High-Level Synthesis of Digital Microfluidic Biochips

FEI SU
Intel Corporation
and
KRISHNENDU CHAKRABARTY
Duke University

Microfluidic biochips offer a promising platform for massively parallel DNA analysis, automated drug discovery, and real-time biomolecular recognition. Current techniques for full-custom design of droplet-based "digital" biochips do not scale well for concurrent assays and for next-generation system-on-chip (SOC) designs that are expected to include microfluidic components. We propose a system design methodology that attempts to apply classical high-level synthesis techniques to the design of digital microfluidic biochips. We focus here on the problem of scheduling bioassay functions under resource constraints. We first develop an optimal scheduling strategy based on integer linear programming. However, because the scheduling problem is NP-complete, we also develop two heuristic techniques that scale well for large problem instances. A clinical diagnostic procedure, namely multiplexed in-vitro diagnostics on human physiological fluids, is first used to illustrate and evaluate the proposed method. Next, the synthesis approach is applied to a protein assay, which serves as a more complex bioassay application. The proposed synthesis approach is expected to reduce human effort and design cycle time, and it will facilitate the integration of microfluidic components with microelectronic components in next-generation SOCs.

Categories and Subject Descriptors: B.7.2 [**Integrated Circuits**]: Design Aids; B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids; J.3 [**Life and Medical Sciences**]: Biology and genetics, Health

General Terms: Algorithms, Performance, Design, Reliability

Additional Key Words and Phrases: High-level synthesis, scheduling, microfluidics, biochips, system-on-chip

This research was supported in part by the National Science Foundation under grants IIS-0312352 and CCF-0541055.

A preliminary and abridged version of this article was presented in *Proceedings of the IEEE International Conference on Computer-Aided Design*. 2004, 223–228.

Authors' addresses: F. Su, Intel Corporation FM7-287, 1900 Prairie City Road, Folsom, CA 95630; email: fei.su@intel.com; K. Chakrabarty, Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708; email: krish@ee.duke.edu.

ACM Reference Format:
Su, F. and Chakrabarty, K. 2008. High-level synthesis of digital microfluidic biochips. ACM J. Emerg. Technol. Comput. Syst. 3, 4, Article 16 (January 2008), 32 pages. DOI = 10.1145/1324177.1324178 http://doi.acm.org/10.1145/1324177.1324178

1. INTRODUCTION

Microfluidic biochips for biochemical analysis are receiving much attention nowadays [Burns et al. 1998; Zhang et al. 2002; Thorsen et al. 2002; Verpoorte and Rooij 2002; Su et al. 2006]. These composite microsystems, also known as lab-on-a-chip or bio-MEMS, offer a number of advantages over conventional laboratory procedures. They automate highly repetitive laboratory tasks by replacing cumbersome equipment with miniaturized and integrated systems, and they enable the handling of small amounts, for example, nanoliters, of fluids. Thus they are able to provide ultrasensitive detection at significantly lower costs per bioassay than traditional methods, and in a significantly smaller amount of laboratory space.

Advances in microfluidics technology offer exciting possibilities in the realm of enzymatic analysis (e.g., glucose and lactate assays), DNA analysis (e.g., PCR and nucleic acid sequence analysis), proteomic analysis involving proteins and peptides, immuno-assays, and toxicity monitoring. An emerging application area for microfluidic biochips is clinical diagnostics, especially immediate point-of-care diagnosis of diseases [Schulte et al. 2002; Srinivasan et al. 2004]. Microfluidics can also be used for countering bio-terrorism threats [Hull et al. 2003; Venkatesh and Memish 2003]. Microfluidics-based devices, capable of continuous sampling and real-time testing of air/water samples for biochemical toxins and other dangerous pathogens, can serve as an always-on "bio-smoke alarm" for early warning.

Most current microfluidic biochips contain permanently etched micropumps, microvalves, and microchannels, and their operation is based on the principle of continuous fluid flow [Thorsen et al. 2002; Verpoorte and Rooij 2002]. A promising alternative is to manipulate liquids as discrete droplets [Pollack et al. 2000; Cho et al. 2002]. Following the analogy of microelectronics, this approach is referred to as "digital microfluidics." In contrast to continuous-flow biochips, digital microfluidic biochips offer a scalable system architecture based on a two-dimensional microfluidic array of identical basic cells. Moreover, because each droplet can be controlled independently, these "digital" systems also have dynamic reconfigurability, whereby groups of cells in a microfluidic array can be reconfigured to change their functionality during the concurrent execution of a set of bioassays. The advantages of scalability and reconfigurability make digital microfluidic biochips a promising platform for massively parallel DNA analysis, automated drug discovery, and real-time biomolecular detection.

As the use of digital microfluidic biochips increases, their complexity is expected to become significant due to the need for multiple and concurrent assays on the chip, as well as more sophisticated control for resource management. Time-to-market and fault tolerance are also expected to emerge as design

considerations. However, current design methodologies for microfluidic biochips are typically full-custom and bottom up in nature. Devices are first optimized using detailed physical simulation, and they can then be used to assemble a complete microfluidic biochip system. These full-custom methodologies will not scale well for larger designs because of higher complexity and the need for quantifiable performance metrics that can be optimized. There is a pressing need to deliver the same level of computer-aided design (CAD) support to the biochip designer that the semiconductor industry now takes for granted [Su et al. 2006]. Moreover, it is expected that these microfluidic components will be integrated with microelectronic components in next-generation system-on-chip (SOC) designs. The 2003 International Technology Roadmap for Semiconductors (ITRS) clearly identified the integration of electrochemical and electro-biological techniques as one of the system-level design challenges that will be faced beyond 2009, when feature sizes shrink below 50 nm [ITRS 2003].

Much of recent research on CAD for microfluidic biochips has been focused on device-level physical modeling of single components [Shapiro et al. 2003; Zeng and Korsmeyer 2004]. Although these modeling and simulation tools facilitate the physical design of microfluidic devices, they are not adequate for system-level design. While top-down system-level design tools are now commonplace in IC design, few such efforts have been reported for digital microfluidics. Microfluidics-specific synthesis tools are needed to relieve biochip users from the burden of manually optimizing a set of assays for increased throughout. These tools will allow users to describe bioassays at a high level of abstraction; they will then map the behavioral description to the microfluidic array and generate an optimized schedule of bioassay operations, the binding of assay operations to resources, and the layout of the microfluidic biochip. Thus, the biochip user can concentrate on the development of the nano- and microscale bioassays, leaving implementation details to the synthesis tools. Motivated by the analogy between digital microfluidic biochips and digital integrated circuits, we propose a system design methodology that attempts to apply classical high-level synthesis techniques to the biochip design, and thus speed up the design cycle and reduce human effort. Figure 1 illustrates the proposed approach. The synthesis of a digital microfluidic biochip can be divided into two major phases, broadly referred to as high-level synthesis (i.e., architectural-level synthesis) and geometry-level synthesis (i.e., physical design). A behavioral model for a biochemical assay is first obtained from the protocol for that assay. Next, high-level synthesis is used to generate a macroscopic structure of the biochip; this structure is analogous to a structural RTL model in electronic CAD. This macroscopic model provides an assignment of assay functions to biochip resources, as well as a mapping of assay functions to time-steps, based in part on the dependencies between them. Finally, geometry-level synthesis creates a physical representation at the geometrical level, that is, the final layout of the biochip consisting of the configuration of the microfluidic array, locations of reservoirs and dispensing ports, and other geometric details.

In this article, we focus on the high-level synthesis problem, that is, the scheduling of assay functions under resource constraints. The first contribution

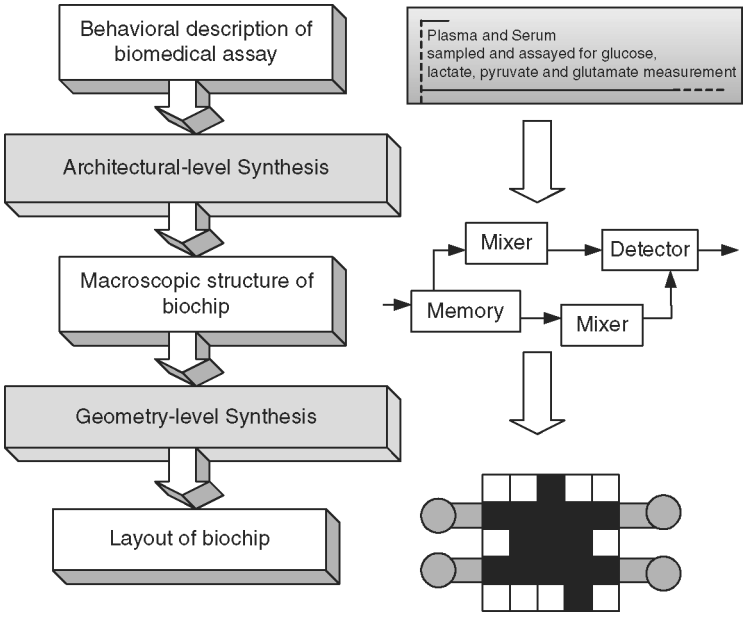

Fig. 1. Synthesis methodology for digital microfluidic biochips.

of this paper lies in the application of conventional high-level synthesis methods to an emerging technology area, that is, biochips based on microfluidics arrays. The second contribution of this work lies in the formulation of biochip-specific CAD problems that incorporate the properties and constraints unique to droplet-based microfluidics. While a number of classical CAD optimization techniques can be used for the system-level design of biochips, these techniques must be tailored to handle the dynamic reconfigurability of digital microfluidic biochips.

The organization of the remainder of the article is as follows. We first present an overview of digital microfluidic biochips in Section 2. Related prior work is next discussed in Section 3. Section 4 presents the proposed high-level synthesis methodology for digital microfluidics. After the general problem formulated in Section 4.1, a real-life clinical diagnosis procedure, namely multiplexed in-vitro diagnostics on human physiological fluids, is used as a case study and described in Section 4.2. Section 4.3 presents a sequencing graph model for this multiplexed bioassay. Based on this graph formulation, an integer linear programming (ILP) model is developed and evaluated in Section 4.4. In Section 4.5, we first show that the scheduling problem is NP-complete, and then we present two heuristic algorithms to solve the problem in a computationally efficient manner. Simulation results are presented to evaluate the proposed methods in Section 5. The problem of resource selection, which is inherent in the design of digital microfluidic biochips, is also investigated. We next evaluate the heuristic approach for a larger problem instance, that is, dilution-based protein assay. Finally, conclusions are drawn in Section 6.

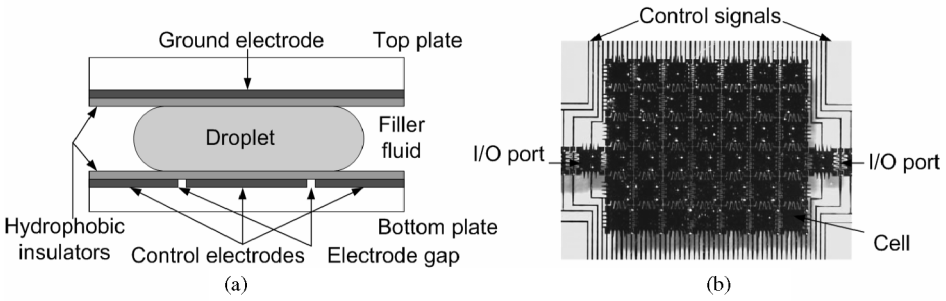

Fig. 2. (a) Basic cell used in an EWOD-based digital microfluidic biochip; (b) a two-dimensional array for digital microfluidics [Pollack 2001].

2. DIGITAL MICROFLUIDIC BIOCHIPS

The basic idea of microfluidic biochips is to integrate all necessary functions for biochemical analysis onto one chip using microfluidics technology. Integrated functions include microfluidic assay operations and detection, as well as sample pre-treatment and preparation. There are two different classes of microfluidic biochips, namely continuous-flow biochips and droplet-based microfluidic biochips.

Alternatives to closed-channel continuous-flow systems include novel open structures, where the liquid is divided into discrete, independently controllable droplets, and these droplets can be manipulated to move on a substrate. A number of methods for manipulating microfluidic droplets have been proposed in the literature [Gallardo et al. 1999; Ichimura et al. 2000; Sammarco and Burns 1999; Washizu 1998; Jones et al. 2001]. Among the proposed techniques, electrowetting-on-dielectric (EWOD) has received considerable attention in recent years. We focus here on the EWOD-based biochips, also referred to as digital microfluidic biochips.

