Chapter 2
Cross-Contamination Avoidance for Droplet Routing

Recent advances in droplet-based digital microfluidics have enabled biochip devices for DNA sequencing, immunoassays, clinical chemistry, and protein crystallization. Since cross-contamination between droplets of different biomolecules can lead to erroneous outcomes for bioassays, the avoidance of cross-contamination during droplet routing is a key design challenge for biochips.

In this chapter, we focus on droplet-routing methods to avoid cross-contamination. First, we present motivation and related prior work on droplet routing and cross-contamination. Next we describe a droplet-routing method that avoids cross-contamination in the optimization of droplet flow paths. The proposed approach targets disjoint droplet routes and synchronizes wash-droplet routing with functional droplet routing, in order to reduce the duration of droplet routing while avoid the cross-contamination between different droplet routes. In order to avoid cross-contamination between successive routing steps, an optimization technique is used to minimize the number of wash operations that must be used between successive routing steps.

The notation used in this chapter is listed in Table 2.1.

2.1 Related Prior Work

In order to solve the droplet-routing problem for digital microfluidics, a number of techniques have been proposed [1–7]. In [4], a network-flow-based routing algorithm that can concurrently route a set of non-interfering nets for the droplet routing problem on biochips is proposed. In [2], the proposed droplet router transports a droplet with higher bypassibility, a strategy that is less likely to block the movement of the others. In [6], a droplet-routing-aware automated synthesis tool for microfluidic biochips is presented. However, it does not generate any specific droplet route during the synthesis.

Y. Zhao and K. Chakrabarty, *Design and Testing of Digital Microfluidic Biochips*, DOI: 10.1007/978-1-4614-0370-8_2,

Table 2.1 Notation used in this chapter

D_i	The i-th functional droplet
W_i	The i-th wash droplet
S_i	The i-th cross-contamination site
P_i	Routing path for the i-th droplet
Td	Maximum droplet-routing time within one subproblem
$M \times N$	Size of microfluidic array
(t_i,s_i)	Source electrode and sink (destination) electrode for the i-th droplet
(x_i, y_i)	Coordinate of the i-th electrode
K	Number of droplet-routing subproblems for a bioassay
T_i	Maximum transportation time needed among the droplet routes in subproblem i if the wash operation is not performed after droplet routing in subproblem $(i-1)$
T_i^*	Maximum transportation time needed in subproblem i if the wash operation is performed after the droplet routing in subproblem $(i-1)$
Tw_i	The time needed for the wash operation after droplet routing in subproblem $(i-1)$
x_i	A 0-1 variable represents whether a wash operation is performed between subproblems $(i-1)$ and i
N_{cs}	Number of cross-contamination sites
N_{cell}	Number of cells used by routing of droplets
N_{wash}	Number of consumed wash droplets
T_{rw}	Maximum droplet-transportation time for droplets with wash steps
T_{r}	Maximum droplet-transportation time for droplets without wash steps

In [3], a systematic droplet routing method is integrated with biochip synthesis. The proposed approach minimizes the number of cells used for droplet routing, while satisfying constraints imposed by throughput considerations and fluidic properties. However, since the routes in microfluidic biochips are viewed as *virtual routes*, droplet routes may intersect or overlap with each other during different time intervals. Therefore, cross-contamination between droplets may happen at the overlap sites.

In many on-chip adaptation of biomedical assays, liquids that contain large molecules such as proteins are transported on the microfluidic platform [8]. However, proteins cannot be transported easily since they tend to adsorb irreversibly to hydrophobic surfaces and contaminate them [8, 9]. Silicone oil with its low surface tension and spreading property has been advocated as a filler medium for protein assays to prevent contamination [8]. However, it has also been reported that the use of silicone oil alone is not sufficient for many types of proteins [10].

Wash operations are used to avoid cross-contamination between different droplet routes [1, 11, 12]. Wash droplets traverse the cross-contamination sites and clean residue between successive functional droplet routing. In [11], a washing step is necessary after each polypyrrole synthesis step to avoid cross-contamination in the

DNA biochip. Wash droplets are also utilized in the clinical application of diagnosis for Huntington's disease [12]. A contamination-aware droplet routing algorithm is proposed in [1], where a minimum cost circulation algorithm and look-ahead prediction are used to optimize wash-droplet routing. The routing algorithm in [1] iteratively compacts the routing paths of wash droplets with previously compacted routing paths of functional droplets, without optimizing the order in which multiple cross-contamination sites are considered. Therefore, the droplet-transportation time with washing steps may not be minimized.

In [7], to cope with cross contamination in pin-constrained digital microfluidic biochips, the authors proposed early-crossing minimization algorithms during placement and a systematic wash droplet scheduling and routing algorithm, in order to minimize the number of control pins introduced by the washing steps. One extra control pin is needed to move wash droplets.

However, the wash-droplet scheduling and routing algorithm in [7] is based on an unrealistic assumption. It is assumed that droplet transportation does not occur concurrently with bioassay operations such as mixing. However, in practice, droplet routing and mixing are often carried out simultaneously. In addition, in [7], the same group of wash droplets are used to clean the residue in multiple bioassay and routing stages. However, since excessive residue within the wash droplet can degrade the ability of cleaning extra residue on electrodes, a specific wash droplet cannot be used for multiple wash operations.

2.2 Disjoint Routes for Cross-Contamination Avoidance

In this section, we describe a droplet-routing method that avoids cross-contamination in the optimization of droplet flow paths. This approach targets disjoint droplet routes to avoid the overlap between routing paths.

2.2.1 Problem Formulation and Constraints

In this subsection, we first present the basic concepts underlying the proposed routing method. Then we introduce the constraints in droplet routing. Finally, we decompose the droplet-routing problem into a series of subproblems.

Problem Formulation

Given a schedule of bioassay operations (derived from architectural-level synthesis) and the locations of these modules on the biochip floorplan (derived from module placement [13]), routing determines the paths for droplet transportation using the available cells in the microfluidic array. Droplets are transported along these routes between modules, or between modules and fluidic I/O ports (e.g., on-chip reservoirs and sensors).

The fluidic ports on the boundary of microfluidic modules are referred to as *pins*. The droplet routes between pins of different modules or on-chip reservoirs are referred to as *nets*. Therefore, a fluidic route on which a single droplet is transported between two terminals (i.e., one source and one sink) can be modeled as a 2-pin net. Two droplets from different terminals are often transported to one common module (i.e., mixer or diluter) for mixing or diluting. To allow droplet mixing simultaneously during their transportation, we model such fluidic routes using 3-pin nets.

Cross-contamination is likely to occur when multiple droplet routes intersect or overlap with each other. At the intersection site of two droplet routes, a droplet that arrives at a later clock cycle can be contaminated by the residue left behind by another droplet that passed through at an earlier clock cycle. The more cells that two droplet-routes share, the higher is the likelihood of cross-contamination.

Our first goal is to avoid cross-contamination between different droplet routes. The second goal is to minimize the time needed for droplet routing. Therefore, we focus on disjoint droplet routes and the minimization of the total path length over all routes, where path length is measured by the number of cells in the path from the source to the sink.

In a set of disjoint routes, a droplet route does not share any cell in its path with each of the other droplet routes in that set. Such routes eliminate the possibility of a droplet being transported via a cell when another droplet has already passed through it in an earlier time interval. Therefore, disjoint routes avoid cross-contamination between different droplets. The minimization of the length of the droplet routes leads to a reduction in the droplet-transportation time. It also frees up more spare cells for parallel fluidic operations and fault tolerance [3].

Constraints: Although disjoint droplet routes avoid cross-contamination between different droplets, some droplets may inadvertently mix if they are routed too close simultaneously during their transportation. Therefore, during droplet routing, a minimum spacing between droplets must be maintained to avoid unintended mixing, except for the case that droplet merging is desired (i.e., in 3-pin nets). A segregation region is added to wrap around the functional region of microfluidic modules to avoid conflicts between droplet routes and assay operations that are scheduled at the same time. Moreover, fluidic constraint rules in [3] need to be satisfied in order to avoid undesirable mixing.

Another constraint in droplet routing is given by an upper limit on the number of electrodes for each droplet route. Since droplets that contain large molecules such as proteins tend to adsorb on electrodes and contaminate them [8], which leads to the loss of chemical materials within droplets and the resulting erroneous assay outcomes, the length of the droplet route should be low and cannot exceed an upper limit.

Problem Decomposition: Since a digital microfluidic array can be reconfigured dynamically at run-time, a series of 2-D placement configurations of modules in different time spans are obtained in the module placement phase [13]. Therefore, the droplet routing is decomposed into a series of subproblems. In each subproblem, the nets to be routed between the sources and the sinks are first determined. The microfluidic modules that are active at the time when droplets are transported are considered as *obstacles*. We obtain a complete droplet-routing solution by solving these subproblems sequentially.

2.2.2 Routing Method

In this subsection, we solve the problem of droplet routing to avoid cross-contamina tion within one subproblem. We first map the problem of finding feasible disjoint routes in the 2-D microfluidic array to the problem of finding disjoint paths (vertex-disjoint or edge-disjoint) in a graph. We next describe a droplet-routing algorithm to determine disjoint droplet routes and minimize the number of cells used for droplet routing, while satisfying the constraints.

Graph Model and Disjoint Routing

The problem of finding feasible disjoint routes for 2-pin or 3-pin nets in a 2-D microfluidic array can be directly mapped to the problem of finding disjoint paths (vertex-disjoint or edge-disjoint) for pairs of vertices in a graph [14, 15]. We consider a planar undirected graph $G = (V, E)$, where V is the vertex set and E is the edge set. Each vertex in the graph represents an electrode in the 2-D microfluidic array, and there is an edge between two vertices if their corresponding electrodes are adjacent. The pins in a 2-pin net are represented by a pair of vertices (t_i,s_i) in the graph. The pins in a 3-pin net are represented by a set of three vertices $(t_{1,i},t_{2,i},s_i)$, where $t_{1,i}$ and $t_{2,i}$ represent the two source pins, and s_i represents the destination pin. The 3-pin net $(t_{1,i},t_{2,i},s_i)$ is handled as follows: First, generate the routing path for two-pin net $(t_{1,i},t_{2,i})$; Next for each electrode in the routing path of net $(t_{1,i},t_{2,i})$, we calculate the Manhattan length (in terms of number of electrodes) between this electrode and the node s_i; The electrode with the minimum Manhattan length will be selected as the mixing point m_i. In this way, the 3-pin net $(t_{1,i},t_{2,i},s_i)$ will be handled as three 2-pin nets: $(t_{1,i},m_i)$, $(t_{2,i},m_i)$, and (m_i,s_i). The route for a net is represented by a path consisting of a set of successive edges in the graph, where the endpoints of a path represent the corresponding pins. Each edge in the path denotes the fact that the electrodes represented by the two endpoints of the edge are adjacent in the droplet route.

