

# 整数計画ソルバーを用いた Networked Labs-on-Chip の設計手法

梅田 悠人<sup>†</sup> 山下 茂<sup>‡</sup>

## 1. はじめに

近年、生化学分野においてさまざまなバイオチップが研究されている。このバイオチップの一種として、Networked Labs-on-Chip (NLoC) というものがある [1]。NLoC はバイオチップの中でもチャンネルベースのものであり、直径 1 ミリメートル未満の経路が複数つながったネットワーク構造を持つ。NLoC はこの経路内を流れている非混和性流体に少量の試薬、いわゆる液滴を外部ポンプから注入する。そして、その外部ポンプから非混和性流体に力を加えて、液滴を操作することによって生化学実験を行う。NLoC の各経路は密閉されており、予期しない汚染や液滴の蒸発を防ぎ、液滴の長期間の保存を可能にするという利点がある [2]。

NLoC は 1 つの経路を 2 つ以上の経路に分岐させることができ、各経路はそれぞれ液滴の通りにくさを表す流体抵抗を持つ。流体抵抗の大きさは経路の長さに比例し、幅の大きさに反比例する。そして、経路に液滴が存在する場合、その経路の流体抵抗は大きくなる。これらの性質により、1 つのチップ上で複数の実験を行うことを可能にする。たとえば、1 つの経路から 2 つの経路に分岐する分岐点に液滴が到達したとき、液滴は流体抵抗の小さい後続経路に流れる。もし、分岐点に液滴が到達したときに、抵抗の小さい後続経路に他の液滴が既に存在しているとする。その場合、液滴が存在する後続経路の抵抗は大きくなり、分岐点に到達した液滴は、もう一方の後続経路に流れる [3]。この性質を活かして液滴の軌道を調整し、液滴に何らかの操作を行うモジュールを各経路に配置することにより、1 つのチップ上で複数の実験を行うことを可能にする。

NLoC における液滴の流れは、体積流量や経路間の圧力差のような多くの物理的要因および相互依存性が原因で予測することが難しい。このことにより、液滴が流れる経路の流体抵抗、液滴が経路を通るのに要する時間を求めるためには複雑な計算が必要である。同様の理由から、NLoC にはまだ自動設計手法がほとんど存在していない。そこで、設計者が液滴の位置と経路を通るのに要する時間を効率的に決定することを可能にする離散モデルが [4] で提案された。離散モデルの提案によって、チップ上の動作が抽象化され、設計の自動化が容易になると期待されている。離散モデルと実物の NLoC の動作を

比較したところ、離散モデルは 95 % 以上の確率で予期していた動作を行えることが [4] で示されている。しかし、この離散モデルを用いた設計手法はまだ提案されていない。

そこで本論文では、離散モデルにおける NLoC の設計手法を提案する。モジュールありの経路となしの経路を並列につなぎ、それを複数直列でつないだ構造にする。このような構造にすることにより、各経路に配置したモジュールを使用するか否かを選択でき、配置したモジュールを用いた複数の実験を行うことを可能にする。また、整数計画ソルバーを用いることによって設計の自動化を行う。分岐点での液滴の有無、分岐点での液滴の軌道などを変数として制約式を作成し、液滴を流すタイミング、液滴が経路を通るのに要する時間を決定する。提案手法により、複数の実験が可能な NLoC の自動設計が可能となる。

本論文の構成は以下の通りである。第 2 章では、NLoC の概要、設計上の問題点、離散モデルについて述べる。第 3 章では、本論文の提案手法について述べる。第 4 章では、実験結果と考察について述べる。最後に第 5 章では、まとめと今後の課題について述べる。

## 2. Networked Labs-on-Chip

本章では、本論文で提案する手法の基礎となる Networked Labs-on-Chip (NLoC) について説明する。まず NLoC のアーキテクチャ、続いてペイロード液滴のルーティング方法、続いて NLoC の設計上の問題点、最後にその問題点を克服した離散モデルについて説明する。

### 2.1 NLoC アーキテクチャ

NLoC アーキテクチャは、直径 1 ミリメートル未満の密閉された経路が複数つながったネットワーク構造となっており、外部ポンプ、経路、およびモジュールで構成されている。図 1 は、外部ポンプ、経路  $c_{in}, c_1, c_2, \dots, c_9$ 、モジュール  $m_1, m_2$  からなる NLoC アーキテクチャを示している。外部ポンプは、経路中を流れている非混和性流体に力を加えることによって、NLoC の入力経路に液滴を注入する。その液滴は経路中を流れている非混和性流体の内部を流れる。モジュールは、ペイロード液滴に対して混合、分割、融合、検出、加熱などのような基本的な操作を実行する。

