Vol.2017-ARC-225 No.21
Vol.2017-SLDM-179 No.21
Vol.2017-EMB-44 No.21
2017/3/9

試料生成における汚染問題を考慮したDMFB合成手法

北川 大樹[1,a)]　山下 茂[2,b)]

概要：近年, 生化学分野にてさまざまなバイオチップが研究されている. DMFB (Digital Microfluidic Biochip) とは, このバイオチップの一種で, チップ上で化学実験を実行できるといった特徴を持つ. DMFB 上において化学実験は, 二次元配列状に配置した電極で液滴を操作することによって実行される. DMFB に実験を実装するには, 実行のための情報をいくつかのプロセスにて決定する必要がある. ルーティングはこの実装のためのプロセスの 1 つで, 各液滴の経路を決定するプロセスである. このルーティングでは, 一度使用したセルは使用した液滴の残留物により汚染されることを考慮しなくてはならない. 一度使用したセルに残留物が残るという問題は汚染問題と呼ばれ, ルーティングでは他の経路が使用した経路はなるべく使用しないように経路探索する必要がある. DMFB 上で実行される実験の 1 つに試料生成があり, この実験では前述した汚染問題を無視できる場合が存在する. そこで, 本論文では試料生成における許容解を考慮した DMFB 合成手法を提案する.

1. はじめに

近年, 実験室規模で行われる従来の実験方法に代わる, 新たな方法として DMFB (Digital Microfluidic Biochip) と呼ばれるバイオデバイスを用いた実験方法が注目されている. DMFB とは, 実験に必要な試薬などを液滴として操作することで, 現在手作業で行われている実験作業のほとんどをチップ上で実現可能にしたデバイスである. DMFB を利用すれば, 実験に必要なさまざまなコストを抑えつつ, 高い信頼性とスループットで実験を行うことが可能となる [1].

DMFB の合成にはさまざまなプロセスが存在する. ルーティングはこの合成プロセスの 1 つで, 各液滴の移動経路を決定するプロセスである. この経路決定の際には, 使用した経路に液滴が一部残留してしまうことを考慮する必要がある. このような制約は汚染問題と呼ばれ, 汚染問題を考慮したルーティング手法として, [2,3] が提案されている.

[2,3] の手法では, 汚染箇所を利用しない様に迂回したり, 洗浄液滴を用いて汚染箇所を洗浄する, などの方法で汚染問題を解決している. しかし, これらの対処方法は, 液滴の遠回りや汚染箇所の洗浄待ちが発生するなどの理由から, 液滴の移動に時間がかかる場合が多い. よって, 液滴の移動時間を削減するには, 汚染箇所の使用を最小限に抑える必要がある.

[1] 立命館大学院 情報理工学研究科
[2] 立命館大学 情報理工学部 情報システム学科
[a)] gawasan@ngc.is.ritsumei.ac.jp
[b)] ger@cs.ritsumei.ac.jp

DMFB 上でよく実行される実験の 1 つに, 試料生成がある. 試料生成とは, 通常の実験の前に実行される実験で, ある濃度の液滴群を生成する実験である. 通常の実験と違い, 試料生成では同じ種類の液滴が複数回利用されているため, 汚染問題を許容できる場合が多い. しかし, 上記の既存手法は試料生成のような実験を考慮していないため, 試料生成の場合には更なる最適化の余地が存在する. そこで本論文では, 試料生成における汚染問題を考慮した, DMFB の合成手法を提案する. また, 提案手法を実現する方法として, 整数計画ソルバーを利用した解法と, ヒューリスティックスによる解法の 2 種類を本論文では提案する. これら 2 種類の解法をそれぞれ比較した結果, ヒューリスティックスによる解法は, 整数計画ソルバーを利用した最適解に最大で 99%近い結果を得ることができた.

本論文の構成は以下の通りである, 第 2 章で DMFB の概要とその合成手法, そして試料生成について述べる. 次に, 第 3 章では, 汚染問題を考慮した試料生成における DMFB 合成手法について提案する. 続いて, 第 4 章で提案手法の評価方法と実験結果について述べる. 最後に第 5 章にて, まとめと今後の課題について述べる.

