanwhile, to ensure bioassay correctness, volume management between devices with a non-fixed input/output ratio is realized. Additionally, two acceleration strategies for ILP, scheduling constraint reduction and upper boundary estimation of fluidic port number, are proposed to improve the efficiency of SlimPort. Experimental results from multiple benchmarks demonstrate that SlimPort leads to high assay execution efficiency and a low number of fluidic ports.

Keywords: microfluidic biochips; high-level synthesis; fluidic port; volume management

check for updates

Academic Editor: Sun Min Kim

Received: 13 April 2025
Revised: 5 May 2025
Accepted: 13 May 2025
Published: 14 May 2025

Citation: Pan, Y.; Xu, Y.; Chen, Z.; Huang, X.; Liu, G. SlimPort: Port-Driven High-Level Synthesis for Continuous-Flow Microfluidic Biochips. *Micromachines* **2025**, *16*, 577. https://doi.org/10.3390/mi16050577

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Continuous-flow microfluidic biochips (CFMBs), also known as lab-on-a-chip systems, have attracted considerable research interest in both academia and industry over the past decade, owing to their high precision, high throughput, and low cost [1,2]. CFMBs have been developed for use in various bioassays, such as point-of-care diagnosis [3], cancer diagnostic [4,5], immunoassays [6,7], and air quality monitoring [8].

A CFMB typically consists of two layers of polydimethylsiloxane (PDMS) material on a glass substrate [9], referred to as the control layer and the flow layer, respectively. Each of the two layers has its own channel network and is connected to the environment through its own ports, as shown in Figure 1a. Channels in the flow layer, also called flow channels, are connected to the external environment of the biochip through fluidic ports.

Fluidic ports are divided into flow ports and waste ports. Flow ports are used for the input of samples/reagents and the injection of air pressure from external pressure sources. Waste ports collect the output fluid and release the air pressure. Channels in the control layer, also called control channels, are connected to the external pressure source through control ports. The overlapping region between a flow channel and a control channel forms an elastic PDMS membrane. The structure is capable of controlling fluid transport in the corresponding flow channel according to the air pressure in the control channel; hence, it is called a valve. Figure 1b,c illustrate the cross-section of the valve in the open and closed states, respectively. To close the valve, the high pressure, generated by an external pressure source injected through the control port, is conducted to the valve via the control channel. This causes the valve's elastic membrane to be pushed downwards, sealing the corresponding flow channel, as shown in Figure 1c. When the high pressure is released, the valve's elastic membrane returns to its original state and the flow channel reopens for fluid transfer.

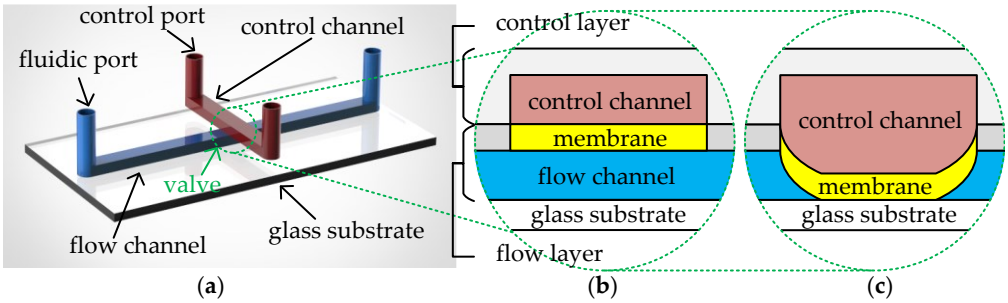


Figure 1. (**a**) Structure of a CFMB. Cross-section of (**b**) an open valve and (**c**) a closed valve.

Complex microfluidic devices can be constructed with various combinations of valves, flow channels, and functional components (e.g., peristaltic pump, heating component, etc.) to perform a variety of biochemical operations [10]. For example, a four-segment rotary mixer, as shown in Figure 2a, consists of valves (a_1–a_{12}), ring channels, and a peristaltic pump (a_{13}–a_{15}), which has been widely used on CFMBs [11]. By orderly transport of samples/reagents between different devices, CFMBs are able to accomplish complex bioassays automatically. The transport of samples/reagents is driven by the pressure injected through the flow port, so flow channels are employed to construct a complete pressure propagation path, called a flow path. A flow path starts at a flow port, passes through the fluid start location and the fluid target location, and ends at a waste port. For example, the flow path for the transport task shown in Figure 2a is flow port $f \rightarrow a_2 \rightarrow a_3 \rightarrow a_4 \rightarrow a_5 \rightarrow a_{19} \rightarrow a_{16} \rightarrow$ waste port w. The fluid start location is the flow port, and the fluid target location is the channel between a_3 and a_4.

The feature size of the CFMB has been continuously reduced with the advancement of microfluidic manufacturing technology [12–14]. This allows CFMBs to perform more complex bioassays. Fluidigm's 96.96 Dynamic Array, for instance, can run 9216 parallel polymerase chain reactions [15]. However, complex bioassays also mean that a large number of transport tasks need to be performed in parallel. To avoid pressure mutual interference and sample/reagent cross-contamination, separate flow paths need to be constructed for these parallel transport tasks. Each flow path requires a pair of fluidic ports, i.e., a flow port and a waste port. This leads to a rapid increase in fluidic ports. Previous work [16] has discussed the damage to the CFMB caused by integrating too many control ports. Analogous to the control ports, the fluidic ports are actually holes punched on the CFMB [17]. Too many holes punched on the CFMB not only take up chip area but also weaken the structural strength of the CFMB, making it more susceptible to physical failure [16]. In addition, fluidic ports also need to be connected to peripheral

equipments (e.g., air pressure sources, etc.) for sample/reagent input and collection and to provide power for transport tasks. These peripheral pieces of equipment are still made for mechanical devices and are large in size [16]. These limit the potential of large-scale integration of CFMBs.

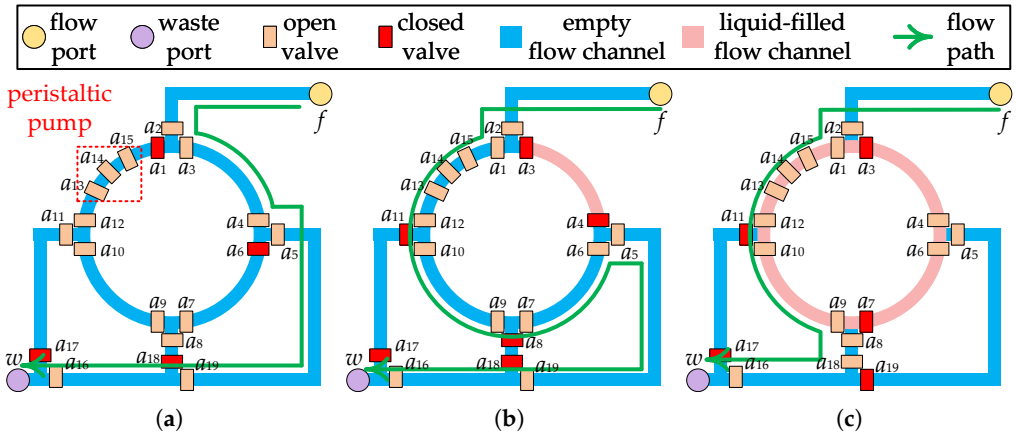


Figure 2. Part of the layout of the CFMB with a 4-segment rotary mixer. (**a**) Input fluid at 1/4 mixer capacity. (**b**) Input fluid at 3/4 mixer capacity. (**c**) Output fluid at 1/2 mixer capacity after mixing.