The basic cell of an EWOD-based biochip consists of two parallel glass plates, as shown in Figure 2(a). The bottom plate contains a patterned array of individually controllable electrodes, and the top plate is coated with a continuous ground electrode. All electrodes are formed by optically transparent indium tin oxide (ITO). A dielectric insulator, that is, parylene C, coated with a hydrophobic film of Teflon AF, is added to the top and bottom plates to decrease the wettability of the surface and to add capacitance between the droplet and the control electrode. The detailed fabrication process is described in Pollack [2001]. The droplet containing biochemical samples and the filler medium, such as the silicone oil, are sandwiched between the plates; the droplets travel inside the filler medium. The use of silicone oil as the filler medium has been shown to greatly reduce cross-sample contamination, which facilitates microfluidic biochip reuse [Fair et al. 2004]. In order to move a droplet, a control voltage is applied to an electrode adjacent to the droplet and at the same time the electrode just under the droplet is deactivated. The EWOD effect causes an accumulation of charge in the droplet/insulator interface, resulting in a surface tension gradient across the gap between the adjacent electrodes, which consequently causes the

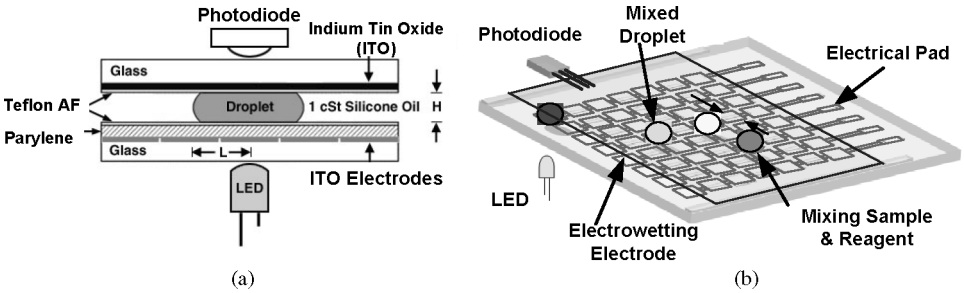

Fig. 3. Schematic of a digital microfluidic biochip used for colorimetric assays: (a) basic cell; (b) top view of microfluidic array.

transportation of the droplet. By varying the electrical potential along a linear array of electrodes, electrowetting can be used to move nanoliter volume liquid droplets along this line of electrodes [Pollack 2001]. The velocity of the droplet can be controlled by adjusting the control voltage ($0 \sim 90$ V), and droplets can be moved at speeds of up to 20 cm/s [Pollack et al. 2002]. Droplets can also be transported, in user-defined patterns and under clocked-voltage control, over a two-dimensional array of electrodes shown in Figure 2(b) without the need for micropumps and microvalves. This regular scalable structure facilitates the use of a hierarchical and cell-based approach for microfluidic biochip design. In this scenario, we envisage that a large-scale integrated digital microfluidic biochip can be constructed out of repeated instances of well-characterized cells in the same way that complex VLSI circuits can be built using well-characterized transistors.

Using a two-dimensional array, many common operations for different biomedical assays can be performed, such as sample introduction (*dispense*), sample movement (*transport*), temporarily sample preservation (*store*), and the mixing of different samples (*mix).* For instance, the *store* operation is performed by applying an insulating voltage around the droplet. The *mix* operation is used to route two droplets to the same location and then turn them around some pivot points. The configurations of the microfluidic array, that is, the routes that droplets travel and the rendezvous points of droplets, are programmed into a microcontroller that controls the voltages of electrodes in the array. In this sense, these mixers and storage units in the array can be viewed as reconfigurable virtual devices. This property of digital microfluidic biochips, referred to as dynamic reconfigurability, leads to a high-level synthesis scenario that is different from high-level synthesis for electronic circuits. The dynamic reconfigurability property of digital microfluidics is key for system integration, because it allows us to handle all the steps of biochemical analysis on-chip, from sampling, sample-processing, mixing, and detection to waste handling.

In addition to a basic microfluidic array platform, a generic digital microfluidic biochip also consists several reservoirs that store and generate the droplets of samples and reagents, and an integrated optical detection system consisting of LEDs and photodiodes; see Figure 3. Both the dynamic reconfigurability of biochips and the regularity of assay protocols facilitate the simultaneous

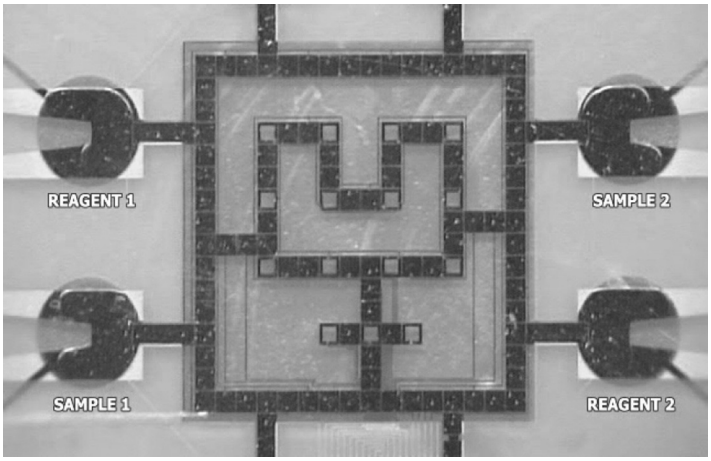

Fig. 4. Fabricated microfluidic array used in multiplexed biomedical assays [Srinivasan et al. 2004].

operation of multiple bioassays on one single platform. Figure 4 shows an image of such a fabricated microfluidic chip used for multiplexed biomedical assays; this chip has been fabricated in Duke University's Digital Microfluidics laboratory [Srinivasan et al. 2004]. Other demonstrated applications of digital microfluidics include the on-chip detection of explosives such as commercial-grade 2,4,6-trinitrotoluene (TNT) and pure 2,4-dinitrotoluene [Pamula et al. 2005], automated on-chip measurement of airborne particulate matter [Fair et al. 2004; Zhao and Cho 2006]. Digital microfluidic biochips are being designed for on-chip gene sequencing through synthesis [Fair et al. 2006], protein crystallization, and clinical diagnostics for high throughput with low sample volumes, and integrated hematology, pathology, molecular diagnostics, cytology, microbiology, and serology onto the same platform [Advanced Liquid Logic]. Recently, Silicon Biosystems has implemented a droplet-based biochip that embeds more than 600,000 20 μm by 20 μm electrodes and uses dielectrophoresis for droplet manipulation and control [Silicon Biosystems].

In a recent review paper on the use of microfluidics for protein crystallization [Woerd et al. 2003], the following question was posed: can we purchase identical crystallization devices, produced under adequate quality control? The authors go on to say, "Drawing upon integrated circuits as an analogy, microfluidics devices may be reducible to a standard set of discrete operations which can then be custom assembled to form more complex operations as needed. With this approach, the success of manufacturing investment does not have to rest upon a single application." The discrete droplet-based biochip being considered in this paper is perfectly suited as a platform technology, since it avoids the common pitfall of custom devices offered by other continuous-flow microfluidic technologies.

3. RELATED PRIOR WORK

Design automation research for digital microfluidic biochips can benefit from classical CAD techniques. Synthesis of integrated circuits is a well-studied

problem [De Micheli 1994] and advances in high-level and logic synthesis techniques continue even today [Camposano 1996]. Walker and Camposano [1991] provides an excellent survey of high-level synthesis work. Driven by the need to integrate digital and analog functions in a mixed-signal circuit, analog circuit synthesis has also gained momentum in recent years [Antao and Brodersen 1992].

MEMS design is a relatively young field compared to IC design. Since the concept of special CAD systems for MEMS was first proposed at Transducer '87 [Senturia 1987], several research groups have reported significant progress in this area, and a number of commercial MEMS CAD tools are now available [Fedder and Jing 1999; De and Aluru 2003]. Many of these tools are focused solely on the modeling of thermal and electro/mechanical properties. Recently, synthesis tools for MEMS have also been developed [Mukherjee and Fedder 1998]. However, because of the differences in actuation methods between MEMS and digital microfluidics, they cannot be directly used for the design of microfluidic biochips.

While MEMS design tools have reached a certain level of maturity, CAD tools for biochips are still in their infancy. Some design automation techniques have been proposed for DNA probe arrays [Kahng et al. 2003]; however, the digital microfluidic biochips described in this article are more versatile and complex than DNA arrays. For microfluidic biochips, some commercial computational fluid dynamics (CFD) tools, such as CFD-ACE+ from CFD Research Corporation and FlumeCAD from Coventor, Inc. support 3D simulation of microfluidic transport. A recent release of CoventorWare from Coventor, Inc. includes microfluidic behavioral models to allow top-down system-level design [CoventorWare]. Pfeiffer et al. [2004] also presented a synthesis approach for multiplexed capillary electrophoresis (CE) separation microchips. Unfortunately, these CAD tools are only able to deal with continuous flow systems, and they are therefore inadequate for the design of digital microfluidic biochips. Recently behavioral modeling for droplet-based microfluidic systems has been investigated [Böhringer 2004; Griffith and Akella 2004; Ding et al. 2001]. In early work in this area, Ding et al. [2001] presented an architectural design and optimization methodology for 2-D arrays based on electrowetting. Integer linear programming was used to minimize the bioassay processing time in Ding et al. [2001]. Unfortunately, this model, while useful for real-time polymerase chain reaction (PCR), does not scale well for larger problems. Moreover, this early work is more for post-layout optimization, where droplet pathways and locations for storage, mixing, and splitting operations need to be predefined by the user. Priority scheduling in digital microfluidic biochips was proposed in Ricketts et al. [2006]. Reconfiguration and defect tolerance techniques for biochips were described in Su and Chakrabarty [2006]. A comprehensive cost-effective test methodology for digital microfluidic systems was proposed in Su et al. [2004]. Based on the detection mechanism, an efficient concurrent testing scheme that interleaves test application with a set of bioassays was proposed in Su et al. [2004]. These testing techniques can be further integrated with system-level synthesis tools to facilitate design-for-test (DFT) for digital microfluidic biochips.

4. HIGH-LEVEL SYNTHESIS METHODOLOGY

4.1 Problem Formulation

The goal of a synthesis procedure is to select a design that minimizes a certain cost function under resource constraints. High-level synthesis for microfluidic biochips can be viewed as the problem of scheduling assay functions and binding them to a given number of resources so as to maximize parallelism, thereby decreasing response time. Conventional high-level synthesis methods can be leveraged for the emerging biochip domain. First, a well-defined droplet-based assay protocol can be modeled by a sequencing graph as in the case of high-level synthesis for integrated circuits. By using discrete unit-volume droplets, a microfluidic function can be reduced to a set of repeated basic operations, that is, moving one unit of fluid (droplet) over one unit of instance (one electrode length). This "digitization" method facilitates the implementation of many well-defined protocols for nano- and microscale bioassays on a microchip. A generic class of microdroplet-based bioassay protocols that can be applied to digital microfluidic biochips usually consists of the following steps: 1) dispensing sample/reagent droplets into the microfluidic array; 2) transporting the droplets to some locations on the array for assays operations (e.g., mixing, dilution or optical detection); 3) finally moving the droplets of assay products or wastes out of the array. We denote such a generic bioassay protocol as P_{assay} and its corresponding sequencing graph as G_p.

We next formulate a scheduling problem, whose goal is to determine the start times and stop times of all assay operations, subject to the precedence constraints imposed by the sequencing graph. Resource constraints also need to be satisfied for bioassay scheduling. Here we use the term resource binding to refer to the mapping of bioassay operations to available functional resources. In a valid schedule, assay operations that share a resource cannot execute concurrently. Due to resource constraints, a resource binding may associate one functional resource with several assay operations of the same type; this necessitates resource sharing. Moreover, due to dynamic reconfigurability, we can allow operations of different types, for example, mixing and storing, to share the same microfluidic cells on the array during different time spans. We also note that there may be several types of resources for any given bioassay operation. For example, a 2×2-array mixer, a 2×3-array mixer and a 2×4-array mixer can be used for a droplet mixing operation [Paik et al. 2003]. These mixers differ in their areas as well as mixing times. In such cases, a resource selection procedure must be used.