2. Digital Microfluidic Biochip

本章では, 本論文で提案する手法の基礎となる Digital Microfluidic Biochip (DMFB) について説明する. まず DMFB のチップアーキテクチャについて説明し, 続いて DMFB の合成プロセスについて, 最後に DMFB のアプリケーションの 1 つである試料生成について説明する.

Vol.2017-ARC-225 No.21
Vol.2017-SLDM-179 No.21
Vol.2017-EMB-44 No.21
2017/3/9

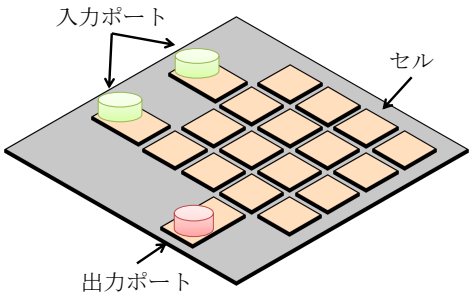

図 1　DMFB アーキテクチャ

2.1　DMFB アーキテクチャ

本論文では汎用の DMFB を扱う．ある 1 つの実験のみに特化した専用の DMFB とは違い, 汎用 DMFB とは実験に応じたチップの再設計, 再構成が可能な DMFB である．チップの再設計, 再構成はチップ製造後でも可能なので, 汎用 DMFB はさまざまなアプリケーションや状況に対応することを可能にしている．汎用 DMFB の基本構造を図 1 に示す．汎用 DMFB は二次元配列状に配置されたセルの集まりで構成されており, 各セルには電極が接続されている．DMFB 上では試薬は少量の液滴として扱われ, 各電極は入力された液滴を操作するために利用される．液滴は隣接セルの電極に電圧をかけることにより, 静電誘導現象と呼ばれる物理現象を発生させて移動する [4]．このように, 汎用 DMFB では全てのセルに電極が配線されているため, さまざまな実験に対応できるように設計されている．

2.2　DMFB 合成プロセス

DMFB 上で行う一連の実験の流れは, 図 2 (a) のようなアプリケーショングラフと呼ばれる有向グラフで表現される．アプリケーショングラフでは, 各ノードが実験作業 (オペレーション) を, 各エッジが液滴の移動経路を表している．DMFB の合成プロセスは, アプリケーショングラフを入力とし, 一般にスケジューリング (図 2 (a)), プレイスメント (図 2 (b)), ルーティング (図 2 (c)) の順で行われる [5]．各プロセスについて簡単に説明すると, まずスケジューリングにて, 各オペレーションの実行順序が決定される．次にプレイスメントにて各オペレーションがチップ上どの部分で実行されるかを決定する．また, このオペレーションが実行される場所はモジュールと呼ばれ, 汎用 DMFB ではあらかじめ同じ大きさのモジュールが等間隔に配置されている．最後にルーティングにて各液滴のモジュール間の移動経路を決定する．

2.2.1　ルーティング

本論文ではルーティングにおける問題を扱うため, 本項ではこのプロセスについて詳しく述べる．ルーティングでは, 各液滴のモジュール間の移動経路を決定する際に, 液滴制約と汚染問題という 2 つの制約を考慮する必要がある．液滴制約の概要を図 3 に示す．図 3 が示すとおり, 液滴と