Consider a set of disjoint routes where a droplet route does not share any cell with each of the other droplet routes. The corresponding paths in the graph are mutually *vertex-disjoint* since a path does not share any vertex with each of the other paths in the set. Similarly, in a set of disjoint routes where a droplet route does not share any pair of adjacent cells with each of the other droplet routes, their corresponding paths in the graph are mutually *edge-disjoint* since a path does not share any edge with each of the other paths in the set.

Since a 3-pin net can be treated as 2-pin nets, henceforth we only consider 2-pin nets for disjoint routing. In the 2-D microfluidic array, given a set of n 2-pin nets with corresponding pins (t_1, s_1), (t_2, s_2), ..., (t_n, s_n), the problem of finding feasible disjoint routes for these nets is equivalent to the problem of finding mutually vertex-disjoint or edge-disjoint paths in G, such that the endpoints of each path in G represent the corresponding pins of each net. Given $G = (V, E)$ and the vertex pairs (t_1, s_1), (t_2, s_2), ..., (t_n, s_n), the problem of determining whether mutually vertex-disjoint paths $P_1, P_2, \ldots, P_n$ exist such that P_i has endpoints t_i and s_i, is NP-complete [14]. Furthermore, the problem of determining whether mutually edge-disjoint paths exist is also NP-complete, even if the graph G is a 2-D mesh [15]. Therefore, we use a heuristic approach in this work. Furthermore, it is often difficult to find vertex-disjoint paths in the underlying graph model to solve the droplet-routing problem; such paths might not exist for a given set of nets. It is therefore more practical to relax the overlap restriction and search for edge-disjoint routes in such cases. For edge-disjoint routes, we only need to wash those individual electrodes at the intersections, instead of washing a large number of consecutive electrodes in the routing path shared by multiple droplets. In this way, the number of electrodes that need to be washed is reduced using edge-disjoint routes. Therefore, edge-disjoint routes lead to a reduction in the number of array sites that need to be washed.

Routing Method for Disjoint Routes

We next present a droplet-routing algorithm based on disjoint routes that minimizes route lengths and avoids cross-contamination within a subproblem. The input to the algorithm is a list of nets to be routed in the subproblem, and the output is a set of vertex-disjoint (preferred) or edge-disjoint (as a design compromise) routes with minimized lengths, subject to the constraints.

Within a subproblem, the Lee algorithm, a popular technique used in grid routing [16], can obtain a single droplet route for each net. The Lee algorithm is guaranteed to find the shortest path between two pins in a two-pin net. For 3-pin net, we use the modified Lee algorithm from [3] to obtain a feasible route connecting these three pins. Note that the interconnection obtained by the modified Lee algorithm from [3] is not guaranteed to be of minimum length. Nevertheless, it is a desirable route in practice, allowing concurrent mixing during transportation. However, the Lee algorithm does not avoid cross contamination.

The microfluidic modules that are active in this subproblem are considered as obstacles. The individual net in this subproblem is routed using the modified Lee algorithm sequentially. After any net has been routed, the cells occupied by its path are marked as obstacles for the unrouted nets. Therefore, the latter route is disjoint with respect to all the previous routes. However, the routing order in the subproblem influences the routability of all the nets. Even if each of the nets is individually routable, the routing order may prevent successful completion of routing for all the nets.

Therefore, we modify the net-routing ordering method proposed in [16] to obtain an optimized order for the routing of n nets in a subproblem. We first define the bounding box of a net. Assuming that two pins of a 2-pin net p are p_1 and p_2 with coordinates (x_1, y_1) and (x_2, y_2) in a 2-D microfluidic array, the coordinates of the four vertices of its corresponding bounding box are (x_1, y_1), (x_1, y_2), (x_2, y_1) and (x_2, y_2). Next we define $pin(p)$ to be the number of pins (for other nets to be routed in the current subproblem) within the bounding box of net p. Let $Xrange(p) = |x_1 - x_2|$ and $Yrange(p) = |y_1 - y_2|$. The bounding box for a 3-pin net is defined in a similar manner. The priority number for a net p is given by the following equation, where A is a user defined parameter. Here the parameter A is set to 1. The larger the priority number, the lower the priority of the corresponding net in the net-routing order.

$$priority(p) = pin(p) + A \cdot \max\{Xrange(p), Yrange(p)\}.$$

There are two stages in the net-routing process. In the first stage, we apply the modified Lee algorithm to each net sequentially in the net-routing order. The cells occupied by the routed paths are marked as obstacles for the unrouted nets. Each route thus obtained needs to pass the "route length constraint check" (RLCC). In addition, each route also needs to go through the "fluidic constraint rule check" (FCRC) with previously acquired routes according to the net-routing order. In [3], three rules are presented for fluidic constraints. They specify the spacing between droplets during the routing, and these rules must be obeyed to avoid inadvertent mixing. These rules are checked for all the droplets in each time slot during routing. If the above rules are violated, modification rules described in [3] are used to adjust the transportation timing of the droplets, in order to override the violation. For example, assume two droplets D_i and D_j are moved on the array. At a specific time, if the activated cell for droplet D_j is adjacent to D_i, D_j will move to the adjacent electrode of D_i and mix inadvertently with D_i. In this case, the modification rule is used to make D_j stall at current position and not move to the adjacent electrode, until D_i leaves current position. Therefore, an inadvertent mixing can be avoided. Using the modification rules in [3], the violation of fluidic constraints during droplet routing can be avoided without any modification of routing paths, therefore reduce the complexity of the routing algorithm.

The shortest route, i.e., one that uses the minimum number of cells, is selected as the output for each net. If all the shortest routes for some net do not satisfy the route length constraint, or violate the fluidic constraint and fail the droplet-motion modification, we put this net to a conflict list in order to resolve it at the end. Note that all routes obtained in this stage are not only edge-disjoint, but also vertex-disjoint (e.g., no two droplet routes share the same cell).

In the second stage, for each net in the conflict list, we execute the modified Lee algorithm to generate several shortest routes without considering the previously routed paths in the first stage as obstacles. From these shortest routes, we select a route that has no common pair of adjacent cells with the previously obtained routes as the output. Therefore, the resulting route may intersect, but still be edge-disjoint, with the previously obtained routes. As before, this route also needs to pass both RLCC and FCRC constraints.

At this stage, we have obtained disjoint routes for all the nets. When no feasible droplet routes can be generated in the disjoint routing method, the placement refinement method in [6] is used to re-synthesize the bioassay to improve routability. The droplet-routability of a route between two modules is quantified in terms of the length, measured by the number of electrodes in the droplet transportation path. The method of [6] is utilized to re-synthesize the bioassay, in order to obtain synthesis results with high routability values, which leads to the module placement where feasible droplet routes can be generated.

2.3 Synchronization of Washing Operations with Droplet Routing

In this section, we integrate washing steps into droplet-routing steps for the target bioassay to alleviate the cross-contamination problem. We refer to the routing of sample and reagent droplets as *functional-droplet routing*. Wash droplets are manipulated to traverse all the cross-contamination sites and clean the residue left behind by functional droplets. The routing of wash droplets is synchronized with functional droplet-routing. This approach leads to reduced droplet-transportation time.

Within one subproblem, there exist multiple sites where different droplet routes intersect with each other. We first address the problem of synchronizing wash-droplet routing with functional-droplet routing for one cross-contamination site, and extend the solution to synchronization for multiple cross-contamination sites.

2.3.1 One Cross-Contamination Site

The proposed synchronization of wash-droplet routing with functional-droplet routing for one cross-contamination site includes two steps. In the first step, we generate the route for the wash droplet (or multiple wash droplets) to traverse the cross-contamination site. This droplet route consists of two sub-routes. The first sub-route connects the dispensing reservoir for wash droplets (source) to the cross-contamination site (sink), and the second sub-route connects the cross-contamination site (source) to a waste reservoir (sink). The modified Lee algorithm from [3] is utilized to generate two sub-routes.

For one cross-contamination site, the routing of the wash droplet (or multiple wash droplets) is synchronized with the routing of two functional droplets. These droplets start from the corresponding sources at the same time and are transported along the computed droplet routes via the cross-contamination site towards the sinks. The arrival time of the wash droplet at the cross-contamination site depends on the length of the first sub-route. Note that the arrival time of a droplet from the source to the sink is calculated based on the length of the corresponding route. If a droplet is transported across one electrode per clock cycle, the arrival time of the droplet is equal to the number of electrodes in the route. The wash droplet (or multiple wash

Table 2.2 Adjustment of droplet-arrival order

Droplet arrival order			
D_1	D_2	W	Adjustment
1	2	3	D_2 is made to arrive after W traverses the site
1	3	2	Maintain this order
3	1	2	Maintain this order
3	2	1	W is made to arrive after D_2 traverses the site
2	1	3	D_1 is made to arrive after W traverses the site
2	3	1	W is made to arrive after D_1 traverses the site

droplets) will reach the site at an appropriate time, between the arrival times of the two functional droplets.