Based on the above definitions, this leads us to the following optimization problem for high-level synthesis of digital microfluidic biochips:

HLS_Bio. Given P_{assay} (and its corresponding sequencing graph G_p) and the available resources, determine a schedule of assays as well as an assignment of assay operations to functional resources, such that the assay completion time is minimized without any resource conflicts.

The minimization of the assay completion time is essential for environmental monitoring applications where sensors can provide early warning. Real-time response is also necessary for surgery and neonatal clinical diagnostics.

Finally, biological samples are sensitive to the environment and to temperature variations, and it is difficult to maintain an optimal clinical or laboratory environment on chip. For example, protein crystallization and high-throughput DNA sequencing (HTS) are two emerging applications of digital microfluidics, especially for concurrent fluid handling on a chip. In protein crystallization, a large number of conditions, defined by different combinations of precipitants, diluents, and other reagents in various concentrations, must be evaluated for the likelihood of the formation of protein crystals. Concurrent processing will allow the evaluation of these conditions rapidly and in parallel. HTS is a procedure that is very useful for sequencing many different templates of DNA with any number of primers. These operations can also be run in parallel on the same chip, reducing the time and cost associated with HTS. To ensure the integrity of assay results, it is therefore desirable to minimize the time that samples spend on-chip before assay results are obtained. Increased throughput also improves operational reliability. Long assay durations imply that high actuation voltages need to be maintained on some electrodes, which accelerate insulator degradation and dielectric breakdown, reducing the number of assays that can be performed on a chip during its lifetime. Therefore, assay processing time must be minimized for both disposable and reusable biochips.

Note that there are some key difference between the new problem *HLS_Bio* and the high-level synthesis problem for digital circuits. The first results from the dynamic reconfigurability of digital microfluidic biochips. Unlike electronic components in circuits, many microfluidic components (e.g., mixers and storage units) can be dynamically formed anywhere on a 2-D array. The dynamic reconfigurability provided by digital microfluidic biochips is in many ways similar to the partial reconfiguration offered by Dynamically Reconfigurable FPGAs (DRFPGAs). However, the programmability of DRFPGAs is limited by the well-defined roles of interconnect and logic blocks. Interconnect cannot be used for storing information, and logic blocks cannot be used for routing. In contrast, the digital microfluidic biochips offer significantly more programmability. The cells in the microfluidic array can be used for storage, functional operations, as well as for transporting fluid droplets.

Another major difference is that while conventional high-level synthesis targets pre-layout design, synthesis for biochip encompasses both pre- and post-manufacture. This feature meets the increasing need for field programmability to support multiple concurrent assays on the same chip. Even for disposable biochip applications, reconfigurability is attractive since it facilitates the microfluidic array structure to be reconfigured for different assay applications, thereby reducing product cost due to the possibility of high-volume manufacturing.

In the following sections, we first use a real-life biochemical assay, namely multiplexed in-vitro diagnostics, to illustrate the *HLS_Bio* problem formulation. We then present algorithms for the high-level synthesis of digital microfluidic biochips. Finally, we apply the proposed heuristic approach to a larger biochemical application, namely, protein assay involving a series of on-chip dilution steps.

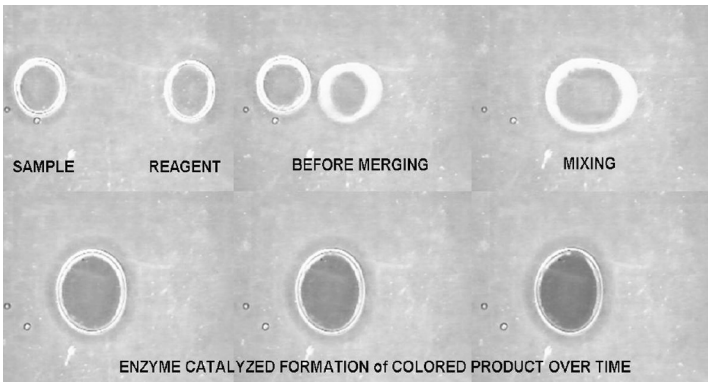

Fig. 5. Photos of different steps of a glucose assay carried out on a digital microfluidic biochip [Srinivasan et al. 2004].

4.2 Illustrative Example: Multiplexed In-Vitro Diagnostics on Human Physiological Fluids

The in-vitro measurement of glucose and other metabolites, such as lactate, glutamate, and pyruvate, in human physiological fluids is of great importance in clinical diagnosis of metabolic disorders. For instance, the change of regular metabolic parameters in the patient's blood can signal organ damage or dysfunction prior to observable microscopic cellular damages or other symptoms. Protocols for enzyme-kinetic measurements of metabolites are suitable for droplet-based microfluidics implementation. The feasibility of performing a colorimetric glucose assay on a digital microfluidic biochip has been successfully demonstrated in experiments [Srinivasan et al. 2003, 2004].

The glucose assay performed on the biochip is based on Trinder's reaction, a colorimetric enzyme-based method. The enzymatic reactions involved in the assay:

$$\text{Glucose} + H_2O + O_2 \xrightarrow{\text{Glucose Oxidase}} \text{Gluconic Acid} + H_2O_2$$

$$2H_2O_2 + \text{4-AAP} + \text{TOPS} \xrightarrow{\text{Peroxidase}} \text{Quinoneimine} + 4H_2O$$

In the presence of glucose oxidase, glucose can be enzymatically oxidized to gluconic acid and hydrogen peroxide. Then, in the presence of peroxidase, the hydrogen peroxide reacts with 4-amino antipyrine (4-AAP) and N-ethyl-N-sulfopropyl-m-toluidine (TOPS) to form violet-colored quinoneimine, which has an absorbance peak at 545 nm. Based on this colorimetric reaction, a complete glucose assay can be performed following three steps, namely, transportation, mixing, and optical detection, as shown in Figure 5. Sample droplets containing glucose and reagent droplets containing glucose oxidase, peroxidase, 4-AAP, and TOPS, are dispensed into the microfluidic array from droplet reservoirs. They are then transported towards a mixer where droplets of the sample and the reagent are mixed together and the enzymatic reaction happens during the mixing. A droplet of the product is moved to the location of the optical detector. The optical detection is performed using a green LED and a photodiode. The

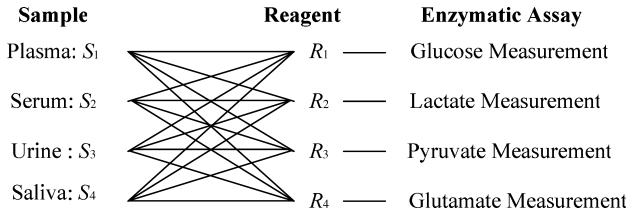

Fig. 6. One example of multiplexed in-vitro diagnostics.

glucose concentration can be detected from the absorbance, which is related to the concentration of colored quinoneimine. Experiments have shown that the results from the digital microfluidic biochip match well with the reference values obtained from conventional measurements [Srinivasan et al. 2004].

In addition to glucose assays, the detections of other metabolites such as lactate, glutamate and pyruvate in a digital microfluidic biochip have also been demonstrated recently [Srinivasan et al. 2004]. Furthermore these assays can be integrated together to form a multiplexed in-vitro diagnostics on different human physiological fluids, which can be performed concurrently on a microfluidic biochip.

4.3 Sequencing Graph Model

The behavioral description of an example of a multiplexed in-vitro diagnostics is shown in Figure 6. Four types of human physiological fluids—plasma, serum, urine, and saliva—are sampled and dispensed into the microfluidic biochip. Next each type of physiological fluid is assayed for glucose, lactate, pyruvate or glutamate measurement. For each enzymatic assay, the droplets containing the suitably modified reagents (e.g., Glucose oxidase, Peroxidase, 4-AAP and TOPS for glucose measurement) are dispensed into the microfluidic array from the relevant reservoirs. The result of each type of bioassays can be detected using a dedicated optical absorbance measurement device.

An abstract model of a bioassay behavior at the architectural level can be developed in terms of operations and the dependencies between them. We use the sequencing graph model from high-level synthesis terminology [Micheli 1994]. We assume that there are a total of n_{ops} operations. The sequencing graph is acyclic and polar. There are two vertices, called source v_0 and sink v_k, that present the first and last no-operation task, where $k = n_{ops} + 1$. Hence the sequencing graph $G(V, E)$ has vertex set $V = \{v_i : i = 0, 1, \ldots, k\}$ in one-to-one correspondence with the set of assay operations, and edge set $E = \{(v_i, v_j) : i, j = 0, 1, \ldots, k\}$ representing dependencies. With each node v_i, we associate a weight $d(v_i)$, which denotes the time taken for operation v_i. The details of these operations and the resources that these operations use are as follows. (We assume that m types of physiological fluids are assayed for n types of enzymatic measurements.)

Input Operations. These operations consist of the generation of the droplets of samples (S_i, $i = 1, \ldots, m$) or reagents ($R_i, i = 1, \ldots, n$) from the on-chip reservoir, which are then dispensed into the microfluidic array. These operations are represented using the nodes shown in Figure 7. There are $m + n$ types

Fig. 7. Nodes representing the input operations.

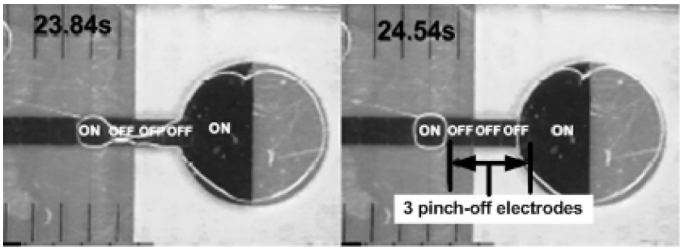

Fig. 8. On-chip reservoirs to store and dispense droplets [Ren and Fair 2002].

of input operations (denoted by $I_i, i = 1, \ldots, m + n$), where $I_j, j = 1, \ldots, m$ represents the generation and dispensing of droplets of sample S_j. Similarly, $I_{j+m}, j = 1, \ldots, n$, denotes the operation for reagent R_j.

Assumption 1. We assume that the time required to generate and dispense droplets from the reservoir is determined mainly by the system parameters, such as the aspect ratio of the channel gap to electrode gap [Ren and Fair 2002]. The properties of the fluid have little impact on the input operation time. This assumption has been verified by experimental data [Ren and Fair 2002].

Assumption 1 implies that the weights of the input operation nodes are equal. That is, there is no difference between the operation times required for generating and dispensing different samples and reagents. Experiments indicate that droplet generation and dispensing takes 2 seconds [Ren and Fair 2002]. Therefore, we set one unit of time to 2 seconds, and let $d(I_i) = 1$ unit of time, where $i = 1, \ldots, m + n$.

In order to avoid unexpected contamination between different samples and reagents, at least one reservoir is needed for each type of fluid. We assume that there are Nr reservoirs for each type of fluid ($Nr \geq 1$). Moreover, these reservoirs belong to the category of non-reconfigurable resources, that is, they are fixed after design and fabrication, as shown in Figure 8 [Ren and Fair 2002].

Mixing Operation. In order to perform the required enzymatic assay, droplets of samples need to be mixed with droplets of reagents on the microfluidic array.

Assumption 2. The viscosities of the different reagents are almost the same because they are highly diluted by the same fluid, such as H_2O, before dispensing [Srinivasan et al. 2003, 2004].Thus the time required for complete mixing mainly depends on the viscosity of the sample. For the same sample, the mixing time can be considered to be the same for different reagents.

Based on this assumption, which is supported by available experimental data, we define m types of mixing operations $M_1, M_2, \ldots, M_m$, represented by the nodes shown in Figure 9.

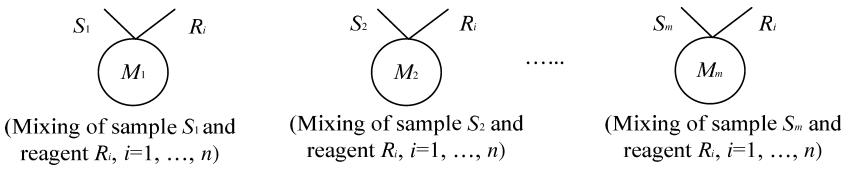

Fig. 9. Nodes representing the mixing operations.