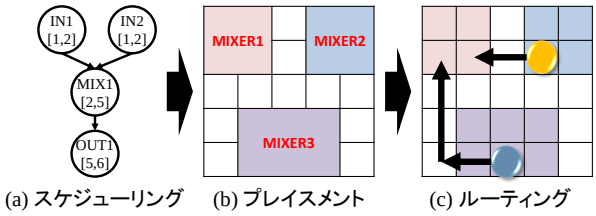

図 2　DMFB 合成プロセス

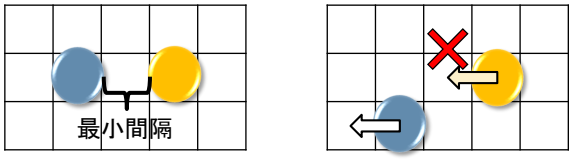

図 3　液滴制約

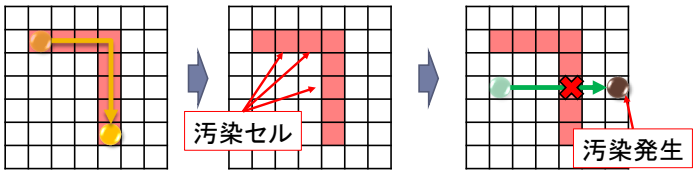

図 4　汚染問題

チップとの接触面は隣接セルから静電誘導現象を受けるために, セルから少しはみ出るよう液滴の量が調節されている．液滴制約とは, 液滴同士の接触を防ぐため, 液滴間に最低でも 1 セルの隙間を設ける制約である．一方汚染問題とは, 液滴が一度通ったセルは, 残留物によって汚染されるため使用できないという制約であり, その概要を図 4 に示す．この汚染セルの使用には洗浄のための液滴を別に入力し, 該当セルを洗浄する必要がある．ルーティングではこれらの制約を考慮した上で, 実験時間を削減するため, 移動経路の最小化が求められている．

前述したとおり, ルーティングでは液滴制約と汚染問題という 2 つの制約を考慮する必要がある．液滴制約は隣接しそうな液滴の移動を一時停止するなど, 簡単に対処が可能なため, 実行時間への影響は少ない．一方汚染問題の対処には, 洗浄液滴による洗浄を待つ必要があるなどの理由から, 対処には時間を要する．上記の理由により, ルーティングでは液滴の総経路長を削減する以上に, 汚染セルの使用を最小化することが重要である．

2.3　試料生成

本論文で提案する手法は, 試料生成に関するものなので本節にて詳しく説明する．試料生成とは, DMFB 上 で実行されるアプリケーションの 1 つで, ある特定の濃度を持った液滴群の生成を行うアプリケーションである．このアプリケーションは実際の実験の前段階として実行される場合が多く, 実験にかかる時間の 90%から 95%を占めているとされている [6]．

液滴の希釈方法として, DMFB 上で実行できるオペレー

Vol.2017-ARC-225 No.21
Vol.2017-SLDM-179 No.21
Vol.2017-EMB-44 No.21
2017/3/9

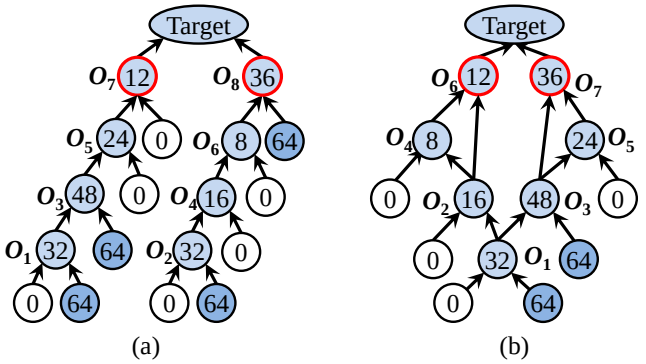

図 5　濃度 $\frac{12}{64}$, $\frac{36}{64}$ を生成する実験

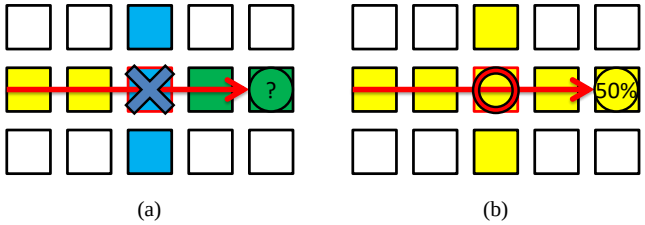

図 6　汚染問題における許容解

ションである混合 (Mix) と分割 (Split) が利用されている. 混合は 2 つの液滴を混ぜるオペレーションであり, 分割は 1 つの液滴を 2 つに分割するオペレーションである. 液滴の希釈作業にはこれら 2 つのオペレーションを組み合わせて実行される. DMFB では同じ量の液滴しか扱うことができないので, 各液滴の濃度を C_i $(0\% \leq Ci \leq 100\%)$ とすると, 一度の希釈結果は必ず $\frac{C_i+C_j}{2}$ となる. 試料生成では, この希釈オペレーションを複数回繰り返して目的の濃度を持った液滴の生成を行う.