Therefore, in the second step, we adjust the arrival order of the wash and functional droplets, to ensure that the wash (or multiple wash droplets) arrives at the cross-contamination site later than one functional droplet, but earlier than the other. Without loss of generality, here we consider one wash droplet for a cross-contamination site. For a cross-contamination site S where the route of droplet D_1 intersects with the route of droplet D_2, we first estimate the arrival order of D_1, D_2, and the wash droplet W based on the lengths of droplet routes connecting their sources to the cross-contamination site, respectively. Next we adjust the arrival order of these three droplets according to Table 2.2. For example, if D_1, D_2 and W arrive at the cross-contamination site serially, we delay the arrival time of D_2 such that W traverses the site before D_2 arrives. In this way, the residue left behind by D_1 at an earlier clock cycle will be cleaned by the wash droplet W before D_2 arrives. Note that the droplet whose arrival time should be postponed is temporarily stored (held) in an on-chip storage unit. After the adjustment for one cross-contamination site is carried out, we record the updated time when droplets D_1, D_2 and W traverse this site. Note that if a droplet has to be stored in an on-chip storage unit, the total transportation time for this droplet is the sum of the droplet-routing time and the duration for the on-chip storage. Therefore, the duration for the on-chip storage of this droplet should be minimized.

Figure 2.1 shows the synchronization of wash-droplet routing with functional-droplet routing for one cross-contamination site. As shown in Fig. 2.1a, the droplet routes of two functional droplets D_1 and D_2 intersect at the cross-contamination site S. A wash droplet W is dispensed from the wash reservoir to clean the residue at S. The wash-droplet route consists of two sub-routes. The first sub-route connects the wash reservoir to S, and the second sub-route connects S to the waste reservoir. We estimate the arrival order of D_1, D_2, and W at the cross-contamination site S assuming that all the droplets start moving at the same time. From Fig. 2.1a, based on the lengths of droplet routes connecting their sources to the cross-contamination site, we calculate that D_1 arrives at S at clock

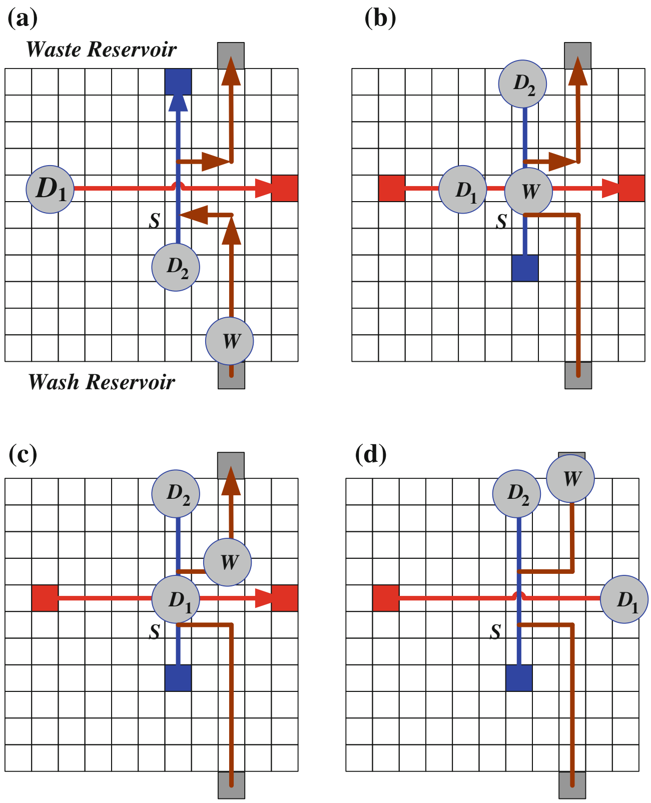

Fig. 2.1 Synchronization for one cross-contamination site: **a** wash-droplet routes and functional-droplet routes; **b** snapshot at clock cycle 9; **c** snapshot at clock cycle 13; **d** snapshot at clock cycle 17

cycle 5, D_2 arrives at S at clock cycle 3, and W arrives at S at clock cycle 9. According to Table 2.2, we should delay the arrival time of D_1 such that W traverses the site before D_1 arrives. As shown in Fig. 2.1b, at clock cycle 9, D_2 has arrived at the sink node, and W arrives at the cross-contamination site S to clean the residue left behind by D_2. D_1 is stored in the on-chip storage unit, instead of being transported along its droplet route. There is a one-electrode spacing between D_1 and W in order to avoid undesirable mixing. In Fig. 2.1c, at clock cycle 13, W has left the cross-contamination site S and is transported to the waste reservoir, and D_1 arrives at S that has been cleaned. In Fig. 2.1d, at clock cycle 17, three droplets D_1, D_2 and W have arrived at the corresponding destinations.

Therefore, the maximum droplet-transportation time is 17 clock cycles. We record the transportation-timing information when functional droplets D_1 and D_2 traverse the cross-contamination site S, i.e., D_1 traverses at clock cycle 13 and D_2 traverses at clock cycle 3. If D_1 or D_2 also traverses other cross-contamination sites in the same

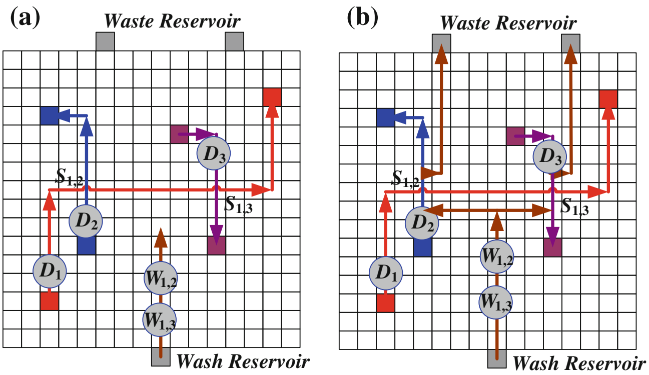

Fig. 2.2 Synchronization for two cross-contamination sites: **a** functional-droplet routes and cross-contamination sites $S_{1,2}$ and $S_{1,3}$; **b** synchronization of wash-droplet routing with functional-droplet routing

subproblem, the transportation-timing information is utilized when we adjust the arrival order of wash droplets and functional droplets at these cross-contamination sites. Next we describe the synchronization of wash droplets and functional droplets for multiple cross-contamination sites in a subproblem.

2.3.2 Multiple Cross-Contamination Sites

Within one droplet-routing subproblem, usually there are multiple cross-contamination sites where the routes for a pair of functional droplets intersect. Each cross-contamination site needs a wash droplet (or even multiple wash droplets) to clean the residue on it. Here we assume that one wash droplet is enough to clean one cross-contamination site. We propose to synchronize wash-droplet routing with functional-droplet routing for each cross-contamination site. If these cross-contamination sites are independent, i.e., a droplet route does not traverse multiple sites, we can simply utilize the synchronization method for one cross-contamination site.

Next we consider the case that a droplet route intersects with multiple routes at different cross-contamination sites. For example, as shown in Fig. 2.2a, the route of droplet D_1 intersects with the routes of droplet D_2 and D_3 at sites $S_{1,2}$ and $S_{1,3}$, respectively. Droplet D_1 is transported along its route to first traverse $S_{1,2}$ and then $S_{1,3}$. Two wash droplets, $W_{1,2}$ and $W_{1,3}$, are dispensed from the wash reservoir to clean the two cross-contamination sites $S_{1,2}$ and $S_{1,3}$, respectively.

To synchronize wash-droplet routing with functional droplet routing, first we generate the droplet routes for $W_{1,2}$ and $W_{1,3}$ to traverse $S_{1,2}$ and $S_{1,3}$, respectively, as shown in Fig. 2.2b. As mentioned in Sect. 2.3.1, each wash-droplet route consists of two sub-routes. Next we adjust the arrival order of the wash droplet and two functional droplets for each cross-contamination site. For site $S_{1,2}$, we adjust the arrival order of $W_{1,2}$, D_1, and D_2, based on their estimated arrival order acquired from the lengths of droplet routes connecting their sources to site $S_{1,2}$. Similarly, for site $S_{1,3}$, we adjust the arrival order of $W_{1,3}$, D_1 and D_3.

Assume that we first adjust the arrival order for site $S_{1,3}$ and then carry out the adjustment for site $S_{1,2}$. Therefore, we adjust for site $S_{1,2}$ after the adjustment for site $S_{1,3}$. Since D_2, D_1 and $W_{1,2}$ arrive at $S_{1,2}$ serially based on the estimate of the droplet-route lengths from their sources to $S_{1,2}$, D_1 has to be temporarily stored for several clock cycles. This has to be done to ensure that $W_{1,2}$ first passes through $S_{1,2}$; i.e., the time when D_1 passes through site $S_{1,2}$ is delayed. However, since the transportation timing of D_1 is modified at $S_{1,2}$, the arrival order of $W_{1,3}$, D_1 and D_3 at site $S_{1,3}$, which has been adjusted based on their initial estimate of the droplet-route lengths from their sources to site $S_{1,3}$, is not valid any more. Therefore, we have to adjust the arrival order for site $S_{1,3}$ again based on the updated transportation timing of D_1, which is obtained from the adjustment for $S_{1,2}$. A total of three adjustments have to be made, one for $S_{1,2}$ and two for $S_{1,3}$. If there are many cross-contamination sites within one subproblem, the adjustment for one site may have to be repeated several times if we adjust all the sites without any carefully determined order.

However, if we first adjust the arrival order for site $S_{1,2}$, we can utilize the updated transportation timing of D_1 when we adjust for site $S_{1,3}$. A total of only two adjustments, one for $S_{1,2}$ and the other for $S_{1,3}$, will now be necessary.

Calculation of Site-Adjustment Order

When we synchronize wash-droplet routing with functional-droplet routing for multiple cross-contamination sites, we have to adjust the arrival order for these sites in a predetermined order. This *site-adjustment order* is determined as follows: for a functional droplet that traverses multiple cross-contamination sites serially, we adjust the arrival order for these sites following the same order as this functional droplet traverses. Therefore, for each functional droplet in a subproblem, we first list the cross-contamination sites that it traverses along the route from the source to the sink serially. Next we combine these lists for different functional droplets together into a site-adjustment order such that the order of these cross-contamination sites maintains the same as in their original lists.