ペイロード液滴を目的のモジュールに送るためにヘッド液滴が利用される。ヘッド液滴は、ペイロード液滴が

<sup>†</sup> 立命館大学大学院 情報理工学研究科

<sup>‡</sup> 立命館大学 情報理工学部

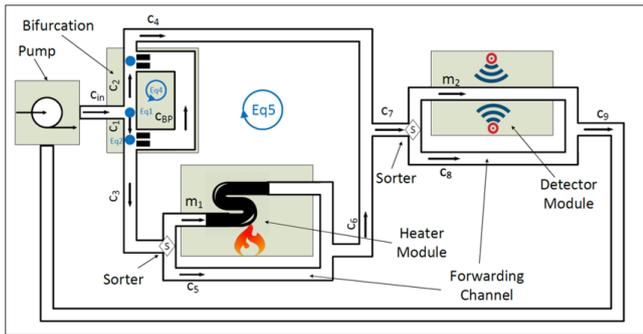


図 1: NLoC アーキテクチャ

通ってはいけない経路を一時的に封鎖する役目を持つ。しかし、このヘッダ液滴はペイロード液滴を破壊してしまう作用があるので、ペイロード液滴と衝突しないように操作する必要がある。また、ヘッダ液滴はモジュールを通らないように操作しなければならない。そのためヘッダ液滴は、モジュールを通らずに転送経路（図 1 の Forwarding Channel）を通り、モジュールの出口に転送される。ヘッダ液滴を転送経路に誘導するにはソータ [5]（図 1 の Sorter）が利用される。ソータはペイロード液滴とヘッダ液滴を体積の違いによって区別し、ペイロード液滴ならばモジュールへ、ヘッダ液滴ならば転送経路へ誘導する。

NLoC は 1 つの経路を 2 つ以上の経路に分岐させることができる。図 1 の NLoC アーキテクチャは、入力経路  $c_{in}$  を 2 つの後続経路  $c_1, c_2$  に分割する分岐（図 1 の Bifurcation）を含んでいる。この分岐における液滴ルーティングにより、設計者はペイロード液滴を検出モジュール（図 1 の Detector Module）を使って解析する前に、加熱モジュール（図 1 の Heater Module）を使って加熱するか否かを決定することができる。

## 2.2 液滴ルーティング

本節では、ヘッダ液滴を使用したペイロード液滴のルーティング方法について説明する。NLoC における液滴ルーティングは、各経路の異なる流体抵抗に基づいて行われる。経路の流体抵抗の大きさは、経路の長さに比例し、断面積に反比例する。液滴が分岐点に到達すると、液滴は最も流体抵抗が小さい後続経路に流れる。つまり、分岐後の複数の後続経路はそれぞれ異なる流体抵抗を持つことになる。また、液滴が経路に流れると、その経路の流体抵抗は大きくなる。この性質を活かし、ヘッダ液滴とペイロード液滴の流す順番を調整することで、ペイロード液滴を目的の経路へ流すことができる。ヘッダ液滴とペイロード液滴を流す順番のことを液滴シーケンスと呼ぶ。

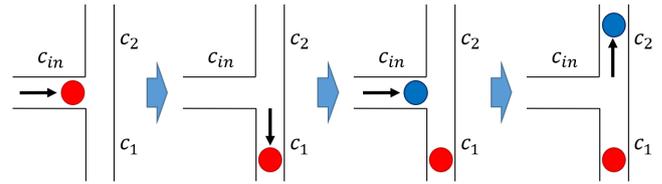


図 2: 液滴ルーティング

たとえば、図 1 の NLoC アーキテクチャには、入力経路  $c_{in}$  が 2 つの後続経路  $c_1, c_2$  に分岐する分岐点が含まれている。 $c_1$  の流体抵抗が  $c_2$  の流体抵抗よりも低いと仮定する。この場合、外部ポンプがペイロード液滴を注入すると、 $c_1$  の流体抵抗は  $c_2$  の流体抵抗より低いので、液滴は  $c_1$  に流れる。そして、その液滴はモジュール  $m_1$  によって加熱され、モジュール  $m_2$  によって分析され、最終的に外部ポンプに戻る。別の例として、 $c_1$  にヘッダ液滴が存在し、入力経路  $c_{in}$  にペイロード液滴が流れてくると仮定する。この場合、ヘッダ液滴が  $c_1$  の流体抵抗を増加させるため、 $c_1$  よりも  $c_2$  の流体抵抗のほうが小さくなる。よって、この場合ペイロード液滴は経路  $c_2$  に流れる。したがって、ペイロード液滴は加熱モジュールで加熱されることなく、直接検出モジュールに流れる。このようにして、設計者はペイロード液滴を検出モジュールを使って解析する前に、加熱するか否かを決定することができる。すなわち、この NLoC アーキテクチャにおいて、設計者は 2 つの異なる実験を選択できる。上記の例で紹介した液滴の動作を図 2 に示す。

## 2.3 設計上の問題点

NLoC における液滴の流れは、経路やモジュールの形状だけでなく、NLoC アーキテクチャ全体にわたる複数の物理特性や相互依存性に依存する。NLoC の物理特性には、時間単位で経路またはモジュールを通る流体の体積を示す体積流量  $Q[m^3/s]$ 、経路やモジュールの通りにくさを示す流体抵抗  $R[Pa \cdot s/m^3]$ 、経路やモジュールの端間の圧力の変化を表す圧力差  $\Delta P[Pa]$  がある。ハーゲン・ポアズイユの方程式 [6] により、これらの物理特性の比例関係を  $\Delta P = R \cdot Q$  で表すことができる。この式は電子回路のオームの法則  $V = IR$  に類似している。また、キルヒホッフの法則 [7] から、分岐前の経路の体積流量は、分岐後の各経路の体積流量の和に等しいことや、経路の閉サイクルにおける圧力差の和は 0 になることがわかる。キルヒホッフの法則を用いて方程式を立てることによって、経路やモジュールの流体抵抗を決定することができる。したがって、NLoC の物理的な振る舞いは、電子工学の法則を用いて表すことができる。