試料生成の例を図 5 に示す. このアプリケーショングラフではノード内の数値は各液滴の濃度を表しており, 実行されるオペレーションは全て希釈オペレーションである. 図 5 (a), (b) は両方とも同じ濃度 $\frac{12}{64}$ と $\frac{36}{64}$ を生成する実験であるが, 図 5 (a) に比べ図 5 (b) では目的の濃度を生成するための中間濃度の液滴を利用することで, 無駄を削減している. このように, 試料生成では希釈オペレーションの順番や中間濃度を無駄なく使用することで, 使用する試料の量や実験時間の削減が可能である.

3. 試料生成における汚染問題を考慮した DMFB 合成手法

2.2.1 項で説明した汚染問題は, 試料生成のような実験では無視できる場合がある. 本論文では, この汚染問題に対する許容解を考慮した, 試料生成におけるルーティング手法を提案する. この許容解については, 3.1 節にて詳しく説明する. また, 提案手法を実現するための手法として, 整数計画ソルバーを用いた解法を 3.2 節にて, ヒューリスティックスによる解法を 3.3 節にて説明する.

3.1 試料生成における許容解

試料生成を実行する場合には, 汚染問題に対する許容解が存在する. この許容解とは, セルを汚染した液滴と汚染セルを利用する液滴の濃度が同じ場合, 汚染問題を無視できるといったものである. この許容解の概要を図 6 に示す. 図 6 (a) が示すとおり, 汚染問題において液滴が汚染セルを利用できない理由は, 液滴の残留物によるセルの汚染が原因である. しかし, 図 6 (b) のように, 残留物と通過する液滴の濃度が同じ場合, 液滴は汚染されずそのまま通過することができる. 試料生成では, 図 5 (b) のように, 液滴の再利用が頻繁に行われるため, 同じ濃度の液滴が大量に使用される. よって, 試料生成において液滴の濃度を考慮したルーティングを行えば, 許容解により汚染セル使用の削減が期待できる. 本論文では, 上記の許容解を考慮したルーティングを提案手法とし, 提案手法の実現方法として整数計画ソルバーを用いた解法とヒューリスティックスによる解法のそれぞれを提案する.

3.2 整数計画ソルバーを用いた解法

提案手法を実現する方法の 1 つとして, 整数計画ソルバーを利用した解法を提案する. 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り, この解法における許容解を考慮したルーティング問題に対して, 確実に最適解を得るために用いられる.

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. これらの目的関数と制約式は線形関数であり, また, 制約式では不等式が使用される場合もある. このままでは線形計画問題とされるが, 整数計画問題ではさらに, 目的関数と制約式内の全ての変数で, 整数が用いられている. 整数計画問題では, 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, 関数内で使用されている変数は全ての制約式の範囲内でなければならない.

本論文では整数計画ソルバーとして CPLEX を使用する. 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 を備えており [7], プログラムによる目的関数や制約式の自動生成が可能なソルバーである. 次項では, この API を利用した提案手法を整数計画問題へと定式化する方法を説明する.

3.2.1 問題定義

3.1 節で説明した許容解を考慮したルーティング問題を, 整数計画問題へと定式化するために, この問題を以下のように定義する.