For a specific subproblem, given the order that each functional droplet traverses the corresponding multiple cross-contamination sites serially, we construct a directed graph $G = (V, E)$ as following: each vertex in the graph represents a cross-contamination site in the 2-D microfluidic array, and there is a directed edge from vertex m to n if a functional droplet traverses two cross-contamination sites m and n serially. The problem of determining the site-adjustment order can be directly mapped to the problem of the topological ordering of the graph G [17]. A topological ordering of the directed graph G is a linear ordering of its vertices such that, for every directed edge e from vertex m to vertex n, m comes before n in the ordering.

Figure 2.3 shows an example for determining the site-adjustment order. As shown in Fig. 2.3a, in a specific subproblem, the routes of four functional droplets, D_1 to

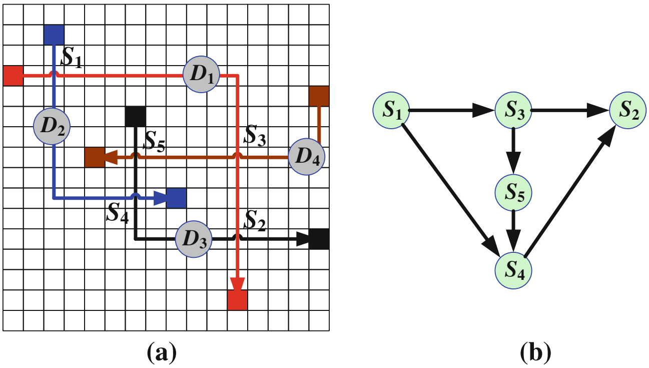

Fig. 2.3 An example for determining the site-adjustment order: **a** droplet routes and the corresponding cross-contamination sites; **b** the corresponding directed graph G for topological ordering

D_4, intersect at five cross-contamination sites from S_1 to S_5. D_1 traverses S_1, S_3 and S_2 serially; D_2 traverses S_1 and S_4 serially; D_3 traverses S_5, S_4 and S_2 serially; D_4 traverses S_3 and S_5 serially. The corresponding directed graph G for topological ordering is shown in Fig. 2.3b. For example, since D_1 traverses S_1, S_3 and S_2 serially, in the graph G, there is a directed edge from S_1 to S_3, and another directed edge from S_3 to S_2.

A topological sorting algorithm based on [17] is used to generate the ordering result based on the directed graph G, which is also the site-adjustment order. The site-adjustment order for the example in Fig. 2.3 is as following: $S_1 \rightarrow S_3 \rightarrow S_5 \rightarrow S_4 \rightarrow S_2$, where the order of these cross-contamination sites is the same as in their original lists. We adjust the arrival order for five sites following this site-adjustment order.

For the directed graph $G = (V, E)$, the worst-case computational complexity of topological sorting algorithm in [17] is $O(|V|+|E|)$. Note that a topological ordering exists if and only if the graph G is a directed acyclic graph. If there is a directed cycle in G, e.g., droplet D_1 traverses S_1 and S_2 serially, while droplet D_2 traverses S_2 and S_1 serially, the site-adjustment order does not exist. In this case, wash-droplet routing cannot be synchronized with functional-droplet routing, and wash operations have to be inserted between successive functional-droplet routing, in order to clean the residue.

Procedure for Synchronization

The procedure for synchronizing washing steps with functional droplet-routing steps in a subproblem is shown in Fig. 2.4. We adjust the arrival order for all the cross-contamination sites following the obtained site-adjustment order in Step 2. After the adjustment for one site, we record the updated transportation timing when each functional droplet traverses the site. In this manner, when adjusting the arrival order

1) **Step 1.** Generate droplet routes for the transportation of functional droplets;
2) **Step 2.** Calculate the site-adjustment order:
   i) Construct a directed graph $G = (V, E)$ for adjustment of cross-contamination sites;
   ii) Use topological sorting algorithm to generate site-adjustment order;
3) **Step 3.** For each cross-contamination site following the site-adjustment order:
   i) Generate the route for the wash droplet to traverse the cross-contamination site;
   ii) Adjust the arrival order of the wash droplet and two functional droplets at the cross-contamination site;
   ii) Update the transportation-timing information when each functional droplet traverses the cross-contamination site;
4) **Step 4.** Modify the transportation timing for functional droplets that do not traverse any cross-contamination site.

Fig. 2.4 Procedure for synchronizing washing steps with functional droplet-routing steps in a subproblem

for the next site, we can utilize the newly updated transportation timing information from the previously adjusted sites within the site-adjustment order.

For an $M \times N$ microfluidic array, the worst-case computational complexity of the routing algorithm used to generate the route for a functional or wash droplet is $O(MN)$ [3]. The worst-case time complexity of the synchronization of washing steps with functional droplet-routing steps for k cross-contamination sites in a subproblem is $O(kMN)$.

2.4 Unification of Disjoint Routing and Wash-Operation Synchronization

In this section, we present a droplet-routing algorithm for cross-contamination avoidance within a single subproblem, by unifying the disjoint droplet routes in Sect. 2.2 with wash-operation synchronization in Sect. 2.3.

In Stage 1 (disjoint routing), the disjoint droplet-routing method in Sect. 2.2 is utilized to obtain the vertex-disjoint (preferred) or edge-disjoint (as a design compromise) droplet routes. After Stage 1, if edge-disjoint routes exist, these droplet routes intersect at cross-contamination sites. In Stage 2 (wash-operation synchronization), wash operations are synchronized with functional droplet-routing using the method of Sect. 2.3 in order to avoid the cross-contamination at these sites. If all droplet routes are vertex-disjoint after Stage 1, no wash step is needed for Stage 2.

The outputs of the above procedure are the droplet routes and the transportation timing when each functional droplet traverses the cross-contamination sites. We also obtain the maximum droplet transportation time with wash steps for this subproblem. Since the number of cross-contamination sites are reduced drastically due to the disjoint routing in Stage 1, less wash droplets are needed in Stage 2, and the maximum droplet-transportation time for all the nets with wash steps may also be reduced.

For an $M \times N$ microfluidic array, for Stage 1, the worst-case computational complexity of the routing algorithm based on the Lee algorithm is $\mathrm{O}(MN)$; For Stage 2, the worst-case time complexity of the wash-operation synchronization method for k cross-contamination sites is $\mathrm{O}(kMN)$. Therefore, the worst-case time complexity of the procedure for unifying disjoint routing and wash-operation synchronization is $\mathrm{O}(kMN)$.

2.5 Cross-Contamination Avoidance Across Successive Routing Subproblems

Cross-contamination can occur within one subproblem, and also across two subproblems, e.g., a droplet route in the current subproblem may share the same cells with a droplet route in the predecessor subproblem. In order to avoid the cross-contamination across two subproblems, the droplet routes for the nets in the current subproblem should avoid sharing cells with the routes in the predecessor subproblem. Therefore, along with the active modules, the routes for all nets in the predecessor subproblem are also treated as obstacles when routes in the current subproblem are generated.

However, in order to bypass these additional obstacles, some routes will become longer and may fail “route length constraint check” (RLCC). Furthermore, a net that can be routed easily without these additional obstacles may become unroutable. Therefore, after the droplet-routing process in one subproblem, a wash operation needs to be introduced. In a wash operation, a wash droplet is routed to traverse selected cells and remove residue from them. The delay introduced by the wash operation is equivalent to the time needed to route the wash droplet. For a subproblem that includes a follow-up wash operation, its droplet-routes will not be treated as obstacles for the next subproblem.

Suppose a bioassay can be decomposed into K droplet-routing subproblems. We use Route_T_i $(1 \leq i \leq K)$ to represent droplet routes in subproblem i if the wash operation is not performed after droplet routing in subproblem $(i-1)$. The parameter T_i is defined as the maximum transportation time needed among Route_T_i. We use Route_T_i^* $(1 \leq i \leq K)$ to represent droplet routes in subproblem i if a wash operation is performed after droplet routing in subproblem $(i-1)$. The parameter T_i^* is defined as the maximum transportation time needed among Route_T_i^*. Let Tw_i, $1 \leq i \leq K$, be the time needed for the wash operation after droplet routing in subproblem $(i-1)$. Therefore, $(T_i^* + Tw_i)$ denotes the total droplet transportation time that includes the wash operation after subproblem $(i-1)$ and routing for subproblem i. If a wash

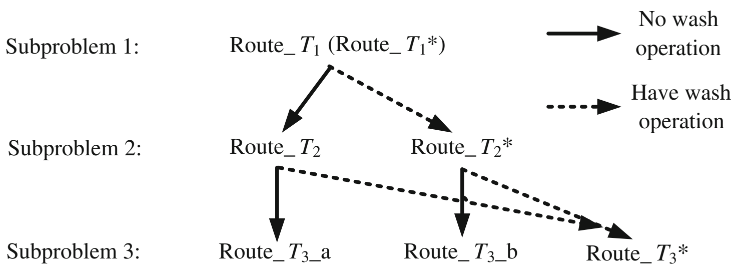

Fig. 2.5 Droplet routes for a bioassay of 3 subproblems

operation is carried out between subproblem $(i-1)$ and subproblem i, it adds to the routing time for subproblem i, but it also frees up more cells for routing and reduces the routing time in subproblem i. As a result, in most cases, $T_i^* < T_i$. The decision on whether to add a washing step between two subproblems therefore depends on the individual routing time for the subproblem and the nature of these routes.

We next describe an optimization model to determine when the wash operation should be performed between successive routing steps. Let x_i be a binary variable defined as follows:

$$x_i = \begin{cases} 1 \text{ if wash between subproblems } (i-1) \text{ and } i \\ 0 \text{ otherwise} \end{cases}$$

Our goal is to minimize the total time needed for droplet routing for the subproblems. The objective function for the optimization problem can therefore be stated as follows:

$$\text{Minimize } \mathcal{F} = T_1 + \sum_{i=2}^{K} \left(x_i \left(T_i^* + Tw_i \right) + (1 - x_i) T_i \right).$$

We also need to incorporate the constraint that the number of wash operations is less than the number of subproblems. Therefore, we now formulate the optimization problem as follows: Minimize $\mathcal{F}$, subject to: $\sum_{i=2}^{K} x_i \leq K-1$. To solve this problem, we simply note that to minimize $\mathcal{F}$, x_i needs to be set to 1 if $T_i^* + Tw_i < T_i$, otherwise $x_i = 0$. If we do not use the above formulation and simply insert wash operations between all successive routing subproblems, the total droplet-routing time will be much larger and unacceptable.

