

# Enrutamiento de microfluidos en biochips digitales. Un enfoque basado en colonia de hormigas multiobjetivas

Carlos Mendoza,  
Facultad Politécnica  
Universidad Nacional de Asunción  
San Lorenzo, Paraguay  
Email: carlos.mendoza@pol.una.py

Eduardo Szaran  
Facultad Politécnica  
Universidad Nacional de Asunción  
San Lorenzo, Paraguay  
Email: eduardo.szaran@pol.una.py

**Resumen**—Los biochips digitales de microfluidos (DMFB) se usan para realizar ensayos biológicos concurrentemente, tales como análisis de ADN, reconocimiento de sustancias, descubrimiento de drogas, detección in-situ y en tiempo real de toxinas, y agentes patógenos. Son muy prometedores por la portabilidad y por la flexibilidad que ofrece su configurabilidad para realizar una gran variedad de ensayos concurrentes. Uno de los retos más importantes en el diseño de los DMFB es el problema del enrutamiento de los microfluidos, en el cual se debe calendarizar el movimiento de cada gota en el tiempo de manera simultánea. Este trabajo trata el problema de enrutamiento de microfluidos como un problema de optimización multiobjetivo no Pareto. Se propone una heurística basada en optimización por colonia de hormigas, que optimiza en orden lexicográfico la interferencia entre gotas, el tiempo total de viaje y el número de celdas utilizadas. Los resultados experimentales indican que la propuesta es promisoría al mejorar los resultados de un trabajo del estado del arte [11] basado en colonia de hormigas.

## I. INTRODUCCIÓN

Lab On Chip o LOC [5] es un concepto que en los últimos años obtuvo mucho interés porque se ha convertido en una alternativa a los laboratorios tradicionales. Aplicaciones prometedoras de esta tecnología emergente incluyen la secuenciación de alto rendimiento de ADN, inmunoensayos, monitoreo de toxicidad ambiental, y el punto de atención para el diagnóstico de enfermedades. El dispositivo DMFB realiza manipulaciones de microfluidos, usualmente a escalas de nano-litros o micro-litros. Está compuesto de dos placas paralelas de cristal; entre éstas se encuentran intercalados los microfluidos de muestras bioquímicas