Vol.2017-ARC-225 No.21
Vol.2017-SLDM-179 No.21
Vol.2017-EMB-44 No.21
2017/3/9

表 1　入力から得られる情報

入力	制約式内での記述
x 軸のセル数	C_wid
y 軸のセル数	C_len
モジュール番号	Mod_num
総モジュール数	$Total\_mod$
各液滴の番号	D
液滴の出発モジュール	$S\_mod[D]$
液滴の目的モジュール	$T\_mod[D]$
実験に使用されている各濃度を表す番号	Con_num
実験に使用されている各濃度の種類数	Max_Con

許容解を考慮したルーティング問題

目的　：汚染セルの使用数の最小化,
　　ただし許容解を考慮する
入力 1 ：x, y 軸のセル数
入力 2 ：各モジュールに関する情報 (位置や総数)
入力 3 ：各液滴の出発モジュールと目的モジュール
入力 4 ：濃度に関する情報
入力 5 ：全経路数
変数 1 ：各液滴の経路
条件 1 ：全ての経路は出発モジュールから
　　目的モジュールへと到達している

上記の目的, 入力, 変数, 条件全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計算できる.

ルーティング問題に必要な変数や制約式は, 前プロセスであるスケジューリングやプレイスメントの結果によって変化する. そこで本節の手法では, プログラムによる動的な目的関数, 変数, 制約式の生成を行う. 本節の手法では上記の問題をプログラムにて定式化した後, CPLEX にて汚染セルの使用が最小な経路の探索を行うことで, 許容解を考慮したルーティングを実現する. 定式化の手法は次項にて説明する.

3.2.2　定式化

本項では, 許容解を考慮したルーティング問題の定式化手法を説明する. プレイスメントへの入力となる, スケジューリングとプレイスメントの結果から得られる情報を表 1 に示す. これらの入力は前項における問題提起での入力 1 から 5 に相当する.

本節の手法では, 式 1 に示すような 3 次元変数配列 $Route$ を, 前項における問題定義内の変数 1 に相当する変数として利用する. 式 1 に示す通り, $Route$ は液滴番号, x 軸のセル数, y 軸のセル数による 3 次元変数配列であり, 各液滴による各セルの使用状況を表現する変数である. また, $Route[D]$ から参照できる 2 次元配列は, 各液滴の経路を表した隣接行列となる. この $Route$ は後述する目的関数において, 整数計画ソルバーが選択する経路候補を評価するために利用される.

$$Route[D][x][y] = \begin{cases} 1 \text{ セル } (x,y) \text{ を使用する} \\ 0 \text{ セル } (x,y) \text{ を使用しない} \end{cases} \quad (1)$$

整数計画ソルバーが計算する際, 変数 $Route$ に各液滴の経路情報が格納されるように, 制約条件を設定する. また, この制約条件は問題定義内の条件 1 に相当するが, 本節の手法では複数の制約条件を合わせて条件 1 を定式化する. まず, 1 つ目の制約条件を式 2 に示す.

$$\forall D((\sum_{x=0}^{C\_wid} \sum_{y=0}^{C\_len} Route[D][x][y]) \geq Min\_length[S\_mod[D]][T\_mod[D]]) \quad (2)$$

式 2 は, 『各経路の長さは必ず, モジュール間の最短距離以上でなければならない』といった条件を表している. この条件により, 各 $Route[D]$ では, 最低でもモジュール間の最短距離以上のセルが使用されるように強制される. また, 式内の $Min\_length[s][g]$ は, モジュール番号 s から g の最短距離を表している. 次に 1 個目の条件式にて決定される経路が, 始点, 終点となるセル間の経路になるように, 制約式を設定する. また, 各経路における始点, 終点はそれぞれ, 各液滴がスタートするモジュールと目的地となるモジュールが配置されているセルとなる. 始点, 終点を設定する制約条件をそれぞれ式 4, 式 4 に示す.

$$\forall D(Module\_cell[D][S\_mod[D]] = 1) \quad (3)$$

$$\forall D(Module\_cell[D][T\_mod[D]] = 1) \quad (4)$$

これらの式で使われている $Module\_cell[D][Mod\_num]$ は, モジュール Mod_num が配置されているセルの, $Route[D][x][y]$ の総和である. これらの式では, 各モジュールにあたるセルが 1 つのみ利用されるよう制約することで, 各経路の始点, 終点を設定している. 最後に, $Route[D]$ に示される各経路が, モジュール間を接続する 1 本の経路として設定されるよう, 式 5 に示す制約式を設定する.