We enumerate all possibilities of droplet routes for a droplet-routing problem that includes three subproblems ($K = 3$), as shown in Fig. 2.5. For the 1st subproblem, since there is no predecessor subproblem, Route_T_1 and Route_T_1^* are the same.

For the 2nd subproblem, if no wash operation is inserted after Route_T_1 (the same as Route_T_1^*), they will be obstacles in the 2nd subproblem. The droplet routes generated in the 2nd subproblem assuming Route_T_1 (Route_T_1^*) as obstacles are denoted as Route_T_2. If there is a wash operation after the 1st subproblem, Route_T_1

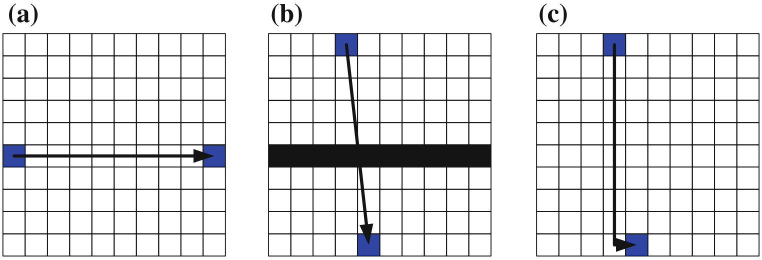

Fig. 2.6 **a** Route_T_1 (Route_T_1^*) for the 1st subproblem; **b** with Route_T_1 (Route_T_1^*) as the obstacles for the 2nd subproblem, there is no solution for Route_T_2; **c** there is always a solution for Route_T_2^*

(Route_T_1^*) will not be obstacles in the 2nd subproblem, and the corresponding droplet routes generated in this subproblem are indicated by Route_T_2^*. Note that Route_T_2 and Route_T_2^* are different.

For the 3rd subproblem, if no wash operation is inserted after Route_T_2, droplet routes generated in the 3rd subproblem assuming Route_T_2 as obstacles are denoted as Route_T_3_a; if no wash operation is inserted after Route_T_2^*, droplet routes generated in the 3rd subproblem assuming Route_T_2^* as obstacles are denoted as Route_T_3_b. If there is a wash operation after 2nd subproblem (either Route_T_2 or Route_T_2^*), Route_T_2 and Route_T_2^* will not be obstacles in the 3rd subproblem, and the corresponding droplet routes generated in this subproblem are denoted as Route_T_3^*.

However, it is not guaranteed that there is always exists a feasible Route_T_i. Figure 2.6a shows the droplet route Route_T_1 (Route_T_1^*) in the 1st subproblem. If there is no wash operation introduced after the 1st subproblem, Route_T_1 (Route_T_1^*) will be considered as obstacles in the 2nd subproblem. As shown in Fig. 2.6b, there is no solution for Route_T_2 since all possible routes from the source node to the sink node are blocked by the obstacles. Since there is no solution for Route_T_2, we cannot proceed to identify whether there is a solution for Route_T_3_a. Therefore, T_3 cannot be obtained from Route_T_3_a.

Since Route_T_2^* includes droplet routes when a wash operation is performed after droplet routing in subproblem 1, it is guaranteed that there is always a feasible solution for Route_T_2^*, as shown in Fig. 2.6c. Therefore, we can proceed to identify whether a solution exists for Route_T_3_b. Therefore, T_3 can be obtained only from Route_T_3_b, not from Route_T_3_a. Note that Route_T_2^* is only related to the sources and sinks, as well as the module placement in subproblem 2. It does not have any relationship to any droplet route in subproblem 1.

In summary, for subproblem i, T_i is directly obtained from the specific Route_T_i (e.g., Route_T_3_b in Fig. 2.5) when wash operation is not performed after the Route_T_{i-1}^*. This is because as mentioned before, it is guaranteed that there is always a feasible solution for Route_T_{i-1}^*. Note that Route_T_{i-1}^* is only related to the sources and sinks, as well as the module placement in subproblem $(i-1)$. Therefore, we can see that T_i only depends on subproblem $(i-1)$. Therefore, the problem formulation is general and we do not need to individually target all transitive (i.e., indirect) precedence relationships.

Similarly, Route_T_i^* is only related to sources and sinks, as well as module placement in subproblem i. Therefore, T_i^* is only related to subproblem i. In addition, since Tw_i is the time needed for the wash operation after droplet routing in subproblem $(i-1)$, it is only related to subproblem $(i-1)$. In summary, for subproblem i, the three constants T_i, T_i^*, and Tw_i only depend on subproblems i and $(i-1)$. The ILP model only examines the consecutive routing subproblems, hence its complexity is linear in the number of subproblems.

Note that since the duration of fluidic operations is larger than that of a routing subproblem, multiple routing subproblems may be scheduled during the implementation of a single fluidic operation. Therefore, the washing steps for fluidic operations are considered separately from the washing steps for routing subproblems. For each fluidic operation, such as mixing, splitting and dilution, one or multiple wash droplets are associated with it before and after the fluidic operation happens, the wash droplets clean the corresponding fluidic module in the array. On one hand, the cells that are used for fluidic operations will be cleaned before the fluidic operation, in order to avoid the cross-contamination with predecessor routing subproblems. On the other hand, those cells are also cleaned after the fluidic operation, and will not be contamination sites for the successor routing subproblems.

In addition, since droplet routing takes less time than a fluidic operation, the duration of wash-droplet routing can be omitted compared to the duration of fluidic operations. However, since the duration of wash-droplet routing is comparable with that functional droplet routing, in this section, we consider the insertion of washing steps for routing subproblems, in order to minimize the total routing time.

For a bioassay including K routing subproblems, in the worst case, the number of variables in the ILP model is $O(K)$, and the number of constraints is also $O(K)$.

2.6 Evaluation

In this section, we evaluate the proposed methods for cross-contamination avoidance during droplet routing. First, we present two baseline methods. Next, we utilize two bioassays, including multiplexed in vitro diagnostics on human physiological fluids and protein assay, to evaluate the proposed methods. Finally, the proposed method is compared with the contamination-aware routing method from [1]. The proposed methods were implemented on a 3.0 GHz INTEL Xeon processor, with 12 GB of memory.

2.6.1 Baseline Approaches

In order to establish the effectiveness of the proposed method, we consider two baseline methods to avoid the cross-contamination within one subproblem.

Baseline Method 1: Wash-Operation Insertion

The first baseline method inserts wash operations between successive functional-droplet routing steps within one subproblem to clean the residue at cross-contamination sites. Note that wash droplets and functional droplets are not transported concurrently in this baseline case. The maximum droplet-transportation time for all the nets is the sum of the droplet-transportation time for wash-operation steps and functional-droplet routing steps.

Baseline Method 2: Appending Wash Droplets

The second baseline method appends one wash droplet to each functional droplet. For example, in Fig. 2.2a, instead of using two wash droplets $W_{1,2}$ and $W_{1,3}$ to clean two cross-contamination sites $S_{1,2}$ and $S_{1,3}$, we append one wash droplet to each of functional droplets D_1, D_2 and D_3. During functional-droplet routing, the wash droplets follow the corresponding functional droplets and clean the residue.

For baseline method 2, the maximum droplet-transportation time for all nets can be divided into three parts. The first part is the droplet-transportation time for transporting all wash droplets from wash-droplet dispensing reservoirs to the source nodes of their corresponding functional droplets. The second part is the droplet-transportation time for moving all wash droplets from the sources nodes to the sink nodes following their corresponding functional droplets. The third part is the transportation time for moving all wash droplets to the waste reservoirs.

2.6.2 Example 1: Multiplexed In Vitro Diagnostics

Here we evaluate the proposed method for a real-life bioassay, namely multiplexed in vitro diagnostics on human physiological fluids. Three types of human physiological fluids, i.e., urine, serum and plasma, are sampled and dispensed to the digital microfluidic biochip. Glucose and lactate measurements are performed for each type of physiological fluid. The feasibility of performing the multiplexed in vitro diagnostics assay on a digital microfluidic biochip has been successfully demonstrated in experiments [18]. The sequencing graph and the schedule for assay operations of the multiplexed bioassay are presented in [3].

The routing problem is decomposed into eleven subproblems [3]. We address these subproblems serially by attempting to determine the set of vertex-disjoint or edge-disjoint droplet routes and minimize the number of cells used by these routes, subject to all the constraints. For the route length constraint, the length of each droplet route should not exceed 20 electrodes. Therefore, the route length constraint Td is equal to 20 electrodes.

Here we use subproblem 3 to illustrate the proposed disjoint-routing and wash-operation synchronization method in Sect. 2.4. As shown in Fig. 2.7a, there are three

2-pin nets and two 3-pin nets. Route 1 is defined as the path between Dt_1 and M_1 (i.e., Net 1), Route 2 is defined as the path between DI_2 and DI_3 (i.e., Net 2), Route 3 is defined as the path between DI_2 and the waste reservoir (i.e., Net 3). Route 4 is defined as the path for 3-pin net whose pins are R_1, DI_1 and M_4 (i.e., Net 4), while Route 5 is defined as the path for 3-pin net whose pins are R_2, DI_1 and M_3 (i.e., Net 5). Module M_2 that is active during this time interval is considered as an obstacle for routing. WR and WS are reservoirs for buffer droplets, wash droplets and waste, respectively.

First, as the Stage 1 of the proposed method in Sect. 2.4, we utilize the disjoint-routing method to obtain droplet routes. We decide the order for routing these nets. The priority number for Net 1 is 5, since the number of pins within its bounding box is 0 and the maximum of its $Xrange$ and $Yrange$ is 5. The priority number for Net 4 is 13, since the number of pins within its bounding box is 2 and the maximum of its $Xrange$ and $Yrange$ is 11. The priority number for Net 2, Net 3 and Net 5 are 15, 3 and 15 respectively. Therefore, the net-routing order is Net 3, Net 1, Net 4, Net 2 and Net 5.