Over the past decade, considerable effort has been devoted to the design automation of CFMBs due to the high complexity of the CFMB architecture and bioassay protocol [18–23]. The architectural synthesis of CFMBs is usually divided into three major stages, high-level synthesis, physical design of the flow layer, and physical design of the control layer. High-level synthesis is an important step in the architectural synthesis of CFMBs. The goal of the high-level synthesis is to obtain binding and scheduling schemes that can be used as inputs for subsequent design steps, such as placement for devices, routing for flow path, etc. It is worth noting that the final chip layout is not determined during the high-level synthesis of biochips. As such, the binding and scheduling scheme generated in the high-level synthesis phase functions merely as an initial scheme that may be adapted in subsequent phases, but its quality directly affects the optimization of the subsequent phases. For example, the binding and scheduling scheme determines the devices to be placed and the connection relationships between them. A number of methods for high-level synthesis of CFMBs have been proposed. In [24], a list scheduling-based heuristic method is proposed to reduce the total application completion time. In [25], a graph-based approach, which formulates the high-level synthesis as a maximum clique finding problem, is proposed to accurately obtain an optimal binding and scheduling scheme. In [10], a component-oriented high-level synthesis method is proposed to improve the utilization of chip resources and to accommodate multiple types of microfluidic devices. In [26], a path-driven synthesis methodology is presented to integrate the actual fluid manipulations into both high-level synthesis and physical design. Volume management between devices and the removal of excess/waste fluids are introduced into the high-level synthesis to ensure the correctness of assay outcomes. In [16], a synthesis flow called MiniControl is proposed to generate chip architectures under strict constraints of control ports, where control-port minimization is considered systematically during the complete flow-layer design, including high-level synthesis. In [27], a high-level synthesis method considering fluid volume and channel storage is proposed to reduce the requirement of cache, thus ensuring the reliability of fluid caching under the distributed channel-storage architecture. In [28], the time constraints are introduced into the high-level synthesis to satisfy the real-time requirements of timing-sensitive bioassays. However, these methods do not take the fluidic port into account and perform numerous transportation tasks concurrently to

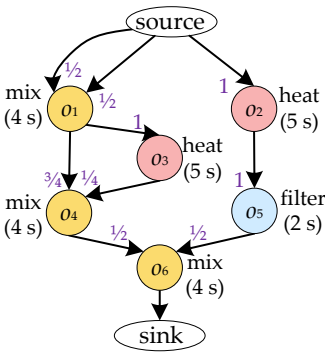
increase efficiency. This results in a large demand for fluidic ports. The enormous potential for reducing the number of fluidic ports in high-level synthesis is neglected.

Due to the limitations of existing methods and the importance of reducing the number of ports, in this paper, we propose SlimPort, a high-level synthesis method for CFMBs. The major contributions are listed below.

- A reduction in the fluidic port number is incorporated into high-level synthesis for the first time, thereby reducing the fabrication cost and improving the reliability of CFMBs.
- We propose extended volume management to achieve volume constraints for devices with a non-fixed input/output ratio, ensuring the correctness of bioassay outcomes.
- We propose two acceleration strategies for integer linear programming (ILP), scheduling constraint reduction and upper boundary estimation of the port number, to reduce the complexity of the ILP model and speed up the time required to solve it.
- The effectiveness of SlimPort is demonstrated by experimental results on five real-world bioassays and five synthetic benchmarks.

2. High-Level Synthesis, Motivation, and Problem Formulation

High-level synthesis inputs include the bioassay protocol to be implemented and a device library to automatically generate an optimized binding and scheduling scheme for a given bioassay. As shown in Figure 3a, the bioassay protocol is modelled as a directed graph $G(O, E)$, also known as a sequencing graph. Each node $o_i \in O$ represents a biochemical operation, such as heating and mixing, and is associated with a weight indicating its duration. Each edge $e_{i,j} \in E$ specifies the dependency between operations—i.e., operation o_i is a parent node of o_j in G. For the sake of simplicity, hereinafter we denote the device bound to operation o_i by $d(o_i)$. We use $f_{i,j}$ to represent the fluid from $d(o_i)$ that needs to be transported to $d(o_j)$. An edge $e_{i,j} \in E$ specifies the dependency between operations—i.e., operation o_i is a parent node of o_j in G. Each edge $e_{i,j}$ is associated with weights $r^{\text{In}}_{i,j}$, which denotes the ratio of the volume $f(i,j)$ to the capacity $d(o_j)$ after inputting $f(i,j)$ into $d(o_j)$. A device library D used for realizing the execution of operations in G. Each device $d_i \in D$ is associated with its area and capacity, as shown in Figure 3b. The device library also provides the input/output ratios supported by each device. Given the sequencing graph of a bioassay and the corresponding device library, the high-level synthesis of CFMBs is usually divided into two major tasks: binding and scheduling.



(**a**)

available device	area (mm^2)	capacity (nL)	ratio supported
d_1: mixer$_1$	20×20	400	1:1,1:3
d_2: mixer$_2$	25×25	800	1:1,1:3
d_3: heater$_1$	10×10	100	1
d_4: heater$_2$	20×10	200	1
d_5: filter$_1$	10×10	100	1
d_6: filter$_2$	20×10	200	1

(**b**)

Figure 3. (**a**) Sequencing graph. (**b**) Device library.

2.1. Volume Management Between Devices with Non-Fixed Input/Output Ratio and Binding

The goal of binding is to select a specific device for the execution of each operation and to satisfy the volume constraint between devices [26]. The reaction chamber of the microfluidic device consists of one or more flow channel segments that are divided by valves [10].

For example, the reaction chamber of a four-segment rotary mixer consists of a ring flow channel divided into four flow channel segments by valves ($a_1, a_3, a_4, a_6, a_7, a_9, a_{10}, a_{12}$), as shown in Figure 2a. The flow channel section of the device is filled with air rather than a vacuum in the initial state. All air should be removed from the device before the operation is carried out so that it does not affect the operation. At the start of the operation, it is also important to avoid leaving the air that drives the samples/reagents in the device. To prevent operations from failing, a minimum volume should be set for each input of a device to allow air to be expelled from the device while preventing outside air from entering.

Prior work has set a fixed minimum input volume limit for each device [26]. However, such setups cannot be applied on devices with non-fixed input/output ratios. Each device supports one or more input/output ratios, depending on the device architecture. For example, the four-segment rotary mixer can mix fluid in 1:1, 1:3, 1:1:2, or 1:1:1:1. As shown in Figure 2a,b, using the flow path as indicated by the green arrows, the mixer was successively input fluid with one-quarter and three-quarter mixer capacity, respectively. After performing a 1:3 ratio mixing operation, the fluid with one-half capacity of mixer can be removed using the flow path as shown by the green arrow in Figure 2c. It is possible to perform a 1:1 ratio mixing operation in the same mixer following the input of a fluid with one-half mixer capacity using the same flow path.

Due to the structural limitations of the device, the device cannot output/input fluid at any ratio. For example, the four-segment rotary mixer shown in Figure 2c is unable to remove all of the fluid from the ring channel with a single output procedure. Therefore, to ensure that the fluid volume input to the device satisfies the requirements of the operation o_i, o_i should be bound to a device that supports the required input ratio. Moreover, the fluid volume output by o_i should be determined based on the output ratio supported by $d(o_i)$ to satisfy the volume constraint between devices. Consider the bioassay described in Figure 3a as an example. When o_1 and o_4 are bound to d_1, if o_i is output at 1:1, the volume of $f_{1,4}$ is 200 nL, which is less than the volume of fluid required for o_4 (300 nL), violating the volume constraint between devices. In contrast, if o_1 is output at 1:3, $f_{1,4}$ has a volume of 300 nL and $f_{1,3}$ has a volume of 100 nL, satisfying the volume constraint.

2.2. Optimization of Fluidic Port Number and Scheduling

The goal of scheduling is to determine the start and end time for each operation and each transport task and to minimize bioassay completion time while satisfying given dependencies. Transport tasks in CFMBs can be divided into three categories: fluid transportation, excess fluid removal, and waste fluid removal [26]. The fluid transportation is the task of transporting samples/reagents between devices or between a port and a device. The excess fluid is cached in both ends of the device when the volume of input fluid is greater than the minimum input volume limit. To prevent excess fluid from interfering with other fluids in the CFMB, excess fluid removal should be performed before the next input to that device. Additionally, when the parent–child operations o_i and o_j are bound to the same device, if there are other inputs to the child operation o_j and the volume of $f_{i,j}$ in $d(o_j)$ is greater than the minimum input volume limit, then waste fluid removal should be performed to remove the fluid that is not needed anymore out of $d(o_j)$.

Figure 4a shows the scheduling results of the bioassay described in Figure 3a, where four devices are allocated to execute operations o_1–o_6, and the bioassay completed in 24 s with nine transport tasks. As mentioned before, each transport task needs to construct a flow path and associate it with a pair of flow ports. To avoid pressure mutual interference and sample/reagent cross-contamination, the same pair of flow ports cannot be shared between parallel transport tasks. In the scheduling scheme shown in Figure 4a, since $\#_3$ and $\#_1$, $\#_5$, and $\#_4$ are executed in parallel, at least two pairs of flow ports are required.

Correspondingly, Figure 4b shows a scheduling scheme considering the optimization of fluidic ports. By delaying the start of $\#_3$ and $\#_5$, there are no transport tasks that need to be performed in parallel throughout the bioassay. All transport tasks are allowed to share the same pair of fluidic ports—in other words, only two flow ports are required.