経路やモジュール内に液滴が存在するか否かも、各経

路の流体抵抗, 各経路での液滴の速度に影響を及ぼす。そのため, 液滴の流れを決定するために, 設計者は分岐点での液滴の動きを考慮する必要がある。液滴の速度は, 経路の体積流量を経路の断面積で除算することにより求めることができる。そして, 求めた速度を液滴の注入時間と組み合わせて計算することにより, ある時間における液滴の位置を求めることができる。

しかし, 液滴が入力経路に注入されたり, ある経路から次の経路に移動したりする度に, NLoC 内の全経路の体積流量が変化し, 流体抵抗も変化するので, 液滴の速度を再計算しなければならない。このような複雑さは, NLoC の設計自動化の妨げとなる。したがって, 設計自動化を実現させるためには, このような物理的な振る舞いを抽象化する必要がある。

## 2.4 離散モデル

設計自動化を実現するために, 物理的な振る舞いを抽象化した離散モデルが [4] で提案された。本節では, その離散モデルについて説明する。離散モデルでは, 液滴が経路を流れる時間をタイムステップとして表す。このことにより, 設計者は液滴が経路やモジュールを流れるのに必要な時間を記録したり, モジュールの動作をタイムステップの単位で実行したりすることが可能となる。液滴が分岐点に到達したとき, その液滴は通るのに必要なタイムステップ数が最も少ない後続経路に流れる。つまり, 通るのに必要なタイムステップ数が最も少ない後続経路が, 最も抵抗の小さい経路となる。また, 離散モデルにおいても, 経路に液滴が含まれている場合は, その経路の抵抗が大きくなる。よって, ある液滴が分岐点に到達したときに, 最も抵抗が小さい後続経路に他の液滴が含まれている場合, 分岐点に到達した液滴は 2 番目に抵抗が小さい後続経路に流れる。

図 3 は図 1 中の  $c_{in}$  から  $c_1, c_2$  に分岐する分岐点における離散モデルを示している。図 3 中の経路内のマス目は, 液滴がこの経路を通過するのに必要なタイムステップ数を表している。タイムステップが 1 つ進むごとに, 液滴も 1 マス進むようになっている。図 3 の各々は, タイムステップが 1 つ進むごとに, 液滴も 1 マスずつ進んでいく様子を示している。離散モデルを使用すると, 経路内の液滴の位置を直感的かつ効率的に確認することができる。

離散モデルと実物の NLoC の動作を比較したところ, 離散モデルは 95% 以上の確率で予期していた動作を行えることが [4] で示されている。この離散モデルを用いることによって, NLoC の設計自動化が比較的容易になった。しかし, 具体的な設計手法はまだ提案されていない。

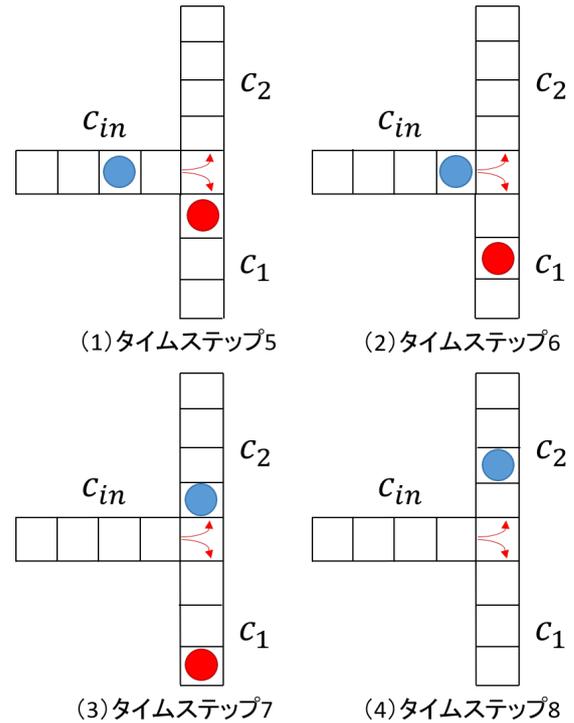


図 3: 離散モデルでの液滴の移動の様子

## 3. 整数計画ソルバーを用いた NLoC の設計手法

本論文では, 2.4 節で説明した NLoC の離散モデルを, 整数計画ソルバーを用いて設計する手法を提案する。NLoC は, 1 つのチップ上で複数の実験が可能な構造に設計されることが求められている。そこで, 複数の実験が可能な構造を 3.1 節にて説明する。そして, 整数計画ソルバーを用いて設計を行うために, NLoC の設計問題を整数計画問題へと定式化する方法を 3.2 節にて説明する。

### 3.1 複数の実験が可能な構造

本節では, 1 つのチップ上で複数の実験が可能な構造を設計するために, モジュールありの経路となしの経路を並列につなぎ, それらを複数直列でつないだ構造を提案する。このような構造にすることにより, 利用者は各経路に配置したモジュールを使用するか否かを選択でき, 配置したモジュールを用いた複数の実験を行うことを可能にする。