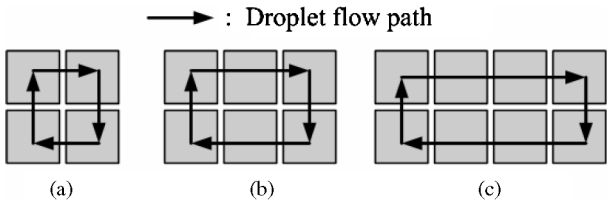

Fig. 10. (a) 2×2-array mixer; (b) 2×3-array mixer; (c) 2×4-array mixers.

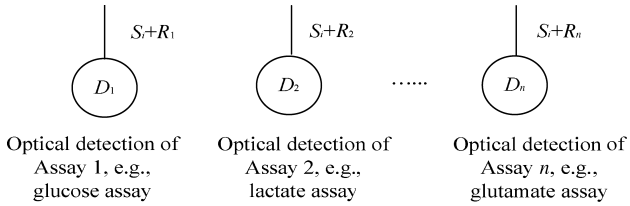

Fig. 11. Nodes representing the detection operations.

The weights of the nodes representing the different type of mixing operations are different; for example, $d(M_1) = 5$ for plasma, $d(M_2) = 3$ for serum, $d(M_3) = 4$ for urine, and $d(M_4) = 6$ for saliva. The resources corresponding to the mixing operations are reconfigurable mixers. Here we use 2×2-array mixers for these operations. Note that, in addition to 2×2-array mixers, there exist other types for resources for mixing operations, for example, 2×3-array mixers or 2×4-array mixers, as shown in Figure 10 [Srinivasan et al. 2003]. These mixers differ in their areas as well as mixing times. In such cases, a resource selection procedure must be used; it will be analyzed later.

Detection Operation. After mixing, the results of enzymatic assays are detected using an integrated LED-photodiode setup.

Assumption 3. The type of enzymatic assay determines the optical detection time. Experiments showed that the types of samples have little impact on optical detection time [Srinivasan et al. 2003, 2004].

Based on this assumption, n types of detection operations $D_1, D_2, \ldots, D_n$ are shown in Figure 11. The weights of the nodes representing different type of detection are different. For instance, $d(D_1) = 5$ for glucose, $d(D_2) = 4$ for lactate, $d(D_3) = 6$ for pyruvate, and $d(D_4) = 5$ for glutamate measurements. The resources corresponding to these operations are integrated optical detectors consisting of a LED-photodiode setup. At least one detector is needed for each type of enzymatic assay. We assume that there are Nd detectors for each

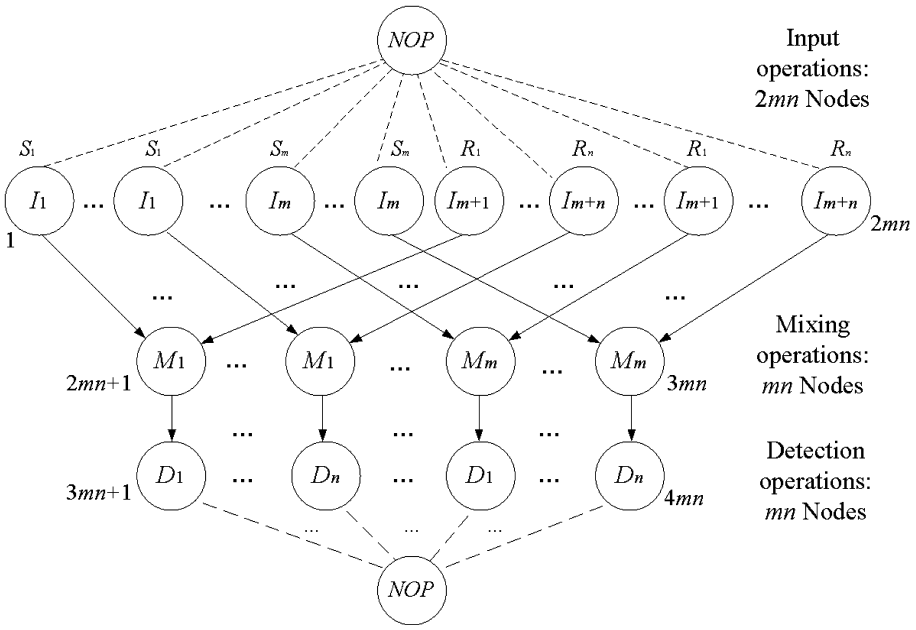

Fig. 12. Sequencing graph model for a multiplexed in-vitro diagnostics.

type of assay ($Nd \geq 1$). These resources also belong to the category of non-reconfigurable resources.

Assumption 4. In contrast to the previous operations, droplet movement on a digital microfluidic array is very fast [Srinivasan et al. 2003, 2004]. Therefore, we can ignore the droplet movement time for scheduling assay operations. This implies that the weight of an edge in the sequencing graph is zero, that is, there is no communication cost between operations. We expect that advances in microfluidics technology will continue to shorten assay operation times (e.g., mixing), and the droplet transportation time will then become comparable to the assay operation times. For such scenarios, the proposed method must be augmented to include nonzero edge weights. Efforts are also underway to model the droplet routing problem and routability-driven high-level synthesis. These problems are however beyond the scope of this paper.

After defining all the basic operations, the multiplexed in-vitro diagnostics on human physiological fluids can now be modeled by the sequencing graph shown in Figure 12.

It is important to note that for optimal scheduling of assay operations under resource constraints, the memory resource also needs to be considered. If two sequential operations are not scheduled in consecutive time-steps, a storage unit, that is, a memory resource, is required to store the droplet temporarily; see Figure 13. These memory resources also belong to the category of reconfigurable virtual devices, which can be dynamically formed by changing the controls voltages of the corresponding cells. Moreover, they can be reused as mixers upon completion of the storage operation. Since the total number of

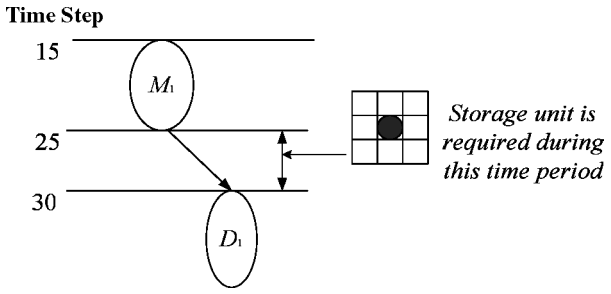

Fig. 13. Illustration of the need for storage units.

cells in the array is fixed, for any given time-slot, the number of available storage units decreases with an increase to the number of designated mixers. This constraint adds to the complexity and specificity of the scheduling problem for digital microfluidic biochips.

4.4 Integer Linear Programming Model

Based on the sequencing graph developed in Section 4.3, we now address the resource-constrained optimization problem for multiplexed in-vitro diagnostics on human physiological fluids. As in Ding et al. [2001], we first develop an integer linear programming (ILP) model for this problem. The model is developed for a general sequencing graph, and without loss of generality, the example of Section 4.3 is used for explanation.

First we define a binary variable as follows:

$$X_{ij} = \begin{cases} 1 & \text{if operation } v_i \text{ start at time slot } j. \\ 0 & \text{otherwise,} \end{cases}$$

where $1 \leq i \leq n_{ops}$ and $1 \leq j \leq T$. As defined in Section 4.3, the total number of operations, excluding non-operation source and sink nodes, in the sequencing graph is n_{ops}. For the sequencing graph shown in Figure 12 for multiplexed in vitro diagnostics, $n_{ops} = 4mn$. Without loss of generality, we use the quantity $4mn$ instead of the variable n_{ops} in the following discussion. The parameter T is the maximum possible index for a time slot and its value can be set to an easily determined loose upper bound.

Since each operation is scheduled exactly once, $\sum_{j=1}^{T} X_{ij} = 1$ for $1 \leq i \leq 4mn$. The starting time of operation v_i can be expressed as $St_i = \sum_{j=1}^{T} j \times X_{ij}$ for $1 \leq i \leq 4mn$. The goal of optimal scheduling is to minimize the assay completion time under resource constraints. The completion time of the last operation is $C = \max\{St_i + d(v_i) : v_i \in D_1, D_2, \ldots, D_n\}$. Therefore, the objective function for the ILP model is to minimize C subject to $C \geq St_i + d(v_i)$, $i = 3mn+1, \ldots, 4mn$.

Next, the following constraint inequalities need to be incorporated into this model.

Dependency Constraints:

If there is a dependency between operation v_i and operation v_j, the constraint $St_j \geq St_i + d(v_i)$ should be satisfied.

Resource Constraints:

Reservoirs/Dispensing Ports. We assume that Nr reservoirs/dispensing ports are assigned to each type of fluid. Without loss of generality, we set $Nr = 1$, that is, there are $m+n$ reservoirs/dispensing ports attached to the microfluidic array, where m reservoirs store and generate the sample droplets, and the others are for n types of assay reagents. This constraint is expressed as:

$$\sum_{i:v_i \in I_1} X_{ij} \leq 1, \quad \sum_{i:v_i \in I_2} X_{ij} \leq 1, \ldots \sum_{i:v_i \in I_{m+n}} X_{ij} \leq 1 : 1 \leq j \leq T.$$

Reconfigurable Mixers and Reconfigurable Storage Units. The *mix* operation is performed on a reconfigurable mixer, which serves a virtual device. In addition, there might exist other virtual devices, for example, reconfigurable storage units, which are formed to temporarily store the droplet between two sequential operations. The total number of the available virtual devices is constrained by the size of a microfluidic array. We model this resource relationship as follows:

$$Nmixer(j) + \beta \cdot Nmemory(j) \leq Na \quad \text{for } 1 \leq j \leq T,$$

where $Nmixer(j)$ is the number of mixers needed at time slot j, $Nmemory(j)$ is the number of storage units needed at time slot j, and Na is a value determined by the array size. In fact, N_a is determined from the size of the microfluidic array, described in terms of the number of cells. It is normalized to the number of mixers of a given size that can be accommodated on the array. We also assume that the size of a storage unit is a fraction β of the size of a mixer. If a 2×2-array mixer is the only type of resource used for mixing, $\beta = 0.25$. Note that the overheads due to isolation cells and cells used for droplet transportation are not included here. These overheads are modeled more appropriately during module placement [Su and Chakrabarty 2006] and droplet routing [Su et al. 2006], which are beyond the scope of this paper. An appropriate value of β can be easily obtained for other types of mixers or if taking isolation cells into account. Finally, the ILP model does not handle mixers of unequal sizes, thus the problem of resource module selection is not addressed here. Instead, it is separately addressed in Section 5.2.

An operation v_i is executed at time slot j when $\sum_{l=j-d(v_i)+1}^{j} X_{il} = 1$. Therefore, the number of mixers used in time slot j is as follows:

$$Nmixer(j) = \sum_{\{i:v_i \in M_1, M_2, \ldots M_m\}} \quad \sum_{l=j-d(v_i)+1}^{j} X_{il} \ .$$

In order to find the number of storage units (memory) needed at time slot j, we define a binary variable as follows:

$$M_{ij} = \begin{cases} 1 & \text{if storage unit } i \text{ is needed at time slot } j. \\ 0 & \text{otherwise} \end{cases}$$

where memory (storage unit) i is assigned to the edge between vertex v_i and its directed successor vertex v_k, $1 \leq i \leq 3mn$ and $1 \leq j \leq T$. Then $Nmemory(j) = \sum_{i=1}^{3mn} M_{ij}$. To determine M_{ij}, we set $A_{ij} = 1 - \sum_{l=1}^{j} X_{il}$, that is, A_{ij} is 1 only

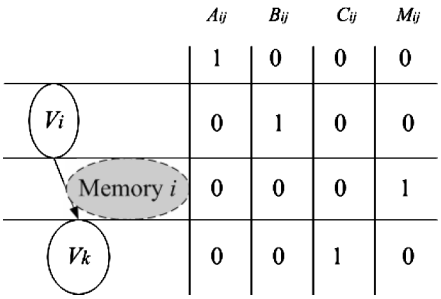

Fig. 14. Variables in different time steps.

before operation v_i starts; $B_{ij} = \sum_{l=j-d(v_i)+1}^{j} X_{il}$, that is, B_{ij} is 1 only when v_i is active; $C_{ij} = \sum_{l=1}^{j} X_{kl}$, that is, C_{ij} is 1 only when v_k starts. Note that the above definition of C_{ij} only considers cases where each vertex has only one successor. This is indeed the case for many multiplexed bioassay protocols that we have encountered. The definition of C_{ij} will need to be modified to handle the cases of multiple successors. For the protein assay considered later in this article, which includes dilution and splitting step leading to multiple successors, we do not use the ILP method. It can be easily seen that $M_{ij} + A_{ij} + B_{ij} + C_{ij} = 1$; see Figure 14. Therefore, $M_{ij} = 1 - A_{ij} - B_{ij} - C_{ij} = \sum_{l=1}^{j-d(v_i)} X_{il} - \sum_{l=1}^{j} X_{kl}$.