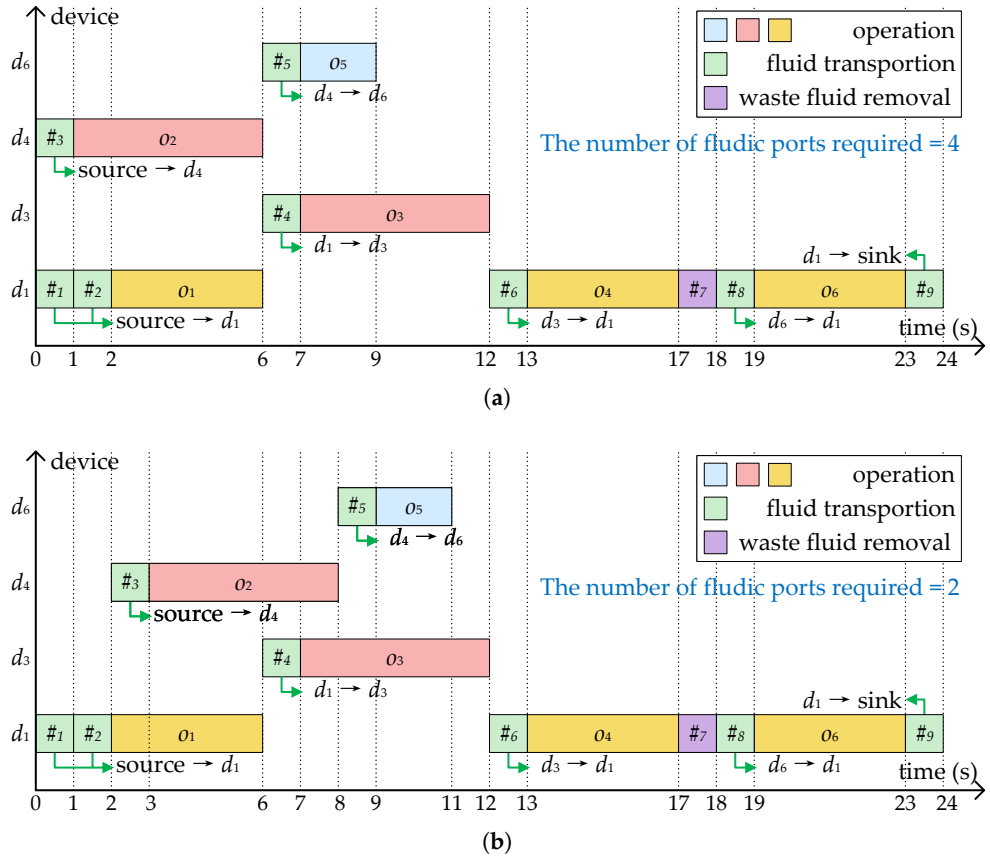


Figure 4. Two binding and scheduling schemes corresponding to Figure 3. (**a**) The scheme without considering the optimization of fluidic ports. (**b**) The scheme generated by SlimPort.

2.3. Problem Formulation

The high-level synthesis problem for CFMBs considered in this paper can be formulated as follows, based on the above analysis:

Inputs:

- A bioassay modeled as a sequencing graph $G(O, E)$ with the type and duration of each operation, as well as the volume ratio of each input.
- A device library D with the area and capacity of each device and the input/output ratios supported by each device.

Outputs:

- A binding scheme satisfying the volume constraints between devices.
- The volume of fluid output from each operations.
- A scheduling scheme indicating the start and end time for each operation and each transport task.

Objectives: Minimizing the following indicators.

- The completion time of the bioassay.
- The number of fluidic ports required.
- The total area of devices employed.
- The volume of excess fluid and waste fluid.

3. Details of the Proposed Port-Driven High-Level Synthesis

In this section, we discuss the proposed SlimPort in detail. In the following, the high-level synthesis is formulated and solved as an integer linear programming (ILP) model to obtain an optimal binding and scheduling scheme. Moreover, we present the two acceleration strategies, constraint reduction and boundary estimation, both of which reduce the computational overhead of our approach. Table 1 lists symbols that are frequently used in SlimPort.

Table 1. Symbols frequently used in SlimPort.

- $T(o_j)$: The execution time of o_j.
- b_{o_i,d_k}: A binary variable representing whether operation o_i is bound to a device d_k.
- ζ_{o_i,d_k}: A binary variable representing whether d_k can support the functions and input ratios required by o_i.
- $f_{i,j}$: The fluid from $d(o_i)$ that needs to be transported to $d(o_j)$.
- $d(o_i)$: The device bound to operation o_i.
- $\mathcal{Q}$: All output modes supported by devices in the device library D.
- Q_l: The l-th output modes supported by devices in the device library D.
- $q_{l,h}$: The h-th ratio between the output fluid volume and the device capacity when the device outputs in one output mode P_l.
- $\theta_{i,l}$: A binary variable representing whether o_i outputs in the output mode P_l.
- Support(d_k, P_l): A binary variable representing whether d_k can support the output mode P_l.
- $\varpi_{i,j,l,h}$: A binary variable representing whether the ratio between the volume of $f_{i,j}$ and the capacity of $d(o_i)$ is $p_{l,h}$.
- $r^{\text{In}}_{i,j}$: The ratio of the volume $f(i,j)$ to the capacity $d(o_j)$ after $f(i,j)$ is input to $d(o_j)$.
- c_k: The capacity of device d_k.
- $v_{i,j}$: The volume of the excess fluid or the waste fluid generated by $f_{i,j}$.
- $\phi_{i,j,l}, l \in [1,5]$: (1) $\phi_{i,j,1}$ represents the fluid transportation from $d(o_i)$ to storage. (2) $\phi_{i,j,2}$ represents the fluid transportation to $d(o_j)$. (3) $\phi_{i,j,l}, l = 3,4$ represents the excess fluid removal after $\phi_{i,j,2}$. (4) $\phi_{i,j,5}$ represents the waste fluid removal of d_i when transporting $f_{i,j}$ to d_j.
- $t^{\text{s}}_{\phi_{i,j,l}}, t^{\text{e}}_{\phi_{i,j,l}}$: The start time/end time of $\phi_{i,j,l}$.
- $\eta_{i,j,l}$: A binary variable representing whether $\phi_{i,j,l}$ should be performed.
- $t^{\text{s}}_{o_i}, t^{\text{e}}_{o_i}$: The start time/end time of o_i.
- $\tau_{i,j}$: A binary variable representing whether $v_{i,j}$ is greater than 0.
- $z_{i,j,k}$: A binary variable representing whether o_i and o_j are bound to d_k.
- $\psi_{i,h,j}$: A binary variable representing the order of the inputs from o_i and o_h to o_j.
- M: A very large constant.
- $\xi_{i,h,j}$: A binary variable representing the order of the outputs from o_j to o_i and o_h.
- $\sigma_{i,j}$: A binary variable representing the order of o_i and o_j.
- $b_{\phi_{i,j,l},p_k}$: A binary variable representing whether the $\phi_{i,j,l}$ is bound to fluidic port pair p_k.
- $\rho_{\phi_{i,j,l},\phi_{i',j',l'}}$: A binary variable representing the order of $\phi_{i,j,l}$ and $\phi_{i',j',l'}$.
- T^{e}: The completion time of the bioassay.
- S_k : The area of d_k.
- use$_{d_k}$, use$_{p_k}$: binary variables representing whether p_k and d_k are allocated, respectively.

3.1. ILP Model Constructed by SlimPort

3.1.1. Binding with Extended Volume Management

To ensure that each operation performs its function, each operation should be bound to a device. Thus, we have the following constraint:

$$\sum_{d_k \in D} b_{o_i,d_k} = 1, \forall o_i \in O, \tag{1}$$

$$b_{o_i,d_k} \leq \zeta_{o_i,d_k}, \forall o_i \in O, \forall d_k \in D, \tag{2}$$

where b_{o_i,d_k} and ζ_{o_i,d_k} are binary variables representing whether operation o_i is bound to a device d_k and whether d_k can support the functions and input ratios required by o_i, respectively.

Each device supports one or more output ratios. When the device outputs at a specific ratio, we refer to the ratio between output fluid volume and device capacity as an output mode. We note all output modes supported by devices in the device library D as $\mathcal{Q}$. When the device outputs in one output mode $Q_l \in \mathcal{Q}$, we use $q_{l,h}$ to represent the h-th ratio between the output fluid volume and the device capacity. For example, the $\mathcal{Q}$ associated with the equipment library shown in Figure 3b is $\{1, \{\frac{1}{2}, \frac{1}{2}\}, \{\frac{1}{4}, \frac{3}{4}\}\}$. Furthermore, the output modes supported by device d_1 are $Q_2 = \{\frac{1}{2}, \frac{1}{2}\}$ and $Q_3 = \{\frac{1}{4}, \frac{3}{4}\}$. The operation o_i can be output in only one output mode, which can be formulated as follows:

$$\sum_{Q_l \in \mathcal{Q}} \theta_{i,l} = 1, \forall o_i \in O, \tag{3}$$

$$\theta_{i,l} \leq \sum_{d_k \in D} \text{Support}(d_k, Q_l) \times b_{o_i,d_k}, \forall Q_l \in \mathcal{Q}, \forall o_i \in O, \tag{4}$$

where $\theta_{i,l}$ and $\text{Support}(d_k, P_l)$ are binary variables representing whether o_i outputs in the output mode Q_l and whether d_k can support the output mode Q_l.