図 4 は, 1 つの経路から 2 つの経路に分岐している分岐点を 4 つ含む構造を示している。各分岐点における 2 つの後続経路のうち, 1 つの後続経路はモジュールを含んでいる。図 4 中の  $c_0, c_1, c_2, \dots, c_{16}$  は各経路の番号,  $m_0, m_1, m_2, m_3$  は各モジュールの番号を示している。また, 図 4 中の赤い矢印は流体抵抗の小さい後続経路, 青い矢印は流体抵抗の大きい後続経路を示している。図 4

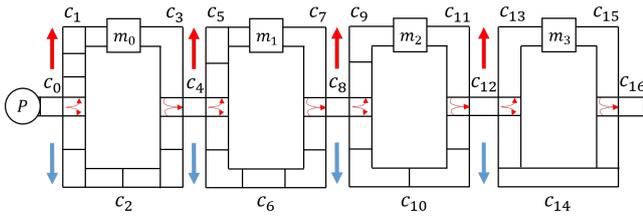


図 4: 分岐点 4 つの構造

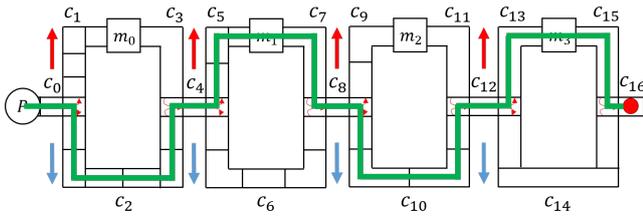


図 5:  $m_1, m_3$  を使った実験

の構造の場合、モジュールありの後続経路は流体抵抗が小さく、モジュールなしの後続経路は流体抵抗が大きい。

図 4 の構造は、2 つの経路を選択できる分岐点を 4 つ含む。よって、 $2^4 = 16$  通りの実験が可能であることがわかる。この 16 通りの実験をすべて行えるようにするためには、各実験を実行可能な液滴シーケンスがすべて存在するかを確認しなければならない。たとえば、図 5 のように、モジュール  $m_1, m_3$  を使う実験について考える。このとき、ペイロード液滴は 1 つ目の分岐点で下に移動しなければならない。しかし、 $c_1$  は  $c_2$  よりも流体抵抗が小さいため、ペイロード液滴は上へ移動する。そこで、ペイロード液滴よりも先にヘッド液滴を注入する必要がある。ペイロード液滴が  $c_0$  に到着したときに、 $c_1$  中のヘッド液滴数が、 $c_2$  中のヘッド液滴数よりも多ければ、 $c_2$  の流体抵抗は  $c_1$  の流体抵抗を下回るため、ペイロード液滴は下に移動する。他の分岐点にペイロード液滴が到着したときも同様に、その分岐点の後続経路にヘッド液滴がいくつ存在するかを考慮する必要がある。つまり、モジュール  $m_1, m_3$  を使う実験を行うためには、ペイロード液滴が  $c_0$  に到着したときは、 $c_1$  中のヘッド液滴数を  $c_2$  中のヘッド液滴数よりも多く、 $c_4$  に到着したときは、 $c_5$  中のヘッド液滴数を  $c_6$  中のヘッド液滴数以下に、 $c_8$  に到着したときは、 $c_9$  中のヘッド液滴数を  $c_{10}$  中のヘッド液滴数よりも多く、 $c_{12}$  に到着したときは、 $c_{13}$  中のヘッド液滴数を  $c_{14}$  中のヘッド液滴数以下にする必要がある。

このように、与えられたすべての実験を実行可能な液滴シーケンスがすべて存在するような構造を設計しなければならない。また、実行時間を短縮するために、各経

路のタイムステップを出来る限り減らす必要がある。このような構造を設計するために、整数計画ソルバーを用いた解法を 3.2 節にて提案する。

### 3.2 整数計画ソルバーを用いた設計

本節では、整数計画ソルバーを用いて設計を行うために、設計問題を整数計画問題へと定式化する方法を提案する。整数計画ソルバーとは、整数計画問題を解くソフトウェアである。整数計画問題とは最適化問題の 1 つで、1 つの目的関数と複数の制約式で構成される。整数計画問題では、目的関数と制約式内の全ての変数で整数が用いられ、設定した目的関数を最大化または最小化するような変数の組を探索することが目的となる。このとき、各変数はすべての制約式を満たさなければならない [8]。本節では、設計問題を整数計画問題へと定式化する手法を提案する。

#### 3.2.1 問題定義

本節では、NLoC の設計問題を整数計画問題へと定式化するために、設計における問題定義を行う。この設計問題における目的は『液滴が各経路を通るのに要するタイムステップ数の最小化』である。与えられる入力は以下である。

1. 総分岐数
2. 総ヘッド液滴数
3. 実行すべき実験の集合

また、NLoC の設計にて決定する必要がある変数は以下である。

1. 液滴が各経路を通るのに要するタイムステップ
2. 液滴が各モジュールを通るのに要するタイムステップ
3. 各ヘッド液滴を注入するときのタイムステップ
4. ペイロード液滴を注入するときのタイムステップ