Optical Detectors. We assign Nd detectors to each enzymatic assay; for example, $Nd = 1$. Similar to the constraint for reservoir/dispensing ports, this functional resource constraint can be modeled as follows:

$$\sum_{i:v_i \in D_1} \sum_{l=j-d(v_i)}^{j} X_{il} \leq 1, \ldots, \sum_{i:v_i \in D_{n1}} \sum_{l=j-d(v_i)}^{j} X_{il} \leq 1 : \quad 1 \leq j \leq T.$$

We have now developed the ILP model for P_{assay} using multiplexed bioassay as an illustrative example. The general ILP model is shown in Figure 15. The complexity of this formulated ILP model for the scheduling problem is O(mnT) in the number of variables and O($mn + Tm + Tn$) in the number of constraints.

This ILP model is evaluated for a problem of modest size. For instance, plasma and serum are sampled and assayed for glucose, lactate and pyruvate measurements; that is, $m = 2$, $n = 3$, as shown in Figure 16. We assume $Nr = Nd = 1$, and $Na = 4$. We use a popular public domain ILP solver called *lpsolve* [Berkelaar]. It took over 3 hours of CPU time on a 1.0 GHz Pentium-III PC with 256 MB of RAM. The optimal schedule for this multiplexed biomedical assay is shown in Figure 17. The completion time for the whole assay is 17 time-slots; that is, 34 seconds.

4.5 Heuristics for the Scheduling Problem

The scheduling problem being studied here is equivalent to the resource-constrained scheduling problem with non-uniform weights of operation nodes, which has been proven to be $\mathcal{NP}$-complete [Garey and Johnson 1979; Kwok

Minimize: $C = \max\{St_i + d(v_i) : v_i \in D_1, D_2, \ldots, D_n\}$

Subjective to

1) $C \geq St_i + d(v_i)$, $i = 3mn+1, \ldots, 4mn$.

2) $St_j \geq St_i + d(v_i)$

3) $\sum_{i:v_i \in I_1} X_{ij} \leq 1, \sum_{i:v_i \in I_2} X_{ij} \leq 1, \ldots \sum_{i:v_i \in I_{m+n}} X_{ij} \leq 1 : \ 1 \leq j \leq T.$

4) $Nmixer(j) + 0.25\, Nmemory(j) \leq Na \quad$ for $1 \leq j \leq T$.

5) $Nmixer(j) = \sum_{\{i:v_i \in M_1, M_2, \ldots M_m\}} \sum_{l=j-d(v_i)+1}^{j} X_{il}$.

6) $Nmemory(j) = \sum_{i=1}^{3mn} M_{ij}$.

7) $M_{ij} = 1 - A_{ij} - B_{ij} - C_{ij} = \sum_{l=1}^{j-d(v_i)} X_{il} - \sum_{l=1}^{j} X_{sl}.$

8) $\sum_{i:v_i \in D_1} \sum_{l=j-d(v_i)}^{j} X_{ij} \leq 1, \ldots, \sum_{i:v_i \in D_{n1}} \sum_{l=j-d(v_i)}^{j} X_{ij} \leq 1 : \ 1 \leq j \leq T.$

Fig. 15. Integer linear programming (ILP) model for the scheduling problem.

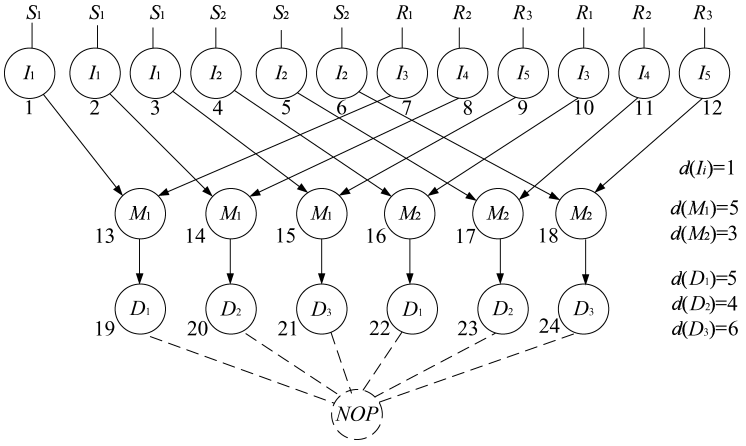

Fig. 16. Sequencing graph corresponding to one instance of multiplexed diagnostics.

and Ahmad 1999]. In order to solve this problem in a computationally efficient manner for large instances, we develop heuristics in this section.

Modified List Scheduling Algorithm. This heuristic extends the well-known *List Scheduling* algorithm [De Micheli 1994]. The pseudocode in Figure 18 illustrates how the *list scheduling* algorithm from the literature is modified to handle reconfigurable resources such as mixers and storage units. Compared

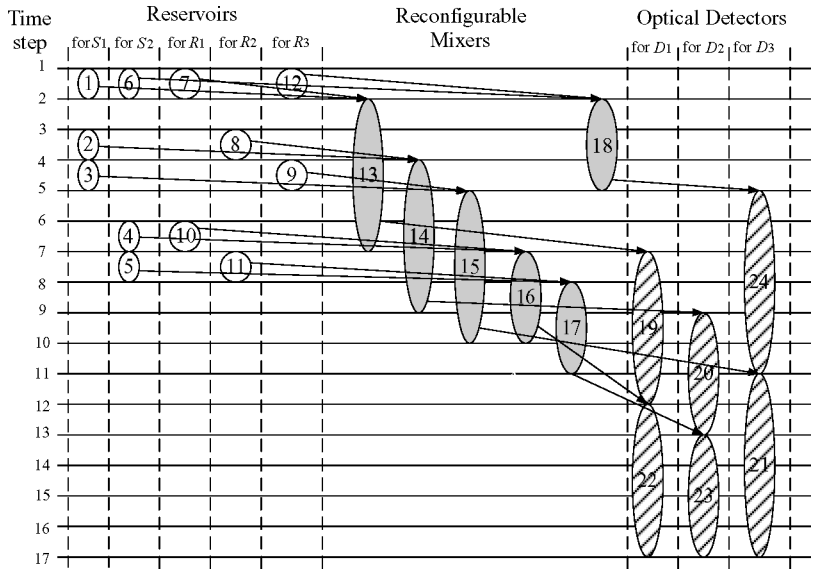

Fig. 17. Optimal schedule obtained using the ILP model.

Procedure Modified *List Scheduling* Algorithm

Given the sequencing graph $G = (V, E)$
Given the resource constraints Nr, Na and Nd
Time step $t = 1$;
Repeat { /* *Phase I (for non-reconfigurable resources)* */
  for *(each non-reconfigurable resource type* k*)*
    Determine candidate operations $U(t, k) = \{v_i \in V$: Type$(v_i) = k$ and $t_j + d(v_j) \leq t \ \forall j$: $(v_j, v_i) \in E\}$;
    Determine unfinished operations $T(t, k) = \{ v_i \in V$: Type$(v_i) = k$ and $t_i + d(v_i) > t, t_i \leq t \}$;
  Select $S(t, k) \subseteq U(t, k)$ based on the urgency priority list, such that $|S(t, k)| + |T(t, k)| \leq Nr$ (or Nd);
    Set $t_i = t \ \forall i$: $v_i \in S(t, k)$
  end for
    /* *Phase II (reconfigurable resources)* */
    Determine *Ns-max*(t) and $T(t, mix)$;
    if $(|T(t, mix)| + \beta \cdot Ns\text{-}max(t) > Na)$
      Rescheduling
    end if
    Determine candidate mixing operations $U(t, mix)$;
    Select $S(t, mix) \subseteq U(t, mix)$ based on the urgency priority list, such that
        $|S(t, mix)| + |T(t, mix)| + \beta \cdot (Ns\text{-}max(t) - 2|S(t, mix)|) \leqslant Na$;
    /* *The factor of 2 is necessary because that two droplets that mix free up two storage units*/
    set $t_i = t \ \forall i$: $v(i) \in S(t, mix)$
  $t = t + 1$;}
until (all operations are scheduled)
Record the schedule

Fig. 18. Pseudocode for the modified *list scheduling* algorithm.

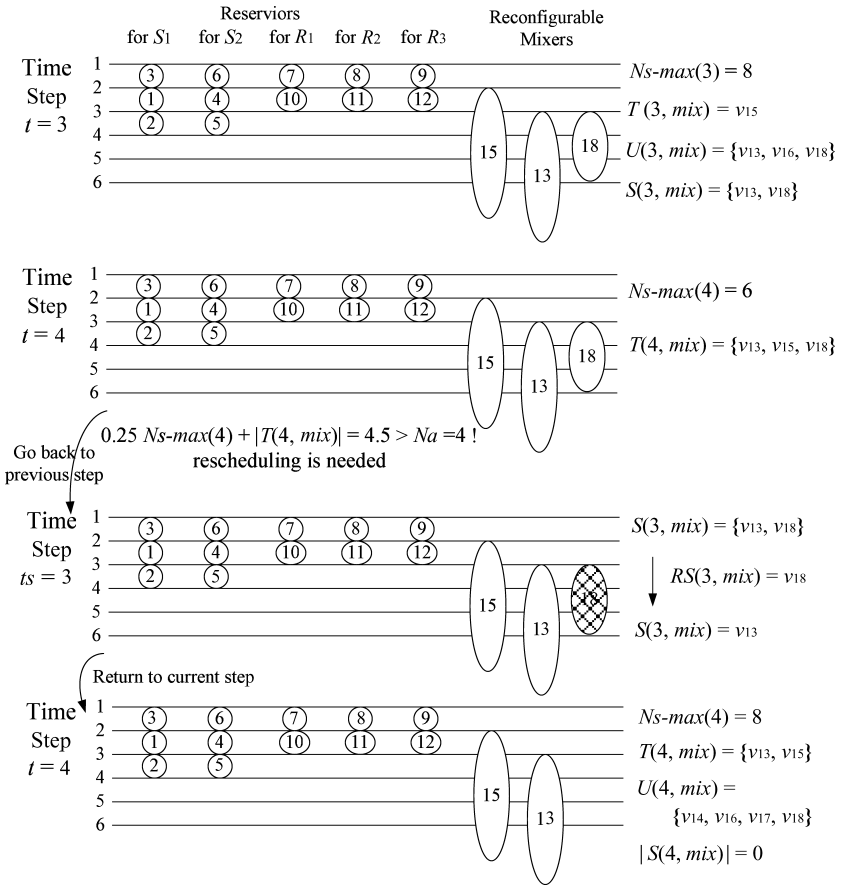

Fig. 19. An example to illustrate the necessity of rescheduling.

to the conventional list-scheduling algorithm, our modified method also introduces a *Rescheduling* step. In time step t, we determine the maximum number of reconfigurable storage units needed, $Ns\text{-}max(t)$, assuming no new mixing operation is scheduled. Unfinished mixing operation $T(t, mix)$ is also determined at this time step. If $(|T(t, mix)| + \beta \cdot Ns\text{-}max(t) > Na)$, we need rescheduling, that is, going back to the previous time step $ts = t - 1$, selecting the rescheduled mixing operations $RS(ts, mix) \subseteq \{S(ts, mix)$: scheduled mixing operations at $ts\}$ based on reversing priority list, and remove them from $S(ts, mix)$ to the ready list of time step t until $|T(t, mix)| + \beta \cdot Ns\text{-}max(t) \leq Na$. For example, for the sequencing graph of Figure 16 the reconfigurable resources necessitate rescheduling, as shown in Figure 19. Note that at time step 4 of the algorithm, we need to go back to the previous step to reschedule the mixing operations.