$f_{i,j}$ should be transported from $d(o_i)$ to $d(o_j)$ for every $e_{i,j} \in E$. To determine the volume of $f_{i,j}$, we use a binary variable $\varpi_{i,j,l,h}$ to represent whether the ratio between the volume of $f_{i,j}$ and the capacity of $d(o_i)$ is $q_{l,h}$. $\varpi_{i,j,l,h}$ can be further constrained as follows:

$$\sum_{q_{l,h} \in Q_l} \varpi_{i,j,l,h} = \theta_{i,l}, \forall Q_l \in \mathcal{Q}, \forall e_{i,j} \in E, \tag{5}$$

$$\sum_{Q_l \in \mathcal{Q}, q_{l,h} \in Q_l} \varpi_{i,j,l,h} = 1, \forall e_{i,j} \in E. \tag{6}$$

Then, the volume constraint between devices with a non-fixed input/output ratio can be formulated as follows:

$$\Big(\sum_{Q_l \in \mathcal{Q}, q_{l,h} \in Q_l} \varpi_{i,j,l,h} \times q_{l,h}\Big) \times \Big(\sum_{d_k \in D} b_{o_i,d_k} \times c_k\Big) - r_{i,j}^{\text{In}} \times \sum_{d_k \in D} b_{o_j,d_k} \times c_k = v_{i,j} \geq 0, \forall e_{i,j} \in E, \tag{7}$$

where c_k is the capacity of device d_k, $r_{i,j}^{\text{In}}$ is the ratio of the volume $f(i,j)$ to the capacity $d(o_j)$ after $f(i,j)$ is input to $d(o_j)$, and $v_{i,j}$ is the volume of the excess fluid or the waste fluid generated by $f_{i,j}$.

3.1.2. Scheduling with Optimization of Fluidic Ports

When transporting $f_{i,j}$ to $d(o_i)$, we can optionally cache $f_{i,j}$ in storage and later transport it to $d(o_i)$. We use $\phi_{i,j,1}$ and $\phi_{i,j,2}$ to represent the fluid transportation from $d(o_i)$ to the storage and the fluid transportation to $d(o_j)$. When $f_{i,j}$ is transported directly from $d(o_i)$ to $d(o_j)$, only the fluid transportation $\phi_{i,j,2}$ is performed. After completing the input of

$f_{i,j}$ to $d(o_j)$, if $v_{i,j} > 0$, the excess liquids cached at both ends of $d(o_i)$ should be removed. This results in excess fluid removal, denoted by $\phi_{i,j,l}, l = 3, 4$. When o_i and o_j are bound to the same device and $f_{i,j}$ does not need to be cached in storage, it is unnecessary to perform $\phi_{i,j,l}, l = 1, 2$. If $v_{i,j} > 0$ in this case, there is waste fluid in $d(o_j)$ that needs to be removed. This results in waste fluid removal, denoted by $\phi_{i,j,5}$. Thus, we have the following constraint:

$$t^{\mathrm{e}}_{\phi_{i,j,l}} - t^{\mathrm{s}}_{\phi_{i,j,l}} = T_\phi \times \eta_{i,j,l}, \forall e_{i,j} \in E, l \in [1, 5], \tag{8}$$

where $t^{\mathrm{e}}_{\phi_{i,j,l}}$ and $t^{\mathrm{s}}_{\phi_{i,j,l}}$ are the end time and the start time of $\phi_{i,j,l}, l \in [1, 5]$, respectively. T_ϕ represents the duration of a transport task. $\eta_{i,j,l}, l \in [1, 5]$ is a binary variable representing whether $\phi_{i,j,l}$ should be performed. $\eta_{i,j,l}$ can be further constrained as follows:

$$\eta_{i,j,1} \leq \eta_{i,j,2}, \forall e_{i,j} \in E, \tag{9}$$

$$1 - \sum_{d_k \in D} z_{i,j,k} \leq \eta_{i,j,2} \leq \eta_{i,j,1} + 1 - \sum_{d_k \in D} z_{i,j,k}, \forall e_{i,j} \in E, \tag{10}$$

$$\tau_{i,j} \times \epsilon \leq v_{i,j} \leq \tau_{i,j} \times M, \forall e_{i,j} \in E, \tag{11}$$

$$2 \times \eta_{i,j,l} \leq \tau_{i,j} + \eta_{i,j,1} \leq 1 + \eta_{i,j,l}, \forall e_{i,j} \in E, l = 3, 4, \tag{12}$$

$$2 \times \eta_{i,j,5} \leq \tau_{i,j} + 1 - \eta_{i,j,1} \leq 1 + \eta_{i,j,5}, \forall e_{i,j} \in E, \tag{13}$$

where $\tau_{i,j}$ and $z_{i,j,k}$ are binary variables representing whether $v_{i,j}$ is greater than 0 and whether o_i and o_j are bound to d_k, respectively. $z_{i,j,k}$ can be further constrained as follows:

$$2 \times z_{i,j,k} \leq b_{o_i,d_k} + b_{o_i,d_k} \leq z_{i,j,k} + 1, \forall o_i, o_j \in O, \forall d_k \in D. \tag{14}$$

The fluid transportation $\phi_{i,j,2}$ should not be started until the fluid transportation $\phi_{i,j,1}$ has been completed. Moreover, the excess fluid removal $t^{\mathrm{s}}_{i,j,l}, l = 3, 4$ should be performed after transportation $\phi_{i,j,2}$ has finished. Then, the waste fluid removal $t^{\mathrm{s}}_{i,j,5}$ is performed. Thus, we have the following constraint:

$$t^{\mathrm{e}}_{\phi_{i,j,1}} \leq t^{\mathrm{s}}_{\phi_{i,j,2}}, \forall e_{i,j} \in E, \tag{15}$$

$$t^{\mathrm{e}}_{i,j,2} \leq t^{\mathrm{s}}_{i,j,l}, \forall e_{i,j} \in E, l = 3, 4. \tag{16}$$

The inputs of o_j should be loaded separately, which can be constrained as follows:

$$\begin{aligned} t^{\mathrm{e}}_{\phi_{i,j,l}} &\leq t^{\mathrm{s}}_{\phi_{h,j,2}} + (1 - \psi_{i,h,j}) \times M \\ t^{\mathrm{e}}_{\phi_{h,j,l}} &\leq t^{\mathrm{s}}_{\phi_{i,j,2}} + \psi_{i,h,j} \times M \end{aligned}, \forall e_{i,j}, e_{h,j} \in E, l = 3, 4, \tag{17}$$

where $\psi_{i,h,j}$ is a binary variable representing the order of the inputs from o_i and o_h to o_j and M is a very large constant for transforming two situations indicated by $\psi_{i,h,j}$ into linear constraints.

The operation can only be performed after all inputs have been completed. The operation should then last for the specified time to implement the corresponding functionality. Thus, we have the following constraint:

$$t^{\mathrm{e}}_{\phi_{i,j,2}} \leq t^{\mathrm{s}}_{o_j}, \forall e_{i,j} \in E, t^{\mathrm{e}}_{o_j} - t^{\mathrm{s}}_{o_j} = T(o_j), \forall o_j \in O, \tag{18}$$

where $t^{\mathrm{e}}_{o_j}$ and $t^{\mathrm{s}}_{o_j}$ are the end time and the start time of o_j, respectively, and $T(o_j)$ is the execution time of o_j.