$$\exists D \exists x \exists y (Route[D][x][y] = 1 \rightarrow Surround[D][x][y] = 2) \quad (5)$$

式 5 の制約式は, 『あるセルが使用されているのなら, 必ず隣接するセルも利用されている』という制約である. この制約にて液滴が上下左右の 4 方向にしか進めないように制限している. また, 式 5 内の配列 $Surround[D][x][y]$ は, 式 6 に示すような変数配列であり, あるセル (x,y) の最大 4 つの近接セルの総和を保持している. この制約式は, モジュール以外のセルにのみ適応する.

$$Surround[D][x][y] = Route[D][x+1][y] + Route[D][x-1][y] + Route[D][x][y+1] + Route[D][x][y-1] \quad (6)$$

このように, 本節の手法では式 2, 式 4, 式 4, 式 5 の 4 つの制約条件を組み合わせ, 問題定義における条件 1 を定式化している.

目的関数には, 汚染セルの使用数を計算する関数を設定する. また, 3.1 節にて説明した許容解を考慮するために, 各濃度のセルの使用状況を表す変数 $Overlap[Con\_num][x][y]$ を用意する. 例えば, ある濃度を持つ複数の経路にて, セル (x, y) が一度でも利用されていれば $Overlap[Con\_num][x][y] = 1$ となる. $Overlap[Con\_num][x][y]$ を用いることで, 目的関数は式 7 のように定式化できる. 式 7 では, $Overlap$ の総和を計算することで, 同じ濃度同士の汚染セル使用を許容した, 汚染セルの使用数を計算している.

$$minimize \sum_{c=1}^{Max\_Con} \sum_{x=1}^{C\_wid} \sum_{y=1}^{C\_len} Overlap[c][x][y]) \quad (7)$$

本節の手法では最後に, 式 1 から式 7 までの式を CPLEX に設定することで, 最適解を計算することができる. CPLEX で計算した最適解は, 計算終了後に目的関数と $Route$ の値を参照することで取得可能である. 本節の手法の場合, 目的関数が許容解を考慮した汚染セルの使用数と $Route[D]$ の近接行列が各液滴の移動経路となり, これらの値を出力することでルーティング完了となる.

3.3 ヒューリスティックスによる解法

前節にて提案した, 整数計画ソルバーによる解法を用いて実験を行った結果, この手法では確実に最適解が得られる反面, 小さい実験でも計算に時間がかかり, 大きな実験には不向きであることが判明した. 1 つの実験に時間をかけられるなら問題ないが, 汎用 DMFB の利点である柔軟な対応力を生かすためには, 計算に時間がかからない手法が必要である. そこで, 本節では整数計画ソルバーを利用する以外の解法として, ヒューリスティックスによる解法を提案する.

3.3.1 アルゴリズム

本項では, 提案手法を実現するヒューリスティックスに使用されるアルゴリズムとして, 以下のステップで実行されるアルゴリズムを説明する.

(1) 各液滴について, 最小コストの経路を探索
(2) 各経路で使用されたセルのコストを更新
(3) 一定回数, ステップ 1, 2 を繰り返す
(4) 繰り返しの中で, 最も汚染セルの使用数が少なかった経路を, ルーティング結果として出力

本アルゴリズムではまずステップ 1 にて, 各液滴の最小コストの経路を探索する. この場合のコストとは, 経路として通過する各セルに設定された値の総和である. 各セルに設定されたコストを今後は $Cost[x][y]$ と表記する. また, 括弧内の x, y はそのセルの x, y 座標を示しており, 全ての $Cost[x][y]$ は 1 で初期化されている. この値は, ステップ 1 にてそのセルが経路として使用されれば, ステップ 2 で更新される. 値の更新は, 式 8 に示されている計算式が用いられる. ステップ 1, 2 は複数回繰り返し実行されるので, $Cost[x][y]$ の値はステップ 1 で経路として使用された回数だけ大きくなる.