While generating the route for Net 2, since the paths for previous routed Net 3, Net 1 and Net 4 have been marked as obstacles, the shortest paths between DI_2 and DI_3 (i.e., Route 2) violate the route length constraint, i.e., $L(\text{Route 2}) = 35$ (in cells) $> Td = 20$ (in cells), as shown in Fig. 2.7b. Thus, we put Net 2 to the conflict list, then generate the route for Net 5. After the route for Net 5 is generated, we generate the route for Net 2 without considering previously generated paths as obstacles. A route that has no common pair of adjacent cells with the previously routed paths is selected. The obtained route for Net 2 satisfies the route length constraint, i.e., $L(\text{Route 2}) = 18$ (in cells) $< Td = 20$ (in cells), as shown in Fig. 2.7c. The route for Net 2 induces two intersections with the routes for Net 4 and Net 5 respectively ($S_{2,4}$ and $S_{2,5}$). Therefore, the desirable edge-disjoint droplet-route set with total 63 cells is finally obtained. All routes satisfy both route length and fluidic constraints, as shown in Fig. 2.7c. In total, four routes (Net 1, Net 3, Net 4, and Net 5) are mutually vertex-disjoint, and one route (Net 2) is edge-disjoint with other routes. Note that for droplets D_1 and D_3 in Route 4 and Route 5 respectively, fluidic constraint might be violated if they start moving at the same time to their destinations. However, based on the modification rules, we force D_1 to stay in the current location until D_2 is transported to the mixing site and mixed with it, at the same time we continue moving D_3 to its destination, thereby overriding the constraint violation. Since the delay for Route 4 is determined by the transportation time for D_2, no extra delay is introduced during this process. The droplet-transportation time for all the routes in this subproblem is 19 clock cycles.

We compare the disjoint-routing results obtained using the proposed method for subproblem 3 with the method that utilizes the modified Lee algorithm in [3] to solve a subproblem without considering cross-contamination between different routes. This method uses four route intersections, two more than that obtained by the proposed method, as shown in Fig. 2.7d. The number of cross-contamination sites can be used to evaluate the likelihood of cross-contamination for a set of routes in a subproblem.

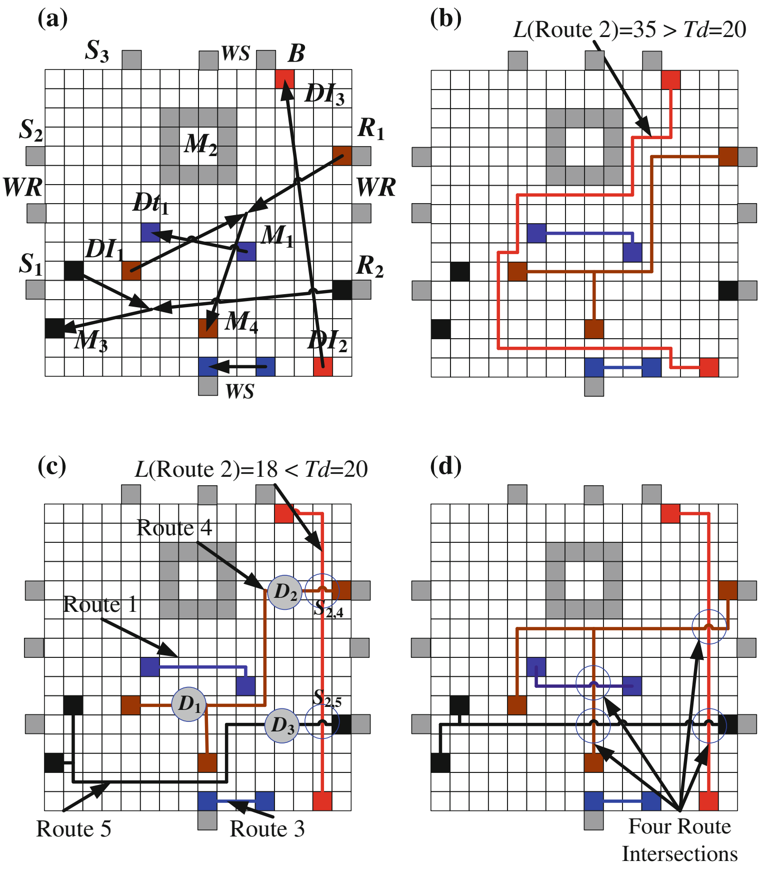

Fig. 2.7 Disjoint routing and cross-contamination-oblivious routing for subproblem 3: **a** three 2-pin nets and two 3-pin nets; **b** Route 2 violates the timing-delay constraint; **c** feasible routes for all the nets using the disjoint-routing method; **d** routing results obtained using the cross-contamination oblivious routing method

As the Stage 2 of the proposed method in Sect. 2.4, we utilize the wash-operation synchronization method to completely avoid the cross-contamination during droplet routing. For the routing results in Fig. 2.7c, there are two cross-contamination sites $S_{2,4}$ and $S_{2,5}$. Note that $S_{i,j}$ is the cross-contamination site of Route i and Route j. We next determine the site-adjustment order. Figure 2.7c shows that the droplet on Route 2 traverses $S_{2,5}$ and $S_{2,4}$ serially; therefore we obtain the site-adjustment order: $S_{2,5} \rightarrow S_{2,4}$.

We synchronize the washing steps with functional droplet transportation for the above two sites following this site-adjustment order. For the first site $S_{2,5}$ in the site-adjustment order, we first generate the route of wash droplet $W_{2,5}$ to clean the cross-contamination site $S_{2,5}$. We next adjust the arrival order of $W_{2,5}$, droplets of

Table 2.3 Comparison between methods for the multiplexed bioassay (contamination within subproblem)

Sub-prob. no.	Results for different methods (DR+WS, DR+WI, NDR+WS, NDR+WI, NDR+AW)			
	N_{cs}	N_{wash}	T_{r} (clock cycles)	T_{rw} (clock cycles)
1	(0, 0, 1, 1, 1)	(0, 0, 1, 1, 2)	(17, 17, 17, 17, 17)	(17, 17, 17, 55, 25)
2	(1, 1, 1, 1, 1)	(1, 1, 1, 1, 2)	(12, 12, 12, 12, 12)	(14, 47, 14, 47, 33)
3	(2, 2, 4, 4, 4)	(2, 2, 4, 4, 7)	(19, 19, 18, 18, 18)	(22, 59, 28, 97, 39)
4	(0, 0, 0, 0, 0)	(0, 0, 0, 0, 0)	(5, 5, 5, 5, 5)	(5, 5, 5, 5, 5)
5	(0, 0, 3, 3, 3)	(0, 0, 3, 3, 4)	(17, 17, 15, 15, 15)	(17, 17, 30, 55, 38)
6	(1, 1, 3, 3, 3)	(1, 1, 3, 3, 4)	(16, 16, 16, 16, 16)	(21, 51, 31, 52, 39)
7	(0, 0, 1, 1, 1)	(0, 0, 1, 1, 3)	(18, 18, 18, 18, 18)	(18, 18, 18, 57, 26)
8	(0, 0, 0, 0, 0)	(0, 0, 0, 0, 0)	(13, 13, 13, 13, 13)	(13, 13, 13, 13, 13)
9	(0, 0, 0, 0, 0)	(0, 0, 0, 0, 0)	(14, 14, 14, 14, 14)	(14, 14, 14, 14, 14)
10	(0, 0, 0, 0, 0)	(0, 0, 0, 0, 0)	(13, 13, 13, 13, 13)	(13, 13, 13, 13, 13)
11	(0, 0, 0, 0, 0)	(0, 0, 0, 0, 0)	(14, 14, 14, 14, 14)	(14, 14, 14, 14, 14)
Total	(4, 4, 13,13, 13)	(4, 4, 13,13, 22)	(158, 158, 155,155, 155)	(168, 268, 197,422, 259)

Route 2 and Route 5 at site $S_{2,5}$ according to Table 2.2. After the adjustment for $S_{2,5}$, we record the updated transportation timing when droplets of Route 2 and Route 5 traverses $S_{2,5}$. In this manner, while adjusting the arrival order for the next site $S_{2,4}$, we utilize the newly updated transportation timing information from $S_{2,5}$. Therefore, the maximum droplet-transportation time for all the nets with wash steps is 22 clock cycles.

For the routing results in Fig. 2.7c obtained using the proposed method, we also utilize the wash-operation insertion method (Baseline method 1) to completely avoid the cross-contamination. At first, droplets are transported along Routes 1 and 3–5 at the same time. After that, two wash droplets are dispensed from the wash reservoirs and transported via the two cross-contamination sites to the waste reservoir, respectively. After the wash operation, the droplet is transported along Route 2 via the two sites that have already been cleaned. The maximum droplet-transportation time with washing steps is 59 clock cycles. For the routing results in Fig. 2.7d obtained using the method in [3], the maximum droplet-transportation time with wash-operation insertion (97 clock cycles) is much higher than that for the edge-disjoint routing results. This is because two wash operations must be inserted sequentially during droplet routing to clean the residue on the four cross-contamination sites.