Heuristic Based on a Genetic Algorithm (GA). We next describe a second heuristic approach based on a genetic algorithm [Srinivas and Patnaik 1994; Bean 1994; Spears and Dejong 1991]. Optimization methods based on genetic algorithms have been extensively used to solve synthesis problems in electronic design automation. We choose a GA based on two reasons. First, we are

```
Procedure Genetic Algorithm-Based Heuristic Approach
1 Initialization: generate initial population P(t) of chromosomes.
2 for g = 1 to G (G: the number of evolution generations)
3     for (each chromosome of P(t))
4       Perform ad-hoc schedule construction procedure
5       Determine the fitness value of the chromosome
6         /* fitness value = – (completion time of the schedule generated from the above procedure). */
7       end for
8        /* Evolution procedure*/
9              Reproduction P(t)→ P(t+1);
10             Crossover P(t)→ P(t+1);
11             Mutation P(t)→ P(t+1);
12       /* Old population is replaced with new population*/
13           P(t) = P(t+1);
14 end for
15 Completion time = – (fitness value of the optimum chromosome).
```

Fig. 20. Pseudocode for Genetic Algorithm-based heuristic approach.

targeting a multiobjective optimization problem for which researchers have often used GAs by formulating the problem in terms of multipriority optimizations. The primary objective here is to minimize the bioassay completion time. A secondary goal is to optimize resource binding, in that number of functional units is minimized. Among various randomized search algorithms, GAs have been deemed in the literature to be appropriate for multiobjective optimization [Srinivas and Patnaik 1994; Aarts and Korst 1989]. Second, GA maintains a pool of solutions instead of a single solution and allows communication between solutions via crossover and mutation. In this way, GA is better equipped to escape the local minima and use information from previous moves. Results in Section 5 showed GA leads to lower completion time compared to the modified list-scheduling algorithm. The pseudocode for this GA-based heuristic approach is shown in Figure 20. The details of this procedure are as follows.

Representation of Chromosome. A robust representation technique called *random keys* is used in this algorithm [Bean 1994]. The important feature of random keys is that all offspring formed by crossover are feasible solutions. We interpret any random key vector as a feasible solution. Any crossover vector is also feasible. This is ensured using an *ad hoc* design of a schedule construction procedure. Each chromosome in the population can be encoded as a vector of random keys. In other words, $Chromosome = \{gene(1), \ldots, gene(k), gene(k+1), \ldots, gene(2k)\}$, where k is the number of operations, that is, $k = 4mn$ in our problem. Each gene is a random number sampled from [0, 1]. The first k genes are used to indicate the operation priorities, that is, priority value of operation $Pv(i) = gene(i), i = 1$ to k. The last k genes are used to determine the delay time of the operations, which is calculated as follows: delay value of operation $Dv(i) = d \times MaxDur \times gene(i + k), i = 1$ *to* k, where d is a constant. The parameter *MaxDur* denotes the maximum duration for all operations. It should be noted that these delay values are further modified in the schedule construction procedure, such that any random number $(gene(i + k))$ can be used to form the feasible solution.

Procedure Construct schedule of input operations

1 **For** each reservoir/dispensing port /* Determine the ready input operations first*/
2 $U = \{ v_i \in V$: v_i can be covered by this reservoir/dispensing port$\}$;
3 Sort the ready operations in descending order of their priority values, $Pv(i)$.
4 The operation v_f with the highest priority value is scheduled first.
Its start time $St(f) = 1 + Dv(f)$; stop time $Sp(f) = St(f) + 1$.
5 Remaining input operations are scheduled in order of their priority values, and their start time $St(i) = Sp(j) + Dv(i)$ /* $Sp(j)$ is the stop time of input operation v_j scheduled consecutively prior to v_i.*/
6 **end for**

Fig. 21. Pseudocode for Phase I: scheduling input operations.

Schedule Construction Procedure. The goal of this *ad-hoc* procedure is to construct a feasible schedule, that is, satisfying dependency and resource constraints, by using a vector of random numbers (chromosome). It consists of three phases: scheduling input operations, scheduling for mixing operations and scheduling optical detection operations. As an illustration, the construction procedure for Phase I is described in Figure 21, where we assume that $Nr = 1$.

Using these three phases, a feasible schedule satisfying both dependency constraints and resource constraints can be constructed by using any random key vector.

Evolution Strategy. In the genetic algorithm, reproduction and crossover operators tend to increase the quality of the populations and force convergence, while mutation opposes convergence and replaces genes lost during reproduction and crossover [Srinivas and Patnaik 1994]. There exist many different types of these operations in the literature. In our heuristic approach, these evolutionary operators are defined as follows:

Reproduction. The chromosomes that have the highest fitness value, that is, the smallest completion time of the generated schedule, in the current population are copied to the next generation.

Crossover. Parameterized uniform crossover is employed in our algorithm [Spears and Dejong 1991]. In this crossover procedure, two parent chromosomes are chosen randomly from the old population. Then gene(i) of their offspring in the new population is inherited (i.e., copied) from gene(i) of the father chromosome with the probability P (e.g., $P = 0.7$), and from the mother chromosome with the probability $1 - P$.

Mutation. The new chromosomes of the population are generated randomly to guarantee population diversity.

For our scheduling problem, there are $4mn$ operation nodes in the sequencing graph. In the GA-based heuristic, we set the number of chromosomes in the population to twice the number of operations, that is, $8mn$. During evolution, the mn best chromosomes are reproduced into the next generation. A total of $5mn$ chromosomes in the new population are the offsprings generated from the previous population. The remaining $2mn$ chromosomes are randomly generated. This proportion is fine-tuned through experiments. The transition between two consecutive generations is shown in Figure 22. After G

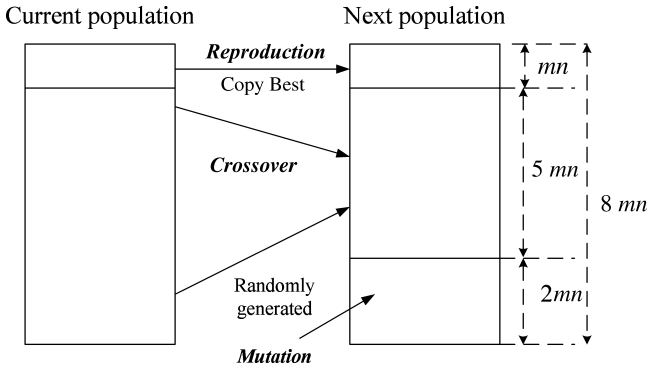

Fig. 22. Transition between consecutive generations during the genetic algorithm.

generations of evolution, we find the optimum chromosome with the best fitness value from the final population. A solution for the scheduling problem is the schedule constructed by using the optimum chromosome. An alternative stopping criterion is to check if the fitness value differences between several consecutive generations are smaller than some set value. We have not implemented it here, since the extension appears to be straightforward.

5. SIMULATIONS

In this section, we present simulation results to evaluate the two heuristic methods for large problem instances. First we present lower and upper bounds on the assay completion time. Figure 23 illustrates how these bounds are derived.

—*Lower Bound (LB).* For the ideal case shown in Figure 23(a), we obtain the following lower bound: $LB = m \times max\{d(D_1), \ldots d(D_n)\} + min\{d(M_1), \ldots d(M_m)\} + d(I_i) + 1$.

—*Upper Bound (UB).* Consider separating the operations into three sequencing phases. In each phase, only one type of operation (*Input*, *Mixing*, or *Detection*) can be performed. For this scheduling approach, we can estimate an upper bound as follows: $UB = m \times max\{d(D_1), \ldots d(D_n)\} + k \times max\{d(M_1), \ldots d(M_m)\} + max(m, n) \times d(I_i) + 1$; where k is the minimum value such that $NMix_1 + \cdots + NMix_k \geq mn$, $NMix_i$ is the maximum number of mixing operations that can be scheduled in step $i$of Phase II, and $NMix_i = \lfloor (2Na - mn + \beta \cdot (NMix_1 + NMix_2 + \cdots + NMix_{i-1})) \rfloor$, where $i > 1$; details shown in Figure 23(b) (where $\beta = 0.25$). The schedule obtained from this three-phase approach is a feasible solution but the corresponding completion time is an upper bound on the optimum completion time.

5.1 Evaluation Experiments

Five examples are used to evaluate the heuristics described previously. The details are presented in Table I. The modified *list scheduling* algorithm (M-LS) and *genetic algorithm*-based heuristic (GA) are applied to these five examples. The simulation results are shown in the Table II. For the smaller problem instances

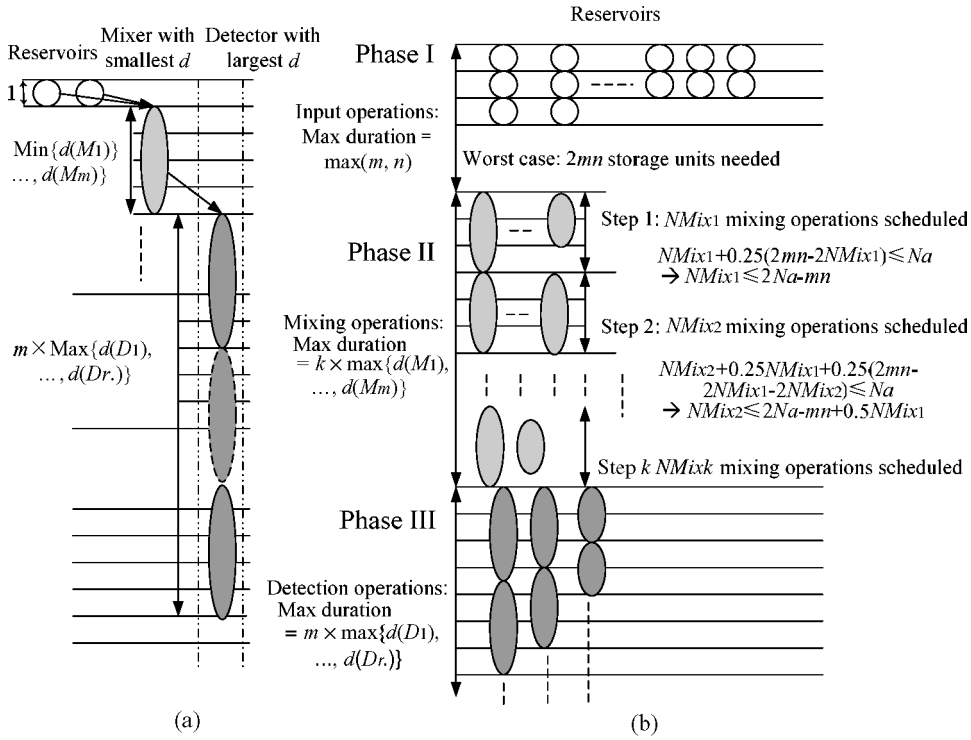

Fig. 23. Derivation of (a) lower and (b) upper bounds.

corresponding to Example 1 and Example 2, the optimal solutions have been obtained using the ILP model. (For the other three problem instances, the ILP model did not yield a solution within reasonable time, i.e., within 5 hours.) Upper bounds and lower bounds are also listed in the Table II. These experiments were performed on a server with two AMD Opteron 250 processors running at 2.4GHz with a 1MB L2 Cache, and 4GB of RAM. The GA procedure took 497 seconds of CPU time for Example 5. The CPU time was negligible for the smaller examples.

The results show that both M-LS and GA are able to generate good solutions, which are very close to the lower bounds. The ratio of the completion time obtained using the heuristic methods to the lower bound is no more than 1.2 in most cases, as shown in Figure 23. While GA yields lower completion times than M-LS, it requires $O(Gn^2)$ complexity compared to the $O(n)$ complexity for M-LS, where n is the number of operations and G is the number of generations used in the genetic algorithm.