Outputs of o_j should only be executed after the operation o_j has been completed and the outputs of o_j should be removed separately. Thus, we have the following constraint:

$$t^{\mathrm{e}}_{o_j} \leq t^{\mathrm{s}}_{\phi_{j,h,1}}, \forall e_{j,h} \in E, \tag{19}$$

$$t^{\mathrm{e}}_{\phi_{j,i,1}} \leq t^{\mathrm{s}}_{\phi_{j,h,1}} + (1 - \xi_{i,h,j}) \times M, \forall e_{j,i}, e_{j,h} \in E, \tag{20}$$

$$t^{\mathrm{e}}_{\phi_{j,i,2}} \leq t^{\mathrm{s}}_{\phi_{j,h,1}} + (1 - \xi_{i,h,j} + \eta_{j,i,1}) \times M, \forall e_{j,i}, e_{j,h} \in E, \tag{21}$$

$$t^{\mathrm{e}}_{\phi_{j,h,1}} \leq t^{\mathrm{s}}_{\phi_{j,i,1}} + \xi_{i,h,j} \times M, \forall e_{j,i}, e_{j,h} \in E, \tag{22}$$

$$t^{\mathrm{e}}_{\phi_{j,h,2}} \leq t^{\mathrm{s}}_{\phi_{j,i,1}} + (\xi_{i,h,j} + \eta_{j,h,1}) \times M, \forall e_{j,i}, e_{j,h} \in E, \tag{23}$$

where $\xi_{i,h,j}$ is a binary variable representing the order of the outputs from o_j to o_i and o_h.

Additionally, without loss of correctness, we make $t^{\mathrm{e}}_{i,j,5}$ equal to the time at which all fluids generated by o_i leave $d(o_i)$ via the following constraints:

$$t^{\mathrm{s}}_{\phi_{i,j,2}} \leq t^{\mathrm{s}}_{\phi_{i,j,1}} + \eta_{i,j,1} \times M, \forall e_{i,j} \in E, \tag{24}$$

$$t^{\mathrm{e}}_{i,j,1} \leq t^{\mathrm{s}}_{i,j,5}, \forall e_{i,j} \in E, \tag{25}$$

$$t^{\mathrm{e}}_{i,j,2} \leq t^{\mathrm{s}}_{i,j,5} + \eta_{i,j,1} \times M, \forall e_{i,j} \in E. \tag{26}$$

When two operations are bound to the same component, all fluids generated by one operation should be removed before the other operation can start executing the inputs, constrained as follows:

$$\begin{aligned} t^{\mathrm{e}}_{\phi_{i,j',5}} &\leq t^{\mathrm{s}}_{\phi_{i',j,2}} + (2 - \sum_{d_k \in D} z_{i,j,k} - \sigma_{i,j}) \times M, \forall e_{i,j'}, e_{i',j} \in E \\ t^{\mathrm{e}}_{\phi_{j,i',5}} &\leq t^{\mathrm{s}}_{\phi_{j',i,2}} + (1 - \sum_{d_k \in D} z_{i,j,k} + \sigma_{i,j}) \times M, \forall e_{j,i'}, e_{j',i} \in E \end{aligned}, \forall o_i, o_j \in O, \tag{27}$$

where $\sigma_{i,j}$ is a binary variable representing the order of o_i and o_j.

We assume that there is at most one storage on the CFMBs. Due to the bandwidth limitations of the storage, all cache inputs and cache outputs should be separated.

$$\begin{aligned} t^{\mathrm{e}}_{\phi_{i,j,l}} &\leq t^{\mathrm{s}}_{\phi_{i',j',l'}} + (3 - \eta_{i,j,1} - \eta_{i',j',1} - \rho_{\phi_{i,j,l},\phi_{i',j',l'}}) \times M \\ t^{\mathrm{e}}_{\phi_{i',j',l'}} &\leq t^{\mathrm{s}}_{\phi_{i,j,l}} + (2 - \eta_{i,j,1} - \eta_{i',j',1} + \rho_{\phi_{i,j,l},\phi_{i',j',l'}}) \times M \end{aligned}, \forall e_{i,j}, e_{i',j'} \in E, l = 1,2, l' = 1,2, \tag{28}$$

where $\rho_{\phi_{i,j,l},\phi_{i',j',l'}}$ is a binary variable representing the order of $\phi_{i,j,l}$ and $\phi_{i',j',l'}$.

As previously mentioned, each transport task should be associated with a pair of fluidic ports to provide pressure. We use a binary variable $b_{\phi_{i,j,l},p_k}$ to represent whether the $\phi_{i,j,l}$ is bound to fluidic port pair p_k. We have

$$\sum_{p_k \in P} b_{\phi_{i,j,l},p_k} = \eta_{i,j,l}, \forall e_{i,j} \in E, l \in [1,5], \tag{29}$$

where P is the set of fluidic port pairs.

Transport tasks bound to the same pair of fluid ports cannot be executed in parallel, constrained as follows:

$$\begin{aligned} t^{\mathrm{e}}_{\phi_{i,j,l}} &\leq t^{\mathrm{s}}_{\phi_{i',j',l'}} + (3 - b_{\phi_{i,j,l},p_k} - b_{\phi_{i',j',l'},p_k} - \rho_{\phi_{i,j,l},\phi_{i',j',l'}}) \times M \\ t^{\mathrm{e}}_{\phi_{i',j',l'}} &\leq t^{\mathrm{s}}_{\phi_{i,j,l}} + (2 - b_{\phi_{i,j,l},p_k} - b_{\phi_{i',j',l'},p_k} + \rho_{\phi_{i,j,l},\phi_{i',j',l'}}) \times M \end{aligned}, \forall e_{i,j}, e_{i',j'} \in E, l, l' \in [1,5], p_k \in P, \tag{30}$$

where $\rho_{\phi_{i,j,l},\phi_{i',j',l'}}$ is a binary variable representing the order of $\phi_{i,j,l}$ and $\phi_{i',j',l'}$.

3.1.3. Optimization Objective

Once all operations and transport tasks have been completed, the bioassay is complete, constrained as follows:

$$t^{\mathrm{e}}_{o_i} \leq T^{\mathrm{e}}, \forall o_i \in O \tag{31}$$

$$t^{\mathrm{e}}_{\phi_{i,j,l}} \leq T^{\mathrm{e}}, \forall e_{i,j} \in E, l \in [1,5], \tag{32}$$

where T^{e} is the completion time of the bioassay.

We use binary variables use_{p_k} and use_{d_k} to represent whether p_k and d_k are allocated, respectively, constrained as follows:

$$b_{\phi_{i,j,l},p_k} \leq \mathrm{use}_{p_k}, \forall e_{i,j} \in E, l \in [1,5], \forall p_k \in P, \tag{33}$$

$$\eta_{i,j,1} \leq \mathrm{use}_{d_{\mathrm{storage}}}, \forall e_{i,j} \in E, \tag{34}$$

$$b_{o_i,d_k} \leq \mathrm{use}_{d_k}, \forall o_i \in O, \forall d_k \in D, \tag{35}$$

where d_{storage} represents the storage in the device library.

Finally, an efficient high-level scheme can be generated by solving the following problem:

$$\text{minimize} \quad \alpha \times T^{\mathrm{e}} + \beta \times \sum_{p_k \in P} \mathrm{use}_{p_k} + \gamma \times \sum_{d_k \in D} (\mathrm{use}_{d_k} \times S_k) + \delta \times \sum_{e_{i,j} \in E} v_{i,j}, \tag{36}$$

$$\text{subject to} \quad (1)\text{–}(35), \tag{37}$$

where α, β, γ, and δ are three weighting factors and S_k is the area of d_k.

3.2. Acceleration Strategies

To ensure the efficiency of SlimPort, two acceleration strategies, scheduling constraint reduction and upper boundary estimation of the port number, are proposed in this paper. The complexity of an ILP model is related to the number of its constraints and variables. The main idea of our proposed acceleration strategies is to reduce the number of constraints and variables that need to be constructed using the a priori knowledge provided by the inputs to achieve the acceleration.

3.2.1. Scheduling Constraint Reduction

The constraint given in (27) restricts the order between two-by-two operations. For the sake of simplicity, we define the upstream operations of o_i as all operations on the path from the source to o_i. Similarly, the downstream operations of o_i are all operations on the path from o_j to the source. Based on the dependencies in the sequencing graph, it is known that the upstream operations of o_i should be completed before o_i is executed and the downstream operations of o_i should be executed after o_i is completed. For operations of known order, we remove them from Constraint (27), thus achieving the reduction of Constraint (27).

Constraints (28) and (30) restrict the order between two-by-two transport tasks. For the sake of simplicity, we define the upstream edges of $e_{i,j}$ as all edges on the path from the source to o_i. Similarly, the downstream edges of $e_{i,j}$ are all operations on the path from o_j to the source. Based on the dependencies in the sequencing graph, it is known that if $e_{i',j'}$ is the upstream edge of $e_{i,j}$, $\phi_{i',j',l'}, l' \in [1,5]$ should be completed before $\phi_{i,j,l}, l \in [1,5]$ is executed. And if $e_{i',j'}$ is the downstream edge of $e_{i,j}$, $\phi_{i',j',l'}, l' \in [1,5]$ should be executed after $\phi_{i,j,l}, l \in [1,5]$ is completed. For transport tasks of known order, we remove them from Constraints (28) and (30), thus achieving the reduction of Constraints (28) and (30).