In Table 2.3, for each subproblem, we compare the number of cross-contamination sites (N_{cs}), the number of consumed wash droplets (N_{wash}), and the maximum droplet-transportation time for the nets with and without wash steps in this subproblem, e.g., T_{rw} and T_{r}. "DR" represents the disjoint droplet routing method proposed in Sect. 2.2, and "NDR" represents the modified Lee algorithm in [3] that solves a subproblem without considering cross-contamination between different routes. "WS" represents the wash-operation synchronization method proposed in Sect. 2.3, "WI" represents the wash-operation insertion method (Baseline method 1), and "AW"

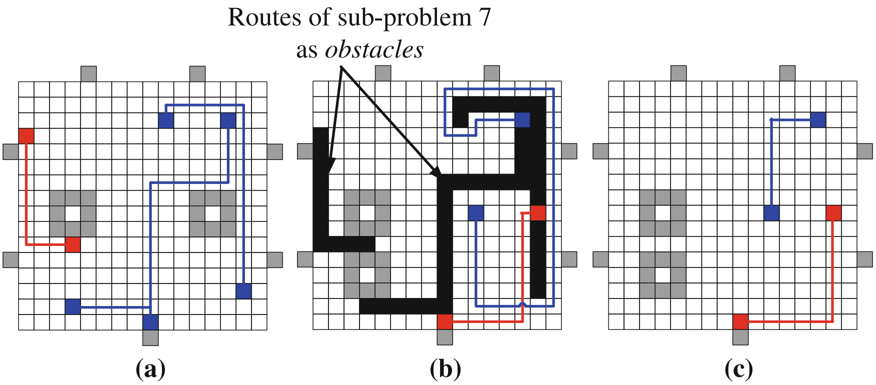

Fig. 2.8 Routes for subproblem 7 with or without wash operation: **a** disjoint-routing solution for subproblem 7; **b** disjoint routes for subproblem 8 when no wash operation is performed before it; **c** disjoint routes for subproblem 8 when a wash operation performed

represents the method that appends wash droplets after functional droplets (Baseline method 2). Therefore, "DR+WS" represents the proposed method in Sect. 2.4 that unifies disjoint routing method and wash-operation synchronization.

$N_{cs} = 0$ denotes that the obtained routes are vertex-disjoint. In subproblems 4 and 8–11, all the methods obtain the same values for N_{cs}, N_{wash}, T_r and T_{rw}, since "DR" and "NDR" generate the same routes without any cross-contamination site. Therefore, no wash operation is needed.

Here we analyze the results for subproblem 3 in Table 2.3. For the "DR+WS" method, since "DR" reduces N_{cs} from 4 to 2 compared with "NDR", and wash-droplet routing is synchronized with functional-droplet routing, the maximum droplet-transportation time with wash steps (T_{rw}) is slightly higher than that without wash steps (T_r). For the "NDR+WS" method, although there are four cross-contamination sites, the value of T_{rw} is only a little higher than that of 'DR+WS", since the wash-operation is synchronized with functional droplet routing steps. For "DR+WI", since the wash operations are inserted between successive functional droplet routing steps, the value of T_{rw} is much higher than that of "DR+WS" and "NDR+WS" methods. The value of T_{rw} for "NDR+WI" is much higher than that of "DR+WI", since there are more cross-contamination sites, and more wash operations are needed to be inserted. The "NDR+AW" method consumes more wash droplets that other methods, since one wash droplet has to be appended to each functional droplet. Although "NDR+AW" achieves lower value of T_{rw} than that of "NDR+WI", it is still higher than that of "NDR+WS". The results shows that the proposed "DR+WS" method significantly reduces the overall routing time by generating disjoint routes and synchronizing wash operations.

In Fig. 2.8, we illustrate the influence of cross-contamination across two subproblems on the droplet routing. Figure 2.8a shows the disjoint routing solution for subproblem 7. If no wash operation is performed between subproblems 7 and 8,

Table 2.4 Parameters for optimization model (in clock cycles)

Subproblem no.	DR+WI			NDR+WI		
	T	T^*	Tw	T	T^*	Tw
1	17	17	0	55	55	0
2	10,000	47	17	10,000	47	17
3	10,000	59	12	10,000	97	12
4	5	5	19	10,000	5	18
5	17	17	5	55	55	5
6	10,000	51	17	10,000	52	15
7	39	18	16	66	57	16
8	76	13	18	76	13	18
9	14	14	13	14	14	13
10	13	13	14	13	13	14
11	14	14	13	14	14	13

the routes for all nets in subproblem 7 are treated as obstacles for subproblem 8, as shown in Fig. 2.8b. The disjoint routes for subproblem 8 are also shown in Fig. 2.8b. The maximum transportation time T_8 is 76 clock cycles. If a wash operation is performed, the routes for subproblem 8 do not consider the routes for subproblem 7 to be obstacles, as shown in Fig. 2.8c. The maximum transportation time T_8^* is now reduced to 13 clock cycles. Moreover, since the routes for subproblem 2 require a large number of cells, we cannot obtain feasible routes for subproblem 3 if a wash operation is not performed between subproblems 2 and 3. For such cases, we set T_i to be a large value, i.e., 10,000, for this work.

Table 2.4 lists the values of T_i, T_i^* and Tw_i, $1 \leq i \leq 11$, for each subproblem using "DR+WI" and "NDR+WI" methods. For the "DR+WI" method, using the optimization model presented in Sect. 2.5, we obtain the minimum routing time for all the subproblems to be 348 clock cycles. A wash operation must be performed before subproblems 2, 3 and 6–8 for this optimal routing plan. For the "NDR+WI" method, the minimum routing time for all the subproblems is much higher, 511 clock cycles, and we have to perform a wash operation before subproblems 2, 3, 4, 6 and 8. The CPU time for solving the ILP model for the multiplexed in vitro diagnostics is 0.5 s.

2.6.3 Example 2: Protein Assay

We next evaluate the proposed method for a real-life protein assay that has been carried out on a digital microfluidic lab-on-chip [19]. The sequencing graph model for the protein-assay protocol of $DF = 128$ is presented in [20]. Based on the synthesis results using the method in [20], the routing problem is decomposed into 127 subproblems.

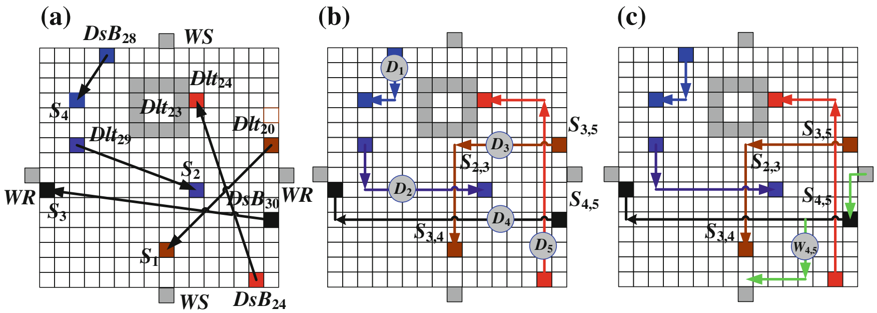

Fig. 2.9 Synchronization of washing steps with functional droplet-routing steps in Subproblem 82: **a** five 2-pin nets; **b** functional droplet routes; **c** the wash-droplet route for cross-contamination site $S_{4,5}$

Table 2.5 Comparison between methods for the protein assay (contamination within subproblem)

Subprob. no.	Results for different methods (DR+WS, DR+WI, NDR+WS, NDR+WI, NDR+AW)			
	N_{cs}	N_{wash}	T_r (clock cycles)	T_{rw} (clock cycles)
...	...	...	...	...
78	(0, 0, 1, 1, 1)	(0, 0, 1, 1, 2)	(8, 8, 8, 8, 8)	(8, 8, 16, 18, 16)
79	(0, 0, 0, 0, 0)	(0, 0, 0, 0, 0)	(4, 4, 4, 4, 4)	(4, 4, 4, 4, 4)
80	(1, 1, 3, 3, 3)	(1, 1, 3, 3, 4)	(17, 17, 16, 16, 16)	(17, 29, 18, 38, 20)
81	(1, 1, 5, 5, 5)	(1, 1, 5, 5, 7)	(20, 20, 19, 19, 19)	(28, 44, 33, 55, 42)
82	(3, 3, 4, 4, 4)	(3, 3, 4, 4, 5)	(18, 18, 18, 18, 18)	(24, 58, 26, 71, 34)
83	(1, 1, 3, 3, 3)	(1, 1, 3, 3, 5)	(14, 14, 13, 13, 13)	(14, 22, 17, 51, 23)
84	(0, 0, 1, 1, 1)	(0, 0, 1, 1, 2)	(10, 10, 8, 8, 8)	(10, 10, 11, 19, 15)
85	(1, 1, 4, 4, 4)	(1, 1, 4, 4, 6)	(20, 20, 18, 18, 18)	(21, 35, 24, 58, 27)
86	(0, 0, 0, 0, 0)	(0, 0, 0, 0, 0)	(11, 11, 11, 11, 11)	(11, 11, 11, 11, 11)
87	(1, 1, 3, 3, 3)	(1, 1, 3, 3, 5)	(19, 19, 17, 17, 17)	(19, 31, 19, 46, 33)
...	...	...	...	...
Total	(28, 28, 89, 89, 89)	(28, 28, 89, 89, 198)	(901, 901, 837, 837, 837)	(983, 1478, 1187, 1702, 1405)

In Subproblem 82, there are five 2-pin nets for the transportation of functional droplets, and Dlt_{23} is concurrently operating, as shown in Fig. 2.9a. As shown in Fig. 2.9b, Droplet D_1 transports along the route from the buffer dispenser for DsB_{28} to an on-chip storage unit S_4; Droplet D_2 transports from the diluter for Dlt_{29} to an on-chip storage unit S_2; Droplet D_3 transports from the diluter for Dlt_{20} to an on-chip storage unit S_1; Droplet D_4 transports from the buffer dispenser for DsB_{30} to an on-chip storage unit S_3; Droplet D_5 transports from the buffer dispenser for DsB_{24} to the diluter for Dlt_{24}. The routes of four functional droplets, D_2 to D_5, intersect at four cross-contamination sites. $S_{i,j}$ is the cross-contamination site of routes for D_i and D_j.

As Stage 2 of the proposed method in Sect. 2.4, we synchronize the wash operations with functional droplet routing steps. We determine the site-adjustment order. Figure 2.9b shows that D_3 traverses $S_{3,5}$, $S_{2,3}$ and $S_{3,4}$ serially; D_4 traverses $S_{4,5}$ and $S_{3,4}$ serially; D_5 traverses $S_{4,5}$ and $S_{3,5}$ serially. We combine these lists together into a site-adjustment order using the topological sorting method in Sect. 2.3: $S_{4,5} \rightarrow S_{3,5} \rightarrow S_{2,3} \rightarrow S_{3,4}$, where the order of these cross-contamination sites maintains the same as in their original lists.