また、設計するときに『すべての実験に対応するペイロード液滴の動きを実現する各ヘッド液滴の注入時間が存在する』という条件を満たさなければならない。

これらの目的、入力、変数、条件すべてが過不足なく定式化できれば、整数計画ソルバーによって最適解を求めることができる。NLoC の設計問題に必要な変数や制約式は、分岐数やヘッド液滴数などの入力によって変化する。そこで提案手法では、プログラムによる目的関数、変数、制約式の自動生成を行う。提案手法では上記の問題をプログラムにて定式化した後、その定式化した問題

を整数計画ソルバーに与えることによって、すべての実行すべき実験が可能かつ経路のタイムステップの合計長が最小となるような NLoC の自動設計を実現する。定式化手法は次節にて説明する。

### 3.2.2 定式化

本節では、NLoC の設計問題の定式化手法を説明する。まず、使用する各変数について説明する。

$Header[i][j]$  は、実験  $i$  における  $j$  番目のヘッダ液滴注入時のタイムステップを表現する 2 次元配列変数である。同様に、 $Payload[i]$  は、実験  $i$  におけるペイロード液滴注入時のタイムステップを表現する 1 次元配列変数である。

$Channel[i]$  は、液滴が各経路を通るのに必要なタイムステップを表現する 1 次元配列変数である。 $i$  は各経路の番号を示す。以後、各合流点と各分岐点をつなぐ経路のことを、連結経路と呼ぶ。

$Module[i]$  は、液滴が各モジュールを通るのにかかるタイムステップを表現する 1 次元配列変数である。 $i$  は各モジュールの番号を示す。ここでは  $Module[i]$  の値はすべて 1 とする。

$Bif.time[i][j][k]$  は、実験  $i$  において、 $j$  番目の連結経路に  $k$  番目のヘッダ液滴が到着するタイムステップを表現する 3 次元配列変数である。

$Bif.move[i][j][k]$  は、各分岐点における各液滴の軌道を表現する 3 次元配列変数である。値が 1 の場合、実験  $i$  において、 $j$  番目の分岐点で  $k$  番目の液滴は上に、値が 0 の場合、下に移動することを示す。

$Pred[i][j][k][l]$  は、各分岐点に各液滴が到着したとき、後続経路に他の液滴が存在するか否かを表現する 4 次元配列変数である。値が 1 の場合、実験  $i$  において、 $j$  番目の分岐点に  $k$  番目の液滴が到着したとき、 $l$  番目の液滴が上の後続経路に、値が  $-1$  の場合、下の後続経路に存在することを示し、値が 0 の場合、上の後続経路にも下の後続経路にも存在しないことを示す。

次に、各変数が満たさなければならない制約式について説明する。まず、『複数のヘッダ液滴が同じタイムステップに注入されてはいけない』、また、ヘッダ液滴には番号が振られており、『ヘッダ液滴は番号が小さい順に注入されなければならない』という制約が必要である。その制約式を式 1 に示す。同様に、『ペイロード液滴は各ヘッダ液滴と同時に注入されてはいけない』という制約が必要である。その制約式を式 2 に示す。

$$Header[i][j+1] > Header[i][j] \quad (1)$$

$$Payload[i] \neq Header[i][j] \quad (2)$$

実験において、1 番目に注入される液滴の注入時のタイムステップは 1 でなければならない。しかし、1 番目に注入される液滴は、ペイロード液滴なのかヘッダ液滴なのかはわからない。よって、式 3 のように、『ペイロード液滴の注入時のタイムステップが番号 0 のヘッダ液滴の注入時のタイムステップよりも早いならば、ペイロード液滴の注入時のタイムステップを 1 とする』という制約式が必要である。同様に、式 4 のように、『ペイロード液滴の注入時のタイムステップが番号 0 のヘッダ液滴を注入するタイムステップよりも遅いならば、番号 0 のヘッダ液滴の注入時のタイムステップを 1 とする』という制約式も必要である。

$$Payload[i] < Header[i][0] \rightarrow Payload[i] = 1 \quad (3)$$

$$Payload[i] > Header[i][0] \rightarrow Header[i][0] = 1 \quad (4)$$

次に、各分岐点の後続経路の流体抵抗に関する制約式について説明する。3.1 節で説明した構造の各分岐点では、モジュールがある上の後続経路とモジュールがない下の後続経路に分岐する。そして、上の後続経路の流体抵抗は下の後続経路の流体抵抗よりも小さいとしている。つまり、上の後続経路を液滴が通過するのに要するタイムステップは、下の後続経路を液滴が通過するのに要するタイムステップよりも少なくしなければならない。よって、 $Channel[i]$  を上の後続経路、 $Channel[i+1]$  を下の後続経路を液滴が通過するのに要するタイムステップとすると、式 5 のような制約式が必要となる。

$$Channel[i+1] > Channel[i] \quad (5)$$

次に、各液滴が連結経路に到着するタイムステップに関する制約式について説明する。まず、1 番目の連結経路に各液滴が到着するタイムステップは、各液滴が注入される時間と同じと考えて良い。各液滴が 2 番目以降の連結経路に到着するタイムステップは、その 1 つ前の分岐点でどの方向に移動したかによって変化する。たとえば、図 4 中の  $c_4$  に、ある液滴が到着する時間  $B$  を求めたいとする。その液滴が  $c_0$  に到着した時間を  $B_0$  とする。その液滴が 1 つ目の分岐点で上の後続経路に移動したとすると、液滴は  $c_4$  に到着するまでに  $c_1, m_0, c_3$  を通る。よって、 $B$  は式 6 のように求めることができる。また、その液滴が 1 つ目の分岐点で下の後続経路に移動したとすると、液滴は  $c_4$  に到着するまでに  $c_2$  を通る。よって、 $B$  は式 7 のように求めることができる。