$$Cost[x][y] = Cost[x][y] + (\text{このセルを使用している経路数} \times \text{更新回数}) \quad (8)$$

前述した通り, ステップ 1 では最小コストの経路を探索するので, $Cost[x][y]$ が高いセルは, 次第に選ばれにくくなる. $Cost[x][y]$ が高いセルというのは, 複数の経路でよく使用されているということなので, ステップ 1 で探索される経路は, 繰り返しが進むにつれ, 汚染セルの使用が少ない経路を探索するようになる. 式 8 で更新回数が使用されているのは, 後に探索される経路の方が, 汚染セルの使用が少ない経路を選択している可能性が高いためである. このように, 本アルゴリズムでは, 繰り返し各セルのコストを更新することによって, 汚染セルの使用が発生しにくいセルを発見すると同時に, 経路探索を行っている.

$Cost[x][y]$ には全ての濃度のコストが合算されているが, 3.1 節にて説明した許容解が考慮されておらず, 同じ濃度の分だけ値が高くなってしまっている. そこで, 本アルゴリズムでは, 各セルの同じ濃度の上昇分を別の変数に保持し, ステップ 1 で計算する際に $Cost[x][y]$ から減算するようにしている. このような減算処理を行うことで, 同じ濃度が使用したセルは他のセルと比べて $Cost[x][y]$ の値が下がるので, 同じ濃度同士で経路を共有しやすくしている.

本アルゴリズムではステップ 4 として最後に, 繰り返しの中で一番汚染セル使用数が少なかったものを結果として出力している. また, 繰り返しの回数は $Cost[x][y]$ の値が十分に更新されるだけの回数が行われればよい. 次章にて行う実験では, 全てのケースにおいて 30 回以降, 結果が更新されることはなかったため, 更新回数は 30 回としている.

4. 実験結果

第 3 章では, 試料生成における許容解を考慮したルーティング手法として, 整数計画ソルバーによる解法と, ヒューリスティックスによる解法の 2 種類の手法を提案した. 本章では, これら 2 種類の手法を用いてルーティングを行った結果と, その考察を示す.

4.1 評価方法

2.2.1 項で説明した通り, ルーティングでは実行時間削減のために, 汚染セルの使用を削減する必要がある. そこで, 3 章にて説明した 2 つの解法を用いて, 同じ実験のルーティングを行い, 汚染セル使用数の比較を行った. 評価に用い

Vol.2017-ARC-225 No.21
Vol.2017-SLDM-179 No.21
Vol.2017-EMB-44 No.21
2017/3/9

表 2　実験環境

OS	Ubuntu 14.04.1 LTS
CPU	Intel(R) Core(TM) i7-4790K 4.00GHz
メモリ	32GB
使用言語	C++

表 3　評価に使用した実験の詳細

実験	チップサイズ	モジュール数	経路数
Test1	13 × 13	4	23
Test2	13 × 13	4	32
Test3	13 × 25	8	43
Test4	13 × 25	8	47
Test5	13 × 25	8	57
Test6	13 × 25	8	85

表 4　実験結果

実験	汚染セル使用数		実験時間 (秒)	
	ILP	Heuristics	ILP	Heuristics
Test1	28	43	1026	2
Test2	44	49	12438	2
Test3	86	103	100093	30
Test4	146	147	126896	26
Test5	330	250	200000 (時間切れ)	43
Test6	1093	393	200000 (時間切れ)	81

る実験は, [8] の手法を用いて, ランダムに生成したものを複数用意した. 評価に使用した実験の詳細を表 3 に示す. また, 実験は両手法共に表 2 に示す環境にて行った.

4.2　実験結果と考察

整数計画ソルバーによる手法と, ヒューリスティックスによる両方の実験結果を表 4 に示す. 表 4 では, 整数計画ソルバーによる結果は ILP の列に, ヒューリスティックスによる実験結果は Heuristics の列に示されている. また, Test5, 6 の整数計画ソルバーの計算時間に, (時間切れ) の記載があるのは, 変数と条件式が多く, 結果の計算が出来なかったためである. これらの実験では, 計算時間を 200000 秒に制限し, この時間内で得られた結果の中から一番良い結果を, 表 4 に記載している また, Test5, Test6 以外の結果では, 整数計画ソルバーは最適解を出力している.