3.2.2. Upper Boundary Estimation of Fluidic Port Number

The number of variables $b_{\phi_{i,j,l},p_k}$ and the number of Constraints (29) and (30) are related to the number of ports in P. To ensure the correctness while reducing the complexity of the ILP model, we estimate the upper bound on the number of fluid ports using Algorithm 1. For the sake of simplicity, we define $e_{i,j}$ and $e_{i',j'}$ such that they are compatible when (1) $e_{i',j'}$ is not the upstream/downstream edge of $e_{i,j}$ and (2) $i \neq i' \wedge j \neq j' \wedge i \neq j' \wedge j \neq i'$. Given the sequencing graph $G(O, E)$ and the component library D, we first construct an edge-compatible graph. A node of the edge-compatible graph is represented as $e_{i,j} \in E$. The edge between $e_{i,j}$ and $e_{i',j'}$ represents that $e_{i,j}$ and $e_{i',j'}$ are compatible. We then obtain all maximal cliques in the edge-compatible graph via the Bron–Kerbosch algorithm [29]. Finally, we traverse all maximal clusters and obtain the upper bound on the number of fluid ports.

Algorithm 1: Upper Boundary Estimation of Fluidic Port Number

Input: The sequencing graph $G(O, E)$ and the device library D
Output: The upper boundary of fluidic port number N_{port}^{upper}

for *each* $e_{i,j} \in E$ **do**
  $\Psi_{i,j} = \emptyset$;
  Add all $e_{i,h} \in E$ into $\Psi_{i,j}$;
  Add all $e_{h,j} \in E$ into $\Psi_{i,j}$;
  Initialize a stack $\mathbb{S}$ and push all $e_{j,h} \in E$ into $\mathbb{S}$;
  while $\mathbb{S} \neq \emptyset$ **do**
    $e_{i',j'} = \mathbb{S}.\text{pop}()$;
    Push all $e_{j',h'} \in E$ into $\mathbb{S}$;
    Add all $e_{j',h'} \in E$ into $\Psi_{i,j}$;
  end while
end for
Construct an edge-compatible graph with each node representing any edge in E, and if $e_{i,j} \notin \Psi_{i',j'} \wedge e_{i',j'} \notin \Psi_{i,j}$, then connect the nodes corresponding to these $e_{i,j}$ and $e_{i',j'}$ with a single edge;
Get all maximal cliques $\mathbb{C}$ in the edge-compatible graph;
for *each* $_i \in \mathbb{C}$ **do**
  Get the parent operations corresponding to the edge nodes in $_i$ and count the number of operations of each type, denoted as $n_1^s, n_2^s, ..., n_{N_{type}}^s$, where N_{type} is the type number of all device in D;
  Get the child operations corresponding to the edge nodes in $_i$ and count the number of operations of each type, denoted as $n_1^e, n_2^e, ..., n_{N_{type}}^e$;
  $N_i = \min\{\sum_{i=0}^{N_{type}} \min\{n_i^s, n_i^d\}, \sum_{i=0}^{N_{type}} \min\{n_i^e, n_i^d\}\}$, where n_i^d is the number of devices with type i;
end for
$N_{port}^{upper} = 2 * max_{_i \in \mathbb{C}} N_i$

4. Experimental Results

The proposed SlimPort was implemented in Python 3.11.11 and tested on a PC with 2.50 GHz CPU and 8 GB memory. The solver Gurobi was employed to solve the ILP model proposed in Section 3. As shown in Table 2, there are ten benchmarks to verify the performance of SlimPort. Five are synthetic benchmarks and the rest are real-world biochemical applications [26]. Parameters $|O|$,$|E|$, and $|device|$ in Table 2 are the operation

number and the edge number of the sequencing graph and the number of corresponding devices provided, respectively. The experimental parameters are set as follows: $\alpha = 0.6$, $\beta = 0.1$, $\gamma = 0.002$, and $\delta = 0.001$.

Table 2. Details of benchmarks used in experiments.

Benchmarks $\lvert O\rvert/(\lvert$**mixer**$\rvert, \lvert$**heater**$\rvert, \lvert$**filter**$\rvert, \lvert$**separator**$\rvert, \lvert$**detector**$\rvert, \lvert$**storage**$\rvert)/\lvert E\rvert$				
PCR	IVD	ProteinSplit	Kinase act-1	Kinase act-2
7/(4, 0, 0, 0, 0, 1)/15	12/(4, 0, 0, 0, 4, 1)/24	14/(4, 0, 0, 3, 3, 1)/27	4/(4, 0, 0, 4, 0, 1)/16	12/(4, 0, 0, 4, 0, 1)/48
Synthetic1	Synthetic2	Synthetic3	Synthetic4	Synthetic5
10/(4, 2, 3, 0, 2, 1)/15	15/(4, 3, 2, 0, 3, 1)/21	20/(4, 4, 3, 4, 2, 1)/28	25/(4, 4, 3, 0, 2, 1)/33	30/(4, 3, 3, 2, 4, 1)/42

4.1. Validation of the Proposed SlimPort

Since there is no existing work considering the volume management between devices with a non-fixed input/output ratio, we implemented another method, called PD, by modifying the high-level synthesis method in [26], to verify the performance of the proposed SlimPort.

We ran both algorithms on the aforementioned benchmarks. Table 3 shows comparison results between SlimPort and PD, where columns T^{e}, V_{r}, and N_{p} are the completion time of the bioassay, the volume of excess/waste fluid, and the number of fluidic ports required, respectively. Moreover, the column "Imp (%)" provides the relative improvements of SlimPort over PD.

Table 3. Comparison results between SlimPort and PD in terms of the completion time of the bioassay, the volume of excess/waste fluid, and the number of fluidic ports required.

Benchmarks	T^{e} **(s)**			V_{r} **(nL)**			N_{p}		
	PD	**Ours**	**Imp (%)**	**PD**	**Ours**	**Imp (%)**	**PD**	**Ours**	**Imp (%)**
IVD	29	29	0.00	1200	1000	16.67	12	4	66.67
PCR	24	24	0.00	1000	800	20.00	6	2	66.67
Kinase act-1	35	35	0.00	600	600	0.00	4	2	50.00
Kinase act-2	52	52	0.00	1300	900	30.77	12	4	66.67
ProteinSplit	88	88	0.00	2300	2300	0.00	8	4	50.00
Synthetic1	33	33	0.00	1200	1200	0.00	6	4	33.33
Synthetic2	39	39	0.00	1600	1100	31.25	6	4	33.33
Synthetic3	53	53	0.00	200	200	0.00	4	2	50.00
Synthetic4	54	54	0.00	1500	1100	26.67	12	4	66.67
Synthetic5	67	67	0.00	400	400	0.00	6	4	33.33
Average	-		0.00	-		12.54	-		51.67

As shown in Table 3, SlimPort achieves a 33.3–66.7% reduction in terms of the number of fluidic ports required, with an average reduction of 51.67%, while maintaining the optimal solution in terms of bioassay completion time. Moreover, the volume of excess/waste fluid is reduced by 12.54%, thus reducing reagent consumption. As shown in Figures 5 and 6, SlimPort maintains the optimal solution in terms of the area of devices allocated and improves the number of transport tasks by 2.72% on average compared with PD. PD takes the bioassay completion time and the area of devices allocated as the optimization objective and achieves the optimal solution. SlimPort is able to optimize the number of fluidic ports required, the volume of excess/waste fluid, and the number of transport tasks and maintain the optimal bioassay completion time and the area of devices allocated.