We synchronize the washing steps with functional droplet transportation for the above four sites following this site-adjustment order. For the first site $S_{4,5}$ in the site-adjustment order, we first generate the route of wash droplet $W_{4,5}$ to clean the cross-contamination site $S_{4,5}$, as shown in Fig. 2.9c. The route of $W_{4,5}$ consists of two sub-routes. The first sub-route connects the wash reservoir (WR) to $S_{4,5}$, and the second sub-route connects $S_{4,5}$ to the waste reservoir (WS). We next adjust the arrival order of $W_{4,5}$, D_4 and D_5 at site $S_{4,5}$. According to Table 2.2, D_5 is made to arrive after $W_{4,5}$ traverses $S_{4,5}$. Therefore, D_4, $W_{4,5}$ and D_5 traverse $S_{4,5}$ serially at clock cycle 1, 5 and 9, respectively, in order that $W_{4,5}$ cleans the residue left by D_4 at $S_{4,5}$ before D_5 traverses. After the adjustment for $S_{4,5}$, we record the updated transportation timing when D_4 and D_5 traverses $S_{4,5}$. In this manner, while adjusting the arrival order for the next site $S_{3,5}$, we utilize the newly updated transportation timing information from $S_{4,5}$. Other sites in the site-adjustment order, i.e., $S_{3,5}$, $S_{2,3}$, and $S_{3,4}$, are adjusted by repeating the above process.

Since it is not feasible to show the results for all the 127 routing subproblems, Table 2.5 shows a fragment of subproblems in the protein assay. For each subproblem, we compare N_{cs}, N_{wash}, T_{r} and T_{rw} for different methods. The “DR+WS” method obtains lowest value of T_{rw}, since it targets disjoint routes to reduce the number of cross-contamination sites, and the wash operations are synchronized with functional droplet routing steps to minimize the overall routing time. The “DR+WI” method also targets disjoint routes, thereby it obtains the same values of N_{cs} and N_{wash} as that of “DR+WS”. However, since it inserts wash operations between successive functional droplet routing within one subproblem, the value of T_{rw} is much higher than that of “DR+WS”. Methods “NDR+WS”, “NDR+WI” and “NDR+AW” do not consider the avoidance of cross-contamination when generating droplet routes, therefore the values of N_{cs} and N_{wash} are much higher than that of “DR+WS” and “DR+WI”. Especially, since wash steps are inserted between successive functional-droplet routing steps, the value of T_{rw} of the “NDR+WI” method is much higher than other methods. The “NDR+AW” method consumes more wash droplets, since one wash droplet has to be appended to each functional droplet. Although this method achieves lower maximum droplet-transportation time than “NDR+WI”, it is still higher than “NDR+WS”.

Using the optimization model for inter-subproblem washing presented in Sect. 2.5, for the “DR+WI” method, we obtain the minimum routing time for all the subproblems to be 2078 clock cycles. A wash operation must be performed before a total of 36 subproblems for this optimal routing plan.

Table 2.6 Comparison of the proposed method ("DR+WS") with the droplet-routing method in [1]

	Simulation results (DR+WS, [1])		
Bioassay	N_{cs} (N_{wash})	T_{rw}	N_{cell}
Multiplexed bioassay	(4, 21)	(168, 193)	(376, 351)
Protein assay	(28, 61)	(983, 1108)	(1471, 1362)

2.6.4 Comparison with [1]

Finally, we compare the proposed method ("DR+WS") in Sect. 2.4 with the routing method in [1]. The CPU times of the "DR+WS" method for the multiplexed bioassay and the protein bioassay are 0.49 and 2.10 s, respectively. As shown in Table 2.6, for both the multiplexed bioassay and the protein assay, droplet routes obtained using the proposed "DR+WS" method have less cross-contamination sites (N_{cs}) and hence utilize less wash droplets (N_{wash}) than [1]. This is because for a specific routing subproblem, "DR+WS" generates disjoint droplet routes that drastically reduce the number of cross-contamination sites, while [1] has to route on the preferred routing tracks. The number of used cells for routing (N_{cell}) of "DR+WS" is a little higher than that of [1]. This is because "DR+WS" generates disjoint routes to avoid the cross-contamination, therefore utilizes more cells for routing.

The maximum droplet-transportation time with wash steps (T_{rw}) of "DR+WS" is lower than that of [1]. This is because wash steps are only needed for subproblems that have cross-contamination sites. By using the "DR+WS" method, cross-contamination sites within a subproblem can be avoided by disjoint routes, hence the number of routing subproblems without any cross-contamination site is more than that of the method in [1]. For a specific subproblem, if there is no cross-contamination site using "DR+WS", no wash step is needed; while there are cross-contamination sites using the method in [1], wash steps must be used to clean these sites, which require extra time and lead to higher T_{rw} than that of "DR+WS". In addition, for routing subproblems that have cross-contamination sites using both "DR+WS" and [1], "DR+WS" synchronizes wash steps with functional droplet-routing steps, therefore it reduces the total droplet-transportation time.

2.7 Chapter Summary and Conclusions

In this chapter, we have presented a droplet-routing and optimization method for digital microfluidics that avoids cross-contamination. This method targets disjoint droplet routes and integrates washing steps into functional droplet-routing steps. By carefully adjusting the arrival orders of wash droplets and functional droplets at cross-contamination sites, we synchronize the routing of wash droplets with functional droplet-routing steps, thereby reducing droplet-transportation time. An optimization method has also been presented to minimize the number of wash operations that must be used between successive routing steps. Two real-life bioassay applications, namely multiplexed in vitro diagnostics on human physiological fluids and protein assay, have been used to evaluate this method.

References

1. T.-W Huang, C.-H. Lin, T.-Y. Ho, A contamination aware droplet routing algorithm for digital microfluidic biochips. in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, (2009), pp. 151–156
2. M. Cho, D. Z. Pan, A high-performance droplet router for digital microfluidic biochips. in *Proceedings of the ACM International Symposium on Physical Design*, (2008), pp. 200–206
3. F. Su, W. Hwang, K. Chakrabarty, Droplet routing in the synthesis of digital microfluidic biochips. in *Proceedings of the IEEE/ACM Design, Automation and Test in Europe Conference*, (2006), pp. 323–328
4. P.-H. Yuh, C.-L. Yang, Y.-W. Chang, BioRoute: A network flow based routing algorithm for digital microfluidic biochips. in *Proceedings of the IEEE/ACM International Conference on, Computer-Aided Design*, (2007), pp. 752–757
5. P.-H. Yuh, S. Sapatnekar, C.-L. Yang, Y.-W. Chang, A progressive-ILP based routing algorithm for cross-referencing biochips. in *Proceedings of the IEEE/ACM Design Automation Conference*, (2008), pp. 284–289
6. T. Xu, K. Chakrabarty, Integrated droplet routing in the synthesis of microfluidic biochips. in *Proceedings of the IEEE/ACM Design Automation Conference*, (2007), pp. 948–953
7. C.-Y. Lin, Y.-W. Chang, Cross contamination aware design methodology for pin-constrained digital microfludic biochips. in *Proceedings of the IEEE/ACM Design Automation Conference*, (2010), pp. 641–646
8. R. B. Fair, A. Khlystov, T. D. Tailor, V. Ivanov, R. D. Evans, P. B. Griffin, V. Srinivasan, V.K. Pamula, M.G. Pollack, J. Zhou, Chemical and biological applications of digital-microfluidic device. IEEE Des. Test Comput. **24**, 10–24, 2007
9. V. Hlady, R. A. Wagenen, J. D. Andrade, in *Surface and Interfacial Aspects of Biomedical Polymers: Protein Adsorption*, ed. by J.D. Andrade, vol. 2 (Plenum Press, New York, 1985), pp. 81
10. H. Moon, A. R. Wheeler, R. L. Garrell, J. A. Loo, C.-J. Kim, An integrated digital microfluidic chip for multiplexed proteomic sample preparation and analysis by MALDI-MS. Lab Chip **6**, 1213–1219 (2006)
11. M. Campas, I. Katakis, DNA biochip arraying, detection and amplification strategies. Trends Anal. Chem. **23**, 49–62 (2003)
12. K. Sermon, V. Goossens, S. Seneca, W. Lissens, A. Vos, M. Vandervorst, A. Steirteghem, I. Liebaers, Preimplantation diagnosis for Huntington's disease (HD): clinical application and analysis of the HD expansion in affected embryos. Prenat. Diagn. **18**, 1427–1436 (1999)
13. F. Su, K. Chakrabarty, Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips. in *Proceedings of the IEEE/ACM Design Automation Conference*, (2005), pp. 825–830
14. J.F. Lynch, The equivalence of theorem proving and the interconnection problem. ACM SIGDA Newslett. **5**, 31–36 (1975)
15. M.R. Kramer, J. Leeuwen, The complexity of wire routing and finding the minimum area layouts for arbitrary VLSI circuits. Adv. Comput. Res. **2**, 129–146 (1984)
16. S. Sait, H. Youssef, *VLSI Physical Design Automation: Theory and Practice* (IEEE Press, NewYork, 1995)
17. A.B. Kahn, "Topological sorting of large networks". Commun. ACM **5**, 558–562 (1962)
18. V. Srinivasan, V. K. Pamula, M. G. Pollack, and R. B. Fair, Clinical diagnositics on human whole blood, plasma, serum, urine, saliva, sweat, and tears on a digital microfluidic platform. in *Proceedings of the MicroTAS*, (2003), pp. 1287–1290

19. V. Srinivasan, V.K. Pamula, P. Paik, R.B. Fair, Protein stamping for MALDI mass spectrometry using an electrowetting-based microfluidic platform. Proc. Soc. Photogr. Instrum. Eng. **5591**, 26–32 (2004)
20. K. Chakrabarty, F. Su, *Digital Microfluidic Biochips: Synthesis, Testing, and Reconfiguration Techniques* (CRC Press, Boca Raton, FL, 2006)