表 4 より, 大きな実験を計算する場合を除き, 計算に時間がかかってもいいのなら整数計画ソルバーによる解法, 計算を急ぐのであればヒューリスティックスによる解法を利用すればいいことが分かる. ただし, ヒューリスティックスによる解法には, 相性の悪い実験が存在することを留意しなくてはならない. 提案したヒューリスティックスでは, 同じ濃度の経路数分セルのコストが下がるので, 自然と経路数の多い濃度の経路から経路が確定していく. しかし経路数が同じ濃度が複数存在する場合, 各濃度間でコストの差が生まれず, 上手く経路探索が出来ない場合がある. 今回の実験結果では Test1 がこのケースにあたり, 実際に Test1 の実験では, 5 種類中 4 種類の濃度が同じ経路数であった.

5.　おわりに

本研究では，DMFB の合成プロセスの 1 つであるルーティングにおいて, 試料生成という実験でのみ存在する, 汚染問題に対する許容解を考慮した合成手法を提案した. さらに, この提案手法を, 整数計画ソルバーによる解法と, ヒューリスティックスによる解法の 2 つのアプローチで実現した. 実験の結果, このヒューリスティックスによるアプローチでは, 整数計画ソルバーによる最適解に, 最大で 99%近い結果を出すことに成功した.

今後の課題としては, 整数計画ソルバーによる手法では, より少ない変数や制約式で問題の定式化を行うことが課題となる. 一方ヒューリスティックスによる解法では, 表 3 における, Test1 のような実験でも最適解に近い結果が計算できるような手法を考案することが課題となる.

参考文献

[1] T. -Y. Ho, K. Chakrabarty and P.Pop.: Digital microfluidic biochips: Recent research and emerging challenges, *CODES+ISSS*, pp. 335–343 (2011).
[2] Huang, T. W., Lin, C. H. and Ho, T. Y.: A Contamination Aware Droplet Routing Algorithm for the Synthesis of Digital Microfluidic Biochips, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 29, No. 11, pp. 1682–1695 (online), DOI: 10.1109/TCAD.2010.2062770 (2010).
[3] Zhao, Y. and Chakrabarty, K.: Cross-Contamination Avoidance for Droplet Routing in Digital Microfluidic Biochips, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 31, No. 6, pp. 817–830 (online), DOI: 10.1109/TCAD.2012.2183369 (2012).
[4] Pollack, M. G., Shenderov, A. D. and Fair, R. B.: Electrowetting-based actuation of droplets for integrated microfluidics, *Lab Chip*, Vol. 2, pp. 96–101 (online), DOI: 10.1039/B110474H (2002).
[5] Grissom, D., O'Neal, K., Preciado, B., Patel, H., Doherty, R., Liao, N. and Brisk, P.: A digital microfluidic biochip synthesis framework, *2012 IEEE/IFIP 20th International Conference on VLSI and System-on-Chip (VLSI-SoC)*, pp. 177–182 (online), DOI: 10.1109/VLSI-SoC.2012.7332097 (2012).
[6] Hsieh, Y. L., Ho, T. Y. and Chakrabarty, K.: A Reagent-Saving Mixing Algorithm for Preparing Multiple-Target Biochemical Samples Using Digital Microfluidics, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 31, No. 11, pp. 1656–1669 (online), DOI: 10.1109/TCAD.2012.2202396 (2012).
[7] IBM: IBM ILOG. IBM ILOG CPLEX V12.1 User's Manual for CPLEX (2009).
[8] Dinh, T. A., Yamashita, S. and Ho, T. Y.: A network-flow-based optimal sample preparation algorithm for digital microfluidic biochips, *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 225–230 (online), DOI: 10.1109/ASPDAC.2014.6742894 (2014).