$$B = B_0 + Channel[1] + Module[0] + Channel[3] + Channel[4] \quad (6)$$

$$B = B_0 + Channel[2] + Channel[4] \quad (7)$$

一般の場合には、『液滴が  $j$  番目の分岐点で上の後続経路に移動したならば、 $j + 1$  番目の分岐点に到着するタイムステップは、その液滴が  $j$  番目の連結経路に到着したときのタイムステップ、 $j$  番目の分岐点の上の後続経路を通るのに必要なタイムステップ、 $j + 1$  番目の連結経路のタイムステップの和となる』という制約と、『液滴が  $j$  番目の分岐点で下の後続経路に移動したならば、 $j + 1$  番目の分岐点に到着するタイムステップは、その液滴が  $j$  番目の連結経路に到着したときのタイムステップ、 $j$  番目の分岐点の下の後続経路を通るのに必要なタイムステップ、 $j + 1$  番目の連結経路のタイムステップの和となる』という制約が必要である。これらの制約式を式 8、式 9 に示す。

$$\begin{aligned} & Bif\_move[i][j][k] = 1 \\ \rightarrow & Bif\_time[i][j+1][k] = Bif\_time[i][j][k] \\ & + Channel[4*j+1] + Module[j] \\ & + Channel[4*j+3] + Channel[4*j+4] \quad (8) \end{aligned}$$

$$\begin{aligned} & Bif\_move[i][j][k] = 0 \\ \rightarrow & Bif\_time[i][j+1][k] = Bif\_time[i][j][k] \\ & + Channel[4*j+2] + Channel[4*j+4] \quad (9) \end{aligned}$$

次に、合流点における液滴同士の衝突を防ぐ制約式について説明する。各液滴が同じ連結経路に到達するタイムステップが等しくなければ、液滴同士は合流点で衝突しないことを表現できる。よって、制約式は式 10 のようになる。

$$Bif\_time[i][j][k] \neq Bif\_time[i][j][l] \quad (10)$$

次に、各分岐点における各液滴の軌道に関する制約式について説明する。 $k$  番目の液滴が  $j$  番目の分岐点に到着したとする。 $k$  番目の液滴は  $j$  番目の分岐点の後続経路に存在する他の液滴の数によって、どちらの後続経路に移動するかが決まる。このように、上の後続経路の液滴数と下の後続経路の液滴数を比較することによって、分岐点に到着した液滴の軌道を決定することができる。

$k$  番目の液滴が  $j$  番目の分岐点に到達したときに、その分岐点の上の後続経路に  $l$  番目の液滴が存在する条件は、『 $l$  番目の液滴は  $j$  番目の分岐点の上の後続経路に移動する』と『 $k$  番目の液滴が  $j$  番目の連結経路に到着するタイムステップから  $l$  番目の液滴が  $j$  番目の連結経路に到着するタイムステップを引いた数が後続経路のタイムステップよりも小さい、かつ負数にならない』の 2 つである。この 2 つの条件を満たした場合、 $k$  番目の液滴

が  $j$  番目の分岐点に到着したとき、 $l$  番目の液滴が上の後続経路に存在する。これらを制約式にすると、式 11 のようになる。

$$\begin{aligned} & (Bif\_move[i][j][l] = 1) \\ \text{かつ} & \\ & (Channel[1+4*j] \\ & \geq Bif\_time[i][j][k] - Bif\_time[i][j][l]) \\ \text{かつ} & \\ & (Bif\_time[i][j][k] - Bif\_time[i][j][l] > 0) \\ \rightarrow & Pred[i][j][k][l] = 1 \quad (11) \end{aligned}$$

また、 $k$  番目の液滴が  $j$  番目の分岐点に到達したときに、その分岐点の下の後続経路に  $l$  番目の液滴が存在する条件は『 $l$  番目の液滴は  $j$  番目の分岐点の下の後続経路に移動する』と『 $k$  番目の液滴が  $j$  番目の連結経路に到着するタイムステップから  $l$  番目の液滴が  $j$  番目の連結経路に到着するタイムステップを引いた数が後続経路のタイムステップよりも小さい、かつ負数にならない』の 2 つである。この 2 つの条件を満たした場合、 $k$  番目の液滴が  $j$  番目の分岐点に到着したとき、 $l$  番目の液滴が下の後続経路に存在する。これらを制約式にすると、式 12 のようになる。

$$\begin{aligned} & (Bif\_move[i][j][l] = 0) \\ \text{かつ} & \\ & (Channel[2+4*j] \\ & \geq Bif\_time[i][j][k] - Bif\_time[i][j][l]) \\ \text{かつ} & \\ & (Bif\_time[i][j][k] - Bif\_time[i][j][l] > 0) \\ \rightarrow & Pred[i][j][k][l] = -1 \quad (12) \end{aligned}$$