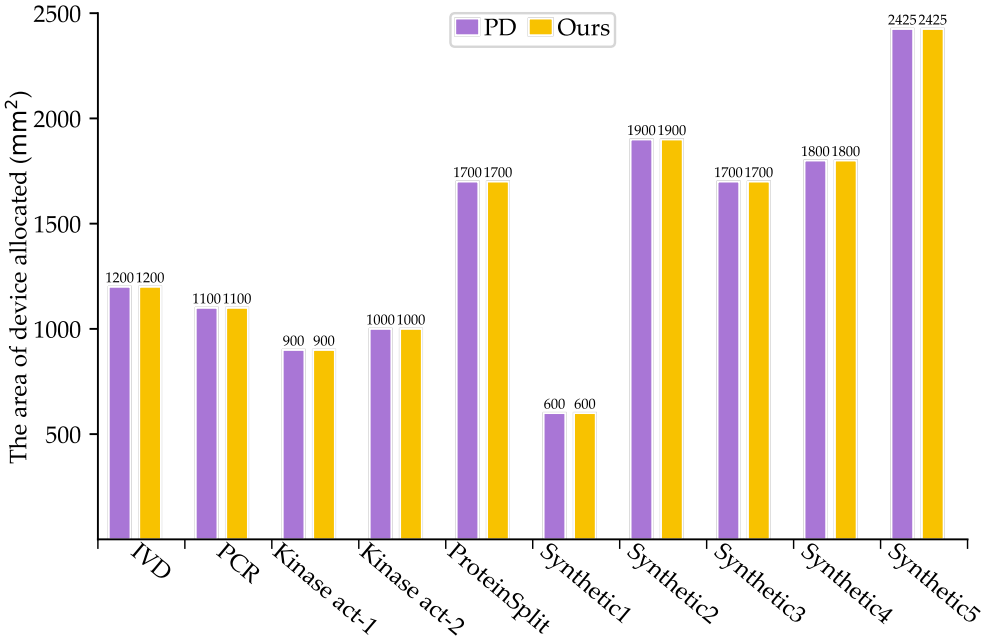


Figure 5. Comparison results between PD and SlimPort in terms of the area of devices allocated.

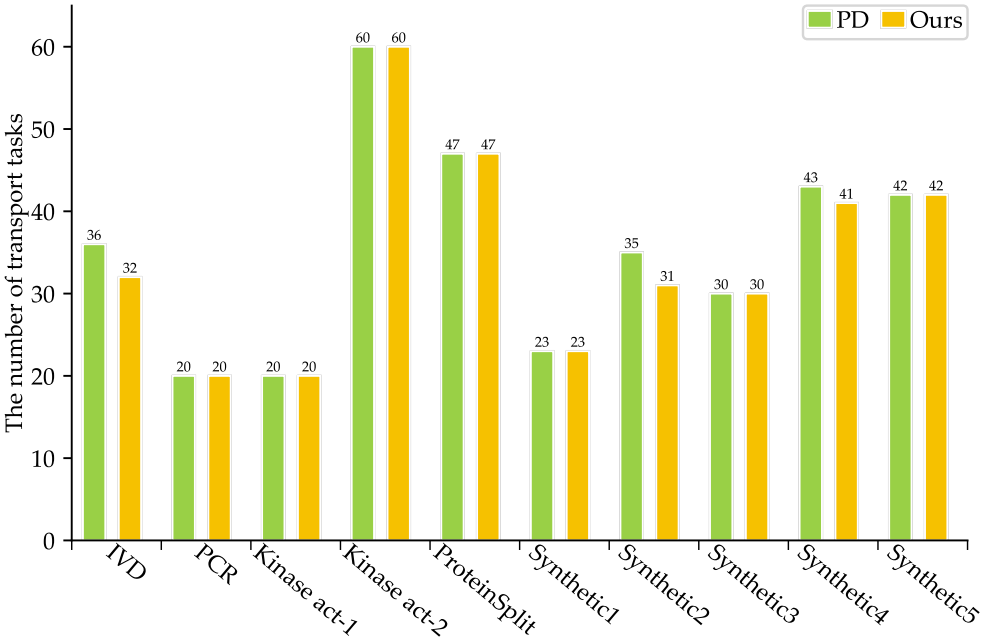


Figure 6. Comparison results between PD and SlimPort in terms of the number of transport tasks.

4.2. Validation of the Acceleration Strategies

To verify the effectiveness of the proposed acceleration strategies, we implemented three other methods, NA, CR, and BE, where NA is based on the ILP model without acceleration, CR is based on the ILP model with scheduling constraint reduction, and BE is based on the ILP model with upper boundary estimation of the fluidic port number. Figure 7 shows CPU times of NA, CR, BE, and SlimPort. It can be seen that the computational efficiency improved for SlimPort across all the benchmarks compared with NA, CR, and BE. Compared with NA, CR achieves a maximum speedup of 3.8× with an average speedup of 2.3×, BE achieves a maximum speedup of 6.2× with an average speedup of 3.1×, and

SlimPort achieves a maximum speedup of 13.5× with an average speedup of 6.7×. The acceleration capability of the acceleration strategies is demonstrated by the results above.

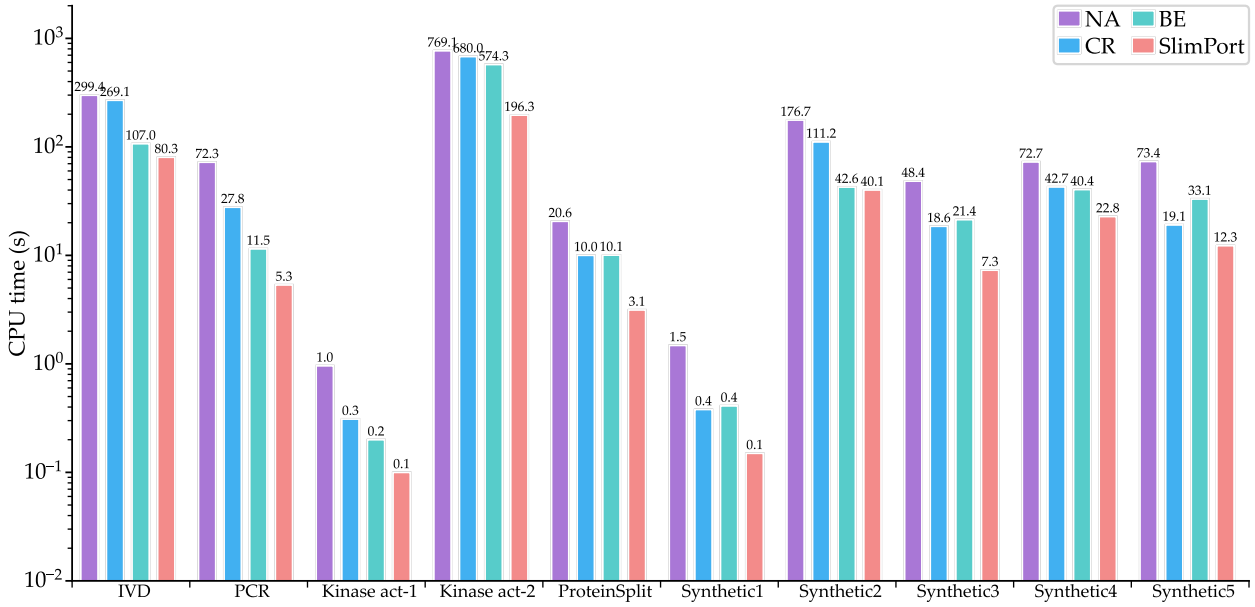


Figure 7. CPU times of NA, CR, BE, and SlimPort.

5. Conclusions

In this paper, we proposed a port-driven high-level synthesis method, called SlimPort, for CFMBs. Compared with the previous methods, SlimPort for the first time integrates the optimization of fluidic port number into the high-level synthesis. Volume management between devices with a non-fixed input/output ratio is also taken into account by SlimPort. Furthermore, two acceleration strategies were proposed to improve the overall performance of SlimPort: reducing scheduling constraints and estimating the upper bound of the fluidic port number. Experimental results on both real-life chip applications and synthetic benchmarks confirm the effectiveness of SlimPort.

Furthermore, in the future, we intend to implement a physical design method, which will be based on the results of the proposed SlimPort, for the co-design of the flow and control layers and flow path planning. The objective of this method is to generate an efficient chip architecture and optimize the number of fluidic and control ports together.