5.2 Resource Selection

We next show that we can easily address the problem of resource selection using the heuristic based on the genetic algorithm. The modified *list scheduling* algorithm cannot be directly used to solve this problem. As indicated in Section 4, there exist several types of reconfigurable resources, for example, 2×2-array

Table I. Five Example Experiments (S_1: Plasma, S_2: Serum, S_3: Urine, S_4: Saliva, Assay1: Glucose assay, Assay2: Lactate assay, Assay3: Pyruvate assay, Assay4: Glutamate assay)

Example	Array Size (Number of) Cells and Dimensions)	Description	Node Weights for Mix Operations	Node Weights for Operations
Example 1 ($Nr = Nd = 1$ $Na = 3$) $m = 2$, $n = 2$	2×6	S_1 and S_2 are assayed for Assay1 and Assay2.	$d(M_1) = 5$ for S_1 $d(M_2) = 3$ for S_2	$d(D_1) = 5$ for Assay1 $d(D_2) = 4$ for Assay2
Example 2 ($Nr = Nd = 1$ $Na = 4$) $m = 2$, $n = 3$	2×8	S_1, and S_2 are assayed for Assay1, Assay2, and Assay3.	$d(M_1) = 5$ for S_1 $d(M_2) = 3$ for S_2	$d(D_1) = 5$ for Assay1 $d(D_2) = 4$ for Assay2 $d(D_3) = 6$ for Assay3
Example 3 ($Nr = Nd = 1$ $Na = 5$) $m = 3$, $n = 3$	2×10	S_1, S_2, and S_3 are assayed for Assay1, Assay2, and Assay3.	$d(M_1) = 5$ for S_1 $d(M_2) = 3$ for $S_2 d(M_3) = 4$ for S_3	$d(D_1) = 5$ for Assay1 $d(D_2) = 4$ for Assay2 $d(D_3) = 6$ for Assay3;
Example 4 ($Nr = Nd = 1$ $Na = 7$) $m = 3$, $n = 4$	2×14	S_1, S_2, and S_3 are assayed for Assay1, Assay2, Assay3 and Assay4.	$d(M_1) = 5$ for S_1 $d(M_2) = 3$ for $S_2 d(M_3) = 4$ for S_3	$d(D_1) = 5$ for Assay1 $d(D_2) = 4$ for Assay2 $d(D_3) = 6$ for Assay3 $d(\mathrm{D}_4) = 5$ for Assay4
Example 5 ($Nr = Nd = 1$ $Na = 9$) $m = 4$, $n = 4$	2×18	S_1, S_2, S_3 and S_4 are assayed for Assay1, Assay2, Assay3 and Assay4.	$d(M_1) = 5$ for S_1 $d(M_2) = 3$ for $S_2 d(M_3) = 4$ for S_3 $d(M_4) = 6$ for S_4	$d(D_1) = 5$ for Assay1 $d(D_2) = 4$ for Assay2 $d(D_3) = 6$ for Assay3 $d(\mathrm{D}_4) = 5$ for Assay4

Table II. Completion Time (in Time Units; 1 Time Unit = 2 Seconds)

Experiment	Opt	LB	UB	M-LS	GA
Example 1	15	15	23	17	15
Example 2	17	17	25	19	17
Example 3	N/A	23	47	26	25
Example 4	N/A	23	43	27	26
Example 5	N/A	29	59	35	34

mixers, 2×3-array mixers, and 2×4-array mixers, all of which can carry out mixing operations, but with different operation times. The mixing times for various mixers are listed in Table III; these times were obtained from lab experiments [Paik et al. 2003]. The selection of the appropriate type of mixer is an important problem in architectural-level synthesis. We can easily modify GA-based heuristic for the module selection problem. In this extended algorithm, *mn*

Table III. Mixing Times for Various Types of Mixers (in Time Units; 1 Time Unit = 2 Seconds)

Mixer Type	Mixing Time for Plasma Samples	Mixing Time for Serum Samples	Mixing Time for Urine Samples	Mixing Time for Saliva Samples
2×2-array mixer	7	5	6	8
2×3-array mixer	6	4	5	7
2×4-array mixer	5	3	4	6

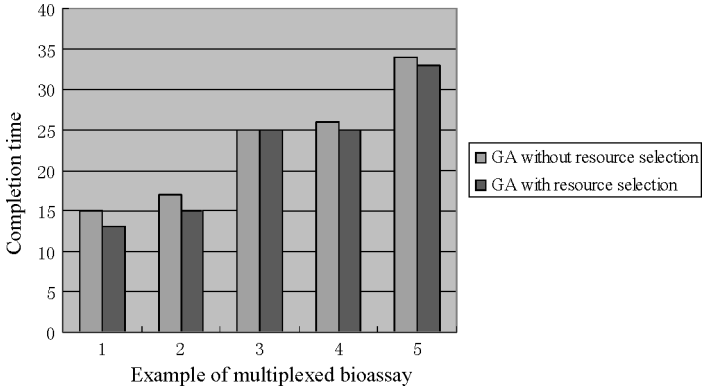

Fig. 24. Completion time for a set of multiplexed bioassays.

additional genes are added to the chromosome to denote the module selection information, that is, $Chromosome = \{gene(1), \ldots, gene(k), \ldots, gene(2k), gene(2k+1), \ldots, gene(2k+mn)\}$. During the *ad-hoc* schedule construction procedure, a mixer module is selected for the mixing operation v_{2mn+i} based on the corresponding $gene(2k+i)$, $i = 1, .., mn$. For example, a 2×4-array mixer is selected if $gene(2k+i) < 0.33$; a 2×3-array mixer is selected, if $gene(2k+i) > 0.67$; otherwise a 2×2-array mixer is selected.

Figure 24 shows the simulation results obtained from the GA-based heuristic algorithm. Note that module selection leads to a better solution, since an effective tradeoff between resource area and operation time can be obtained through careful resource selection.

5.3 Application to Protein Assay

Finally we evaluate the proposed GA-based heuristic method by using it to design a microfluidic array for a larger application, that is, a dilution-based protein assay. As in colorimetric glucose assays, the protocol for a protein assay based on the Bradford reaction [Srinivasan et al. 2004] also belongs to the generic class of droplet-based bioassay operations P_{assay} discussed in Section 4.1. Compared to the previous examples, there is a new type of operation, that is, dilution, that is used in a protein assay. Buffer droplets, such as 1M NaOH solution, are used to dilute the sample containing protein(s) to obtain a desired dilution factor (*DF*), before mixing with reagents droplets (e.g., Coomassie brilliant blue G-250 dye). This on-chip dilution is performed using multiple hierarchies of binary mixing/splitting phases, referred to as the interpolating serial dilution method [Fair et al. 2003]. The mixing of a sample droplet of protein

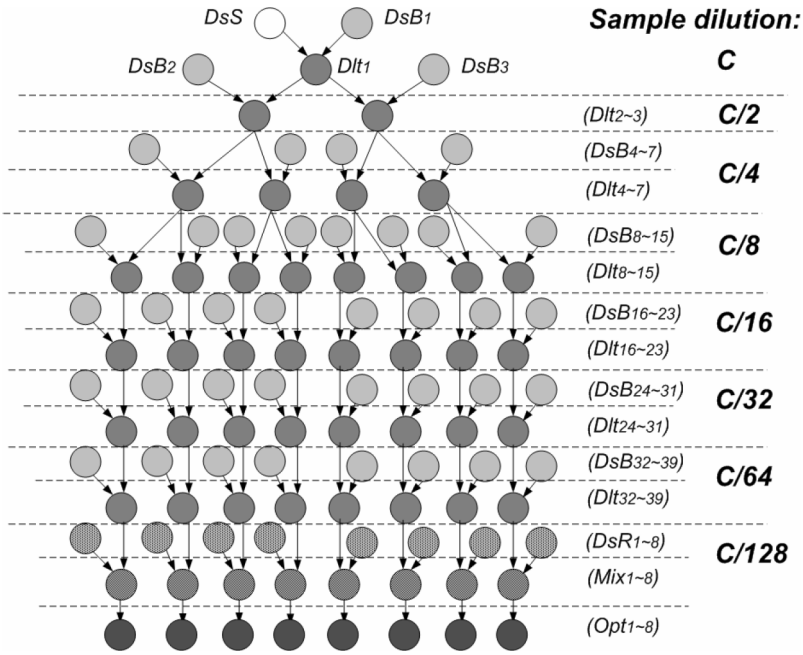

Fig. 25. Sequencing graph model of a protein assay.

concentration C and a unit buffer droplet results in a droplet with twice the unit volume, and concentration $C/2$. Splitting this large droplet results in two unit-volume droplets of concentration $C/2$ each. Continuing this step in a recursive manner using diluted droplets as samples, an exponential dilution factor of $DF = 2^N$ can be obtained in N steps.

A sequencing graph model can be developed from the protocol for a protein assay ($DF = 128$), as shown in Figure 25. There are a total of 103 nodes in one-to-one correspondence with the set of operations in a protein assay, where DsS, DsB_i ($i = 1, \ldots, 39$), and DsR_i ($i = 1, \ldots, 8$) represents the generation and dispensing of sample, buffer and reagent droplets, respectively. In addition, Dlt_i ($i = 1, \ldots, 39$) denotes the binary dilution (including mixing/splitting) operations, $Mix_i (i = 1, \ldots, 8)$ represents the mixing of diluted sample droplets, and reagent droplets; $Opt_i (i = 1, \ldots, 8)$ denotes the optical detection of the mixed droplets. Until the fourth step of a serial dilution, all diluted sample droplets are retained in the microfluidic array. After that stage, for each binary dilution step, only one diluted sample droplet is retained after splitting, while the other droplet is moved to the waste reservoir. The basic operations for protein assay have been implemented on a digital microfluidic biochip [Srinivasan et al. 2004-2] [Fair et al. 2003].

Table IV lists the available functional resources for the protein assay. We assume that there is only one on-chip reservoirs/dispensing port available for sample fluids, but two such ports for buffer fluids, two for reagent fluids, and one for waste fluids. We also assume that four optical detectors can be integrated into a 10×10 microfluidic array.

Table IV. Functional Resources for Synthesis

Operation	Resources	Time (s)
Dispensing: *DsS; DsB; DsR*	On-chip reservoir/dispensing port	7
Dilution: *Dlt*	2×4-array dilutor	5
Mixing: *Mix*	2×4-array mixer	3
Optical detection: *Opt*	LED+Photodiode	30
Storage	Single cell	N/A

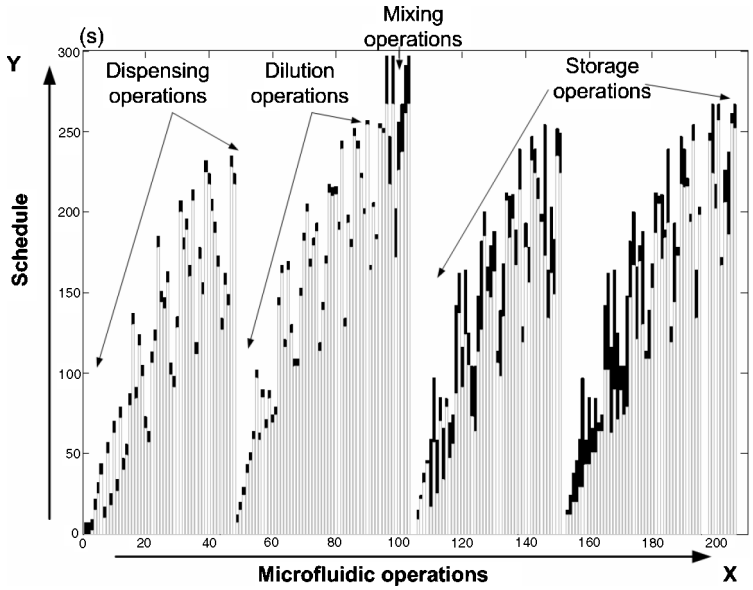

Fig. 26. Illustration of the schedule for the protein assay obtained by GA-based algorithm.

We now attempt to minimize the protein assay processing time using the GA-based heuristic proposed in Section 4.5. A completion time of 297 seconds for the protein assay is obtained using this method, with a CPU time of 15 minutes and 21 seconds. Figure 26 illustrates the scheduling result for this large problem instance. Each microfluidic operation involved in the protein assay is listed along the X-axis. The Y-axis denotes the scheduled times for the operations, that is, the start time and the stop time, for this digital microfluidics-based bioassay; the time-span for each operation is represented by a black rectangle. Note that in addition to the 103 operations represented by the sequencing graph in Figure 25, there are 99 more storage operations needed for this protein assay. These operations are not explicitly shown in Figure 25.

6. CONCLUSION

We have presented a system design methodology that attempts to apply classical architectural-level synthesis techniques to the design of digital microfluidic biochips. We have developed an optimal strategy based on integer linear programming for scheduling assay operations under resource constraints. However, because the scheduling problem is $\mathcal{NP}$-complete, we have also developed

two heuristic techniques that scale well for large problem instances. Two real-life biochemical assays, namely multiplexed *in-vitro* diagnostics and protein assays, have been used to evaluate the proposed methodology.