また、上の後続経路に液滴が存在する条件を満たさない、かつ下の後続経路に液滴が存在する条件を満たさない場合、 $k$  番目の液滴が  $j$  番目の分岐点に到着したとき、 $l$  番目の液滴は上の後続経路にも下の後続経路にも存在しない。これらを制約式にすると、式 13 のようになる。

$$\begin{aligned} & \neg\{(Bif\_move[i][j][l] = 1)\} \\ \text{かつ} & \\ & (Channel[1+4*j] \\ & \geq Bif\_time[i][j][k] - Bif\_time[i][j][l]) \\ \text{かつ} & \\ & (Bif\_time[i][j][k] - Bif\_time[i][j][l] > 0)\} \\ \text{かつ} & \\ & \neg\{(Bif\_move[i][j][l] = 0)\} \end{aligned}$$

かつ

$$\begin{aligned} & (Channel[2 + 4 * j] \\ & \geq Bif\_time[i][j][k] - Bif\_time[i][j][l]) \end{aligned}$$

かつ

$$(Bif\_time[i][j][k] - Bif\_time[i][j][l] > 0)$$

$$\rightarrow Pred[i][j][k][l] = 0 \quad (13)$$

$Pred[i][j][k]$  の値をすべて求め、その総和を求めることにより、分岐点での液滴の軌道が決まる。 $Pred[i][j][k]$  の総和が 0 以下ならば、実験  $i$  において、 $k$  番目のヘッダ液滴が  $j$  番目の分岐点に到着したとき、その分岐点の上の後続経路に存在する液滴数は、下の後続経路に存在する液滴数以下であることがわかる。よって、実験  $i$  において、 $k$  番目のヘッダ液滴は  $j$  番目の分岐点で上に移動する。つまり、制約式は式 14 のようになる。

また、 $Pred[i][j][k]$  の総和が 0 よりも大きいならば、実験  $i$  において、 $k$  番目のヘッダ液滴が  $j$  番目の分岐点に到着したとき、その分岐点の上の後続経路に存在する液滴数は、下の後続経路に存在する液滴数よりも多いことがわかる。よって、実験  $i$  において、 $k$  番目のヘッダ液滴は  $j$  番目の分岐点で下に移動する。つまり、制約式は式 15 のようになる。 $Total\_H$  は入力で与えられた総ヘッダ液滴数を表す。

$$\begin{aligned} & \sum_{m=0}^{Total\_H} Pred[i][j][k][m] \leq 0 \\ & \rightarrow Bif\_move[i][j][k] = 1 \end{aligned} \quad (14)$$

$$\begin{aligned} & \sum_{m=0}^{Total\_H} Pred[i][j][k][m] > 0 \\ & \rightarrow Bif\_move[i][j][k] = 0 \end{aligned} \quad (15)$$

目的関数には、 $Channel$  の総和を設定する。最も経路のタイムステップ数が少ない構造が最適な構造であると言えるので、目的関数は式 16 のように定式化できる。 $Total\_Bif$  は入力で与えられた総分岐数を表す。

$$minimize \sum_{i=0}^{4 * Total\_Bif} Channel[i] \quad (16)$$

整数計画ソルバーは、制約条件を満たす解を見つけてくれる。整数計画ソルバーで計算した最適解は、計算終了後に  $Channel$ ,  $Header$ ,  $Payload$  の値を参照することで取得可能である。提案手法の場合、 $Channel$  の値が最適な構造における経路のタイムステップとなり、 $Header$  と  $Payload$  の値が各実験を実行可能な液滴シーケンスとなるので、これらの値を決定することで設計完了となる。

## 4. 実験結果と考察

本章では、第 3 章で提案した手法を適用した実験結果と、その考察を示す。

### 4.1 評価方法

提案手法の評価を行うために、提案手法を C++ で実装した。また、整数計画ソルバーとして CPLEX[9] を使用した。CPLEX は整数計画ソルバーとしての機能以外にも、C++ や Java などの他のプログラムからも利用できる API を備えている。この API を利用することで、プログラムによる目的関数や制約式の自動生成が可能となる。

提案手法の目的は、液滴が各経路を通るのに要するタイムステップの最小化を行い、ソルバーの解として得られた構造において、与えられた実験を実行できる液滴シーケンスをすべて求めることである。

### 4.2 実験結果と考察

実験結果を表 1 に示す。表 1 には、各テストケースにおける分岐数 (Bif)、ヘッダ液滴数 (H)、実験数 (Ex) を示している。また、実験結果として、設計された構造の各経路を液滴が通過するのに要するタイムステップの合計 (Sum) と、与えられた整数計画問題を解くのに要した計算時間 (Time) を示している。実験では、計算時間を 50000 秒に制限した。Sum に "×", Time に "over" の記載があるのは、時間内に結果が得られなかったためである。

各実験において、整数計画ソルバーは与えられた分岐数、ヘッダ液滴数、実行すべき実験という条件のもとで、各経路を液滴が通過するのに要するタイムステップの合計が他の解と比較して最も小さい構造を出力している。また、その構造において、与えられたすべての実験を行うことができる液滴シーケンスを出力している。