Author Contributions: Conceptualization, Y.P., X.H. and G.L.; methodology, Y.P., Y.X. and Z.C.; software, Y.P., Y.X. and Z.C.; validation, Y.P., X.H. and G.L.; formal analysis, Y.P., X.H. and G.L.; investigation, Y.P., X.H. and G.L.; resources, Y.P., Y.X. and Z.C.; data curation, Y.P., Y.X. and Z.C.; writing—original draft preparation, Y.P.; writing—review and editing, Y.P.; visualization, Y.P.; supervision, Y.P., X.H. and G.L.; project administration, Y.P., X.H. and G.L.; funding acquisition, X.H. and G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grant number 62372109, and the Natural Science Foundation of Fujian Province under grant numbers 2023J06017 and 2024J01984.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PDMS	Polydimethylsiloxane
CFMB	Continuous-flow microfluidic biochip
ILP	Integer linear programming

References

1. Huang, X.; Ho, T.Y.; Guo, W.; Li, B.; Chakrabarty, K.; Schlichtmann, U. Computer-Aided Design Techniques for Flow-Based Microfluidic Lab-on-a-Chip Systems. *ACM Comput. Surv.* **2021**, *54*, 97. [CrossRef]
2. Liu, G.; Huang, H.; Chen, Z.; Lin, H.; Liu, H.; Huang, X.; Guo, W. Design Automation for Continuous-Flow Microfluidic Biochips: A Comprehensive Review. *Integration* **2022**, *82*, 48–66. [CrossRef]
3. Chin, C.D.; Laksanasopin, T.; Cheung, Y.K.; Steinmiller, D.; Linder, V.; Parsa, H.; Wang, J.; Moore, H.; Rouse, R.; Umviligihozo, G.; et al. Microfluidics-Based Diagnostics of Infectious Diseases in the Developing World. *Nat. Med.* **2011**, *17*, 1015–1019. [CrossRef] [PubMed]
4. Huang, S.P.; Chuang, Y.J.; Lee, W.B.; Tsai, Y.C.; Lin, C.N.; Hsu, K.F.; Lee, G.B. An Integrated Microfluidic System for Rapid, Automatic and High-Throughput Staining of Clinical Tissue Samples for Diagnosis of Ovarian Cancer. *Lab Chip* **2020**, *20*, 1103–1109. [CrossRef] [PubMed]
5. Chu, Y.; Gao, Y.; Tang, W.; Qiang, L.; Han, Y.; Gao, J.; Zhang, Y.; Liu, H.; Han, L. Attomolar-Level Ultrasensitive and Multiplex microRNA Detection Enabled by a Nanomaterial Locally Assembled Microfluidic Biochip for Cancer Diagnosis. *Anal. Chem.* **2021**, *93*, 5129–5136.
6. Kartalov, E.P.; Zhong, J.F.; Scherer, A.; Quake, S.R.; Taylor, C.R.; Anderson, W.F. High-Throughput Multi-Antigen Microfluidic Fluorescence Immunoassays. *BioTechniques* **2006**, *40*, 85–90. [CrossRef] [PubMed]
7. Zhao, M.; Li, X.; Zhang, Y.; Wang, Y.; Wang, B.; Zheng, L.; Zhang, D.; Zhuang, S. Rapid Quantitative Detection of Chloramphenicol in Milk by Microfluidic Immunoassay. *Food Chem.* **2021**, *339*, 127857. [CrossRef] [PubMed]
8. Zheng, L.; Zhao, M.; Dai, B.; Xue, Z.; Kang, Y.; Liu, S.; Hou, L.; Zhuang, S.; Zhang, D. Integrated System for Rapid Enrichment and Detection of Airborne Polycyclic Aromatic Hydrocarbons. *Sci. Total Environ.* **2023**, *864*, 161057. [CrossRef] [PubMed]
9. Unger, M.A.; Chou, H.P.; Thorsen, T.; Scherer, A.; Quake, S.R. Monolithic Microfabricated Valves and Pumps by Multilayer Soft Lithography. *Science* **2000**, *288*, 113–116. [CrossRef] [PubMed]
10. Li, M.; Tseng, T.M.; Li, B.; Ho, T.Y.; Schlichtmann, U. Component-Oriented High-Level Synthesis for Continuous-Flow Microfluidics Considering Hybrid-Scheduling. In Proceedings of the 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, USA, 18–22 June 2017; pp. 1–6. [CrossRef]
11. Huang, C.M.; Liu, C.H.; Huang, J.D. Volume-Oriented Sample Preparation for Reactant Minimization on Flow-Based Microfluidic Biochips with Multi-Segment Mixers. In Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), EDA Consortium, Grenoble, France, 9–13 March 2015; pp. 1114–1119.
12. Thorsen, T.; Maerkl, S.J.; Quake, S.R. Microfluidic Large-Scale Integration. *Science* **2002**, *298*, 580–584. [CrossRef] [PubMed]
13. Melin, J.; Quake, S.R. Microfluidic Large-Scale Integration: The Evolution of Design Rules for Biological Automation. *Annu. Rev. Biophys.* **2007**, *36*, 213–231. [CrossRef] [PubMed]
14. Hu, K.; Chakrabarty, K.; Ho, T.Y. *Computer-Aided Design of Microfluidic Very Large Scale Integration (mVLSI) Biochips*; Springer: Cham, Switzerland, 2017; pp. 26–27. [CrossRef]
15. Perkel, J.M. Life Science Technologies: Microfluidics—Bringing New Things to Life Science. *Science* **2008**, *322*, 975–977.
16. Huang, X.; Ho, T.Y.; Li, Z.; Liu, G.; Wang, L.; Li, Q.; Guo, W.; Li, B.; Schlichtmann, U. MiniControl 2.0: Co-Synthesis of Flow and Control Layers for Microfluidic Biochips with Strictly Constrained Control Ports. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 5449–5463. [CrossRef]
17. Stanford Microfluidics Foundry Basic Design Tips. Available online: https://www.stanfordmicrofluidics.com/design-basics (accessed on 2 April 2025).
18. Tseng, T.M.; Li, M.; Freitas, D.N.; McAuley, T.; Li, B.; Ho, T.Y.; Araci, I.E.; Schlichtmann, U. Columba 2.0: A Co-Layout Synthesis Tool for Continuous-Flow Microfluidic Biochips. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2018**, *37*, 1588–1601. [CrossRef]
19. Liu, C.; Huang, X.; Li, B.; Yao, H.; Pop, P.; Ho, T.Y.; Schlichtmann, U. DCSA: Distributed Channel-Storage Architecture for Flow-Based Microfluidic Biochips. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2021**, *40*, 115–128. [CrossRef]
20. Huang, X.; Pan, Y.; Chen, Z.; Guo, W.; Wang, L.; Li, Q.; Wille, R.; Ho, T.Y.; Schlichtmann, U. Design Automation for Continuous-Flow Lab-on-a-Chip Systems: A One-Pass Paradigm. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2023**, *42*, 327–331. [CrossRef]

21. Zhu, Y.; Liu, G.; Guo, W.; Huang, X. FTCD: Fault-Tolerant Co-Design of Flow and Control Layers for Fully Programmable Valve Array Biochips. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2025**, 1–14. [CrossRef]
22. Huang, X.; Cai, H.; Guo, W.; Liu, G.; Ho, T.Y.; Chakrabarty, K.; Schlichtmann, U. Control-Logic Synthesis of Fully Programmable Valve Array Using Reinforcement Learning. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2024**, *43*, 277–290. [CrossRef]
23. Liu, G.; Zeng, Y.; Zhu, Y.; Cai, H.; Guo, W.; Li, Z.; Ho, T.Y.; Huang, X. Towards Automated Testing of Multiplexers in Fully Programmable Valve Array Biochips. In Proceedings of the 2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC), Incheon, Republic of Korea, 22–25 January 2024; pp. 570–575. [CrossRef]
24. Minhass, W.H.; Pop, P.; Madsen, J. System-Level Modeling and Synthesis of Flow-Based Microfluidic Biochips. In Proceedings of the 2011 14th International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES), Taipei, Taiwan, 9–14 October 2011; pp. 225–233. [CrossRef]
25. Dinh, T.A.; Yamashita, S.; Ho, T.Y.; Hara-Azumi, Y. A Clique-Based Approach to Find Binding and Scheduling Result in Flow-Based Microfluidic Biochips. In Proceedings of the 2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC), Yokohama, Japan, 22–25 January 2013; pp. 199–204. [CrossRef]
26. Huang, X.; Pan, Y.; Zhang, G.L.; Li, B.; Guo, W.; Ho, T.Y.; Schlichtmann, U. PathDriver+: Enhanced Path-Driven Architecture Design for Flow-Based Microfluidic Biochips. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 2185–2198. [CrossRef]
27. Chen, Z.; Zhu, Y.; Chen, Z.; Chen, Z.; Liu, G. High-Level Synthesis for Microfluidic Biochips Considering Actual Volume Management and Channel Storage. In Proceedings of the 2024 25th International Symposium on Quality Electronic Design (ISQED), San Francisco, CA, USA, 3–5 April 2024; pp. 1–8. [CrossRef]
28. Ye, Z.; Chen, Z.; Pan, Y.; Liu, G.; Guo, W.; Ho, T.Y.; Huang, X. Timing-Driven High-Level Synthesis for Continuous-Flow Microfluidic Biochips. In Proceedings of the 2024 25th International Symposium on Quality Electronic Design (ISQED), San Francisco, CA, USA, 3–5 April 2024; pp. 1–6. [CrossRef]
29. Bron, C.; Kerbosch, J. Algorithm 457: Finding All Cliques of an Undirected Graph. *Commun. ACM* **1973**, *16*, 575–577. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.