While the heuristic based on list scheduling is computationally more efficient, the second heuristic based on a genetic algorithm yields lower completion times for bioassays. The two methods appear to provide a trade-off between the quality of the results (assay completion time) and the CPU time. While the CPU time for the GA-based algorithm is reasonable for the larger protein assay, computation might be an issue for future applications, for example, protocols that require decisions based on the feedback from intermediate assay results.

The proposed synthesis approach is expected to reduce human effort and design cycle time, and it will facilitate the integration of fluidic components with microelectronic components in next-generation SOCs. Research on CAD tools for microfluidics system design is still in its infancy. Nevertheless, microfluidic biochips promise to emerge as a major application driver for continued research on synthesis techniques.

ACKNOWLEDGMENTS

We thank Tao Xu of Duke University for helping with the simulations.

REFERENCES

AARTS, E. AND KORST, J. 1989. *Simulated Annealing and Boltzmann Machines*. John Wiley.

ADVANCED LIQUID LOGIC, INC. www.liquid-logic.com.

ANTAO, B. AND BRODERSEN, A. 1992. Techniques for synthesis of analog integrated circuits. *IEEE Des. Test Comput. 9*, 8–18.

BEAN, J. C. 1994. Genetics and random keys for sequencing and optimization. *ORSA J. Comput. 6*, 154–160.

BERKELAAR, M. lpsolve. Eindhoven University of Technology, Eindhoven, Netherlands. ftp://ftp.ics.ele.tue.nl/pub/lp_solve.

BÖHRINGER, K. F. 2004. Towards optimal strategies for moving droplets in digital microfluidic systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 1468–1474.

BURNS, M. A., JOHNSON, B. N., BRAHMASANDRA, S. N., AND HANDIQUE, K. 1998. An integrated nanoliter DNA analysis device. *Science 282*, 484–487.

CAMPOSANO, R. 1996. Behavioral synthesis. In *Proceedings of the IEEE/ACM Design Automation Conference*, 33–34.

CHO, S. K., FAN, S. K., MOON, H., AND KIM, C. J. 2002. Toward digital microfluidic circuits: Creating, transporting, cutting and merging liquid droplets by electrowetting-based actuation. In *Proceedings of the IEEE Conference on Micro-Electro-Mechanical Systems.* 32–52.

COVENTORWARE. Micro-electro-mechanical systems. http://www.coventor.com.

DE, S. K. AND ALURU, N. R. 2003. Physical and reduced-order dynamic analysis of MEMS. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. 270–273.

DING, J., CHAKRABARTY, K., AND FAIR, R. B. 2001. Scheduling of microfluidic operations for reconfigurable two-dimensional electrowetting arrays. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Sys. 20*, 1463–1468.

FAIR, R. B., KHLYSTOV, A., TAYLOR, T. D., IVANOV, V., KVANS, R. D., SRINIVASAN, A., PAMOLA, V. K., POLLOCK, M. G., GRIFFIN, P. B., AND ZHOU, J. 2006. Chemical and biological applications of digital microfluidic devices. *IEEE Des. Test Comput*. 24, 10–24.

FAIR, R. B., KHLYSTOV, A., SRINIVASAN, V., PAMULA, V. K., AND WEAVER, K. N. 2004. Integrated chemical/biochemical sample collection, pre-concentration, and analysis on a digital microfluidic lab-on-a-chip platform. In *Proceedings of the SPIE Lab-on-a-Chip: Platforms, Devices, and Applications*. L. A. Smith and D. Sobek, Eds. 5591, 113–124.

FAIR, R. B., SRINIVASAN, V., PAIK, P., REN, H., AND PAMULA, V. K. 2003. Electrowetting-based on-chip sample processing for integrated microfluidics. In *Proceedings of the IEEE International Electronic Devices Meeting (IEDM)*. 32.5.1–32.5.4.

FEDDER, G. K. AND Q. JING, Q. 1999. A hierarchical circuit-level design methodology for micro-electromechinal system. *IEEE Trans. Circ. Syst. II. 46*, 1309–1315.

GALLARDO, B. S., GUPTA, V. K., EAGERTON, F. D., JONG, L. I., CRAIG, V. S., SHAH, R. R., AND ABBOTT, N. L. 1999. Electrochemical principles for active control of liquids on submillimeter scales. *Science 283*, 57–60.

GAREY, M. R. AND JOHNSON, D. S. 1979. *Computers and Intractability—A Guide to the Theory of NP-Completeness*. Freeman, New York, NY.

GRIFFITH, E. AND AKELLA, S. 2004. Coordinating multiple droplets in planar array digital microfluidics systems. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*.

HULL, H. F., DANILA, R., AND EHRESMANN, K. 2003. Smallpox and bioterrorism: Public-health responses. *J. Lab. Clin. Medicine 142*, 221–228.

ICHIMURA, K., OH, S., AND NAKAGAWA, M. 2000. Light-driven motion of liquids on a photoresponsive surface. *Science* 288, 1624–1626.

INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS (ITRS). http://public.itrs.net/Files/2003ITRS/Home2003.htm.

JONES, T. B., GUNJI, M., WASHIZU, M., AND FELDMAN, M. J. 2001. Dielectrophoretic liquid actuation and nanodroplet formation. *J. Appl. Phys. 89*, 1441–1448.

KAHNG, B., MANDOIU, I. I., REDA, S., XU, X., AND ZELIKOVSKY, A. 2003. Evaluation of placement techniques for DNA probe array layout. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. 262–269.

KWOK, Y. AND AHMAD, I. 1999. Static scheduling algorithms for allocating directed task graphs to multiprocessors. *ACM Comput. Surv. 31*, 406–471.

MICHELI, G. DE. 1994. *Synthesis and Optimization of Digital Circuits.* McGraw-Hill, New York, NY.

MUKHERJEE, T. AND FEDDER, G. K. 1998. Design methodology for mixed-domain systems-on-a-chip [MEMS design]. In *Proceedings of the IEEE Conference on VLSI System Level Design*. 96–101.

PAIK, P., PAMULA, V. K., AND FAIR, R. B. 2003. Rapid droplet mixers for digital microfluidic systems. *Lab on a Chip 3*, 253–259.

PAMULA, K., SRINIVASAN, V., CHAKRAPANI, H., FAIR, R. B., AND TOONE, E. J. 2005. A droplet-based lab-on-a-chip for colorimetric detection of nitroaromatic explosives. In *Proceedings of the IEEE Conference on Micro-Electro-Mechanical Systems*. 722–725.

PFEIFFER, J., MUKHERJEE, T., AND HAUAN, S. 2004. Simultaneous design and placement of multiplexed chemical processing systems on microchip. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. 229–236.

POLLACK, M. G. 2001. Electrowetting-based microactuation of droplets for digital microfluidics. Ph.D. thesis, Duke University.

POLLACK, M. G., FAIR, R. B., AND SHENDEROV, A. D. 2000. Electrowetting-based actuation of liquid droplets for microfluidic applications. *Appl. Phys. Lett. 77*, 1725–1726.

POLLACK, M. G., SHENDEROV, A. D., AND FAIR, R. B. 2002. Electrowetting-based actuation of droplets for integrated microfluidics. *Lab on a Chip 2*, 96–101.

REN, H. AND FAIR, R. B. 2002. Micro/nano liter droplet formation and dispensing by capacitance metering and electrowetting actuation. *Technical Digest of IEEE-NANO*, 36–38.

RICKETTS, J. IRICK, K. VIJAYKRISHNAN, N. AND IRWIN, M. J. 2006. Priority scheduling in digital microfluidics-based biochips. In *Proceedings of the Conference on Design, Automation and Test in Europe*. 329–334.

SAMMARCO, T. S. AND BURNS, M. A. 1999. Thermocapillary pumping of discrete droplets in microfabricated analysis devices. *Amer. Inst. Chem. Engin. J. 45*, 350–366.

SCHULTE, T. H., BARDELL, R. L., AND WEIGL, B. H. 2002. Microfluidic technologies in clinical diagnostics. *Clinica Chimica Acta 321*, 1–10.

SENTURIA, S. 1987. Microfabricated structures for the measurement of mechanical properties and adhesion of thin films. In *Proceedings of the International Conference on Sold-State Sensors and Actuators (Transducers)*. 11–16.

SHAPIRO, H., MOON, H., GARRELL, R., AND KIM, C. J. 2003. Modeling of electrowetted surface tension for addressable microfluidic systems: dominant physical effects, material dependences, and limiting phenomena. In *Proceedings of the IEEE Conference on Micro-Electro-Mechanical Systems*. 201–205.

SILICON BIOSYSTEMS. What is DEPArray™? http://www.siliconbiosystems.com/technology/DEPArray.htm.

SPEARS, W. M. AND DEJONG, K. A. 1991. On the virtues of parameterized uniform crossover. In *Proceedings of the International Conference on Genetic Algorithms*. 230–236.

SRINIVAS, M. AND PATNAIK, L. M. 1994. Genetic algorithms: a survey. *IEEE Comput. 27*, 17–26.

SRINIVASAN, V., PAMULA, V. K., AND FAIR, R. B. 2004. An integrated digital microfluidic lab-on-a-chip for clinical diagnostics on human physiological fluids. *Lab on a Chip*, 310–315.

SRINIVASAN, V., PAMULA, V. K., PAIK, P., AND FAIR, R. B. 2004. Protein stamping for MALDI mass spectrometry using an electrowetting-based microfluidic platform. In *Proceedings of the SPIE*. 5591, 26–32.

SRINIVASAN, V., PAMULA, V. K., POLLACK, M. G., AND FAIR, R. B. 2003. Clinical diagnostics on human whole blood, plasma, serum, urine, saliva, sweat, and tears on a digital microfluidic platform. In *Proceedings of the International Conference on Micro Total Analysis Systems*. 1287–1290.

SU, F. AND CHAKRABARTY, K. 2006. Module placement for fault-tolerant microfluidics-based biochips. *ACM Trans. Des. Automat. Elect. Syst. 11*, 682–710.

SU, F., CHAKRABARTY, K., AND FAIR, R. B. 2006. Microfluidics-based biochips: technology issues, implementation platforms, and design automation challenges. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst. 25*, 211–223.

SU, F., HWANG, W. AND CHAKRABARTY, K. 2006. Droplet routing in the synthesis of digital microfluidic biochips. In *Proceedings of the Design, Automation and Test in Europe (DATE) Conference*. 323–328.

SU, F., OZEV, S., AND CHAKRABARTY, K. 2004. Concurrent testing of droplet-based microfluidic systems for multiplexed biomedical assays. In *Proceedings of the IEEE International Test Conference.* 883–892.

SU, F., OZEV, S., AND CHAKRABARTY, K. 2005. Ensuring the operational health of droplet-based microelectrofluidic biosensor systems. *IEEE Sensors 5*, 763–773.

THORSEN, T., MAERKL, S., AND QUAKE, S. 2002. Microfluidic large-scale integration. *Science 298*, 580–584.

VAN DER WOERD, M., FERREE, D., AND PUSEY, M. 2003. The promise of macromolecular crystallization in microfluidic chips. *J. Struct. Biol. 142*, 180–187.

VENKATESH, S. AND MEMISH, Z. A. 2003. Bioterrorism: A new challenge for public health. *Int. J. Antimicrobial Agents 21*, 200–206.

VERPOORTE, E. AND DE ROOIJ, N. F. 2003. Microfluidics meets MEMS. In *Proceedings of the IEEE.* 91, 930-953.

WALKER, R. A. AND CAMPOSANO, R. 1991. *A Survey of High-Level Synthesis Systems*. Kluwer Academic Publishers, Boston, MA.

WASHIZU, M. 1998. Electrostatic actuation of liquid droplets for microreactor applications. *IEEE Trans. Indus. Appl. 34*, 732–737.

ZENG, J. AND KORSMEYER, F. T. 2004. Principles of droplet electrohydrodynamics for lab-on-a-chip. *Lab on a Chip 4*, 265–277.

ZHANG, T., CHAKRABARTY, K., AND FAIR, R. B. 2002. *Microelectrofluidic Systems: Modeling and Simulation*. CRC Press, Boca Raton, FL.

ZHAO, Y. AND CHO, S. K. 2006. Microparticle sampling by electrowetting-actuated droplet sweeping. *Lab on a Chip 6*, 137–144.

Received February 2007; revised April 2007; accepted May 2007