表 1 から、ヘッダ液滴を多く注入すれば、液滴が各経路を通過するのに要するタイムステップの合計を削減できることがわかる。しかし、Test2-b と Test2-c を比較すると、ヘッダ液滴数が Test2-c のほうが多いが、タイムステップの合計値は等しい。これは、2 分岐の構造においては、制約式からタイムステップの合計が 13 より小さくならないためである。よって、2 分岐の構造に最も適しているヘッダ液滴数は 2 だということがわかる。

また、分岐数、ヘッダ液滴数、実験数が多くなるにしたがって、最適解の計算にはかなりの計算時間を要するようになることがわかる。また、ヘッダ液滴数を減らすことによって、整数計画ソルバーに与える変数と制約式を減らし、計算時間を削減できる。しかし、ヘッダ液滴数が少ない状態で複数の実験を行いたい場合、各経路のタイムステップ数の合計値は大きくなってしまふ。Test3-c,

表 1: 実験結果

Test	Bif	H	Ex	Sum	Time (秒)
Test2-a	2	1	4	17	0.02
Test2-b	2	2	4	13	0.326
Test2-c	2	3	4	13	5.73
Test3-a	3	1	8	37	0.15
Test3-b	3	2	8	25	765.15
Test3-c	3	3	8	×	over
Test3-d	3	3	6	19	33543.78
Test4-a	4	1	16	77	237.13
Test4-b	4	2	16	×	over
Test4-c	4	2	6	30	38251.94
Test5-a	5	1	32	157	2562.01
Test5-b	5	2	32	×	over
Test4-c	5	2	4	35	1947.53

Test4-b, Test5-b はすべての制約条件を満たす変数の値を求めることが困難となり、時間内に計算できないことがわかった。

## 5. おわりに

本研究では、離散モデルを用いた NLoC の設計手法を提案した。離散モデルはチップ上の動作が抽象化されており、95%以上の確率で予期していた動作を行うことが可能である。この離散モデルの提案により、複雑な計算を行う必要がなくなり、NLoC の設計自動化が比較的容易になったと [4] で示されている。しかし、この離散モデルを用いた具体的な設計手法はまだ提案されていなかった。そこで提案手法では、この離散モデルを用いて、NLoC の設計自動化を実現した。また、設計自動化を実現するために、NLoC の設計問題を整数計画問題に定式化し、整数計画ソルバーを用いて計算を行った。その結果、与えられた分岐数、ヘッド液滴数、行いたい実験という条件のもとで、各経路を液滴が通過するのに要するタイムステップの合計が最も小さい構造の設計と、その構造において、与えられたすべての実験を行うことができる液滴シーケンスの生成を自動化することに成功した。

提案手法では、分岐数、ヘッド液滴数、実験数が多くなるにしたがって、すべての制約条件を満たす変数の値を求めることが困難となり、最適解の計算にはかなりの計算時間が必要となった。そのため、分岐数、ヘッド液滴数、実験数が多い場合は計算できないことがわかった。よって今後は、より少ない変数でより単純な制約式に定式化できるように改善し、実行時間の短縮を行うことが課題となる。また、整数計画ソルバーを用いず、ヒュー

リスティックスによる解法を提案し、実行時間の大幅な短縮を行い、6分岐以上の構造にも対応できるようにすることも今後の課題である。

## 参考文献

- [1] Elena De Leo, Laura Galluccio, Al Lombardo, and Giacomo Morabito. Networked labs-on-a-chip (nloc): Introducing networking technologies in micro uidic systems. *Nano Communication Networks*, Vol. 3, No. 4, pp. 217-228, 2012.
- [2] Daniel Mark, Stefan Haeberle, Gunter Roth, Felix von Stetten, and Roland Zengerle. Microfluidic lab-on-a-chip platforms: requirements, characteristics and applications. *Chem. Soc. Rev.*, Vol. 39, pp. 1153-1182, 2010.
- [3] Werner Haselmayr, Andrea Biral, Andreas Grimmer, Andrea Zanella, Andreas Springer, and Robert Wille. Addressing multiple nodes in networked lab-on-a-chips without payload re-injection. In *International Conference on Communications (ICC)*, pp. 1-6, 2017.
- [4] Andreas Grimmer, Werner Haselmayr, Andreas Springer, and Robert Wille. A discrete model for networked labs-on-chips: Linking the physical world to design automation. In *Proceedings of the 54th Annual Design Automation Conference 2017*, pp. 1-6. ACM, 2017.
- [5] Yung-Chieh Tan, Yao Li Ho, and Abraham Phillip Lee. Micro uidic sorting of droplets by size. *Micro uidics and Nano uidics*, Vol. 4, No. 4, p. 343, Jun 2007.
- [6] D.J. Acheson. *Elementary Fluid Dynamics*. Oxford Applied Mathematics and Computing Science Series. Clarendon Press, 1990.
- [7] Kwang W. Oh, Kangsun Lee, Byungwook Ahn, and Edward P. Furlani. Design of pressure-driven micro uidic networks using electric circuit analogy. *Lab Chip*, Vol. 12, pp. 515-545, 2012.
- [8] 藤江哲也. 整数計画法による定式化入門. *オペレーションズ・リサーチ*, pp. 190-197, 2012年4月.
- [9] IBM. *IBM ILOG. IBM ILOG CPLEX V12.1 User's Manual for CPLEX*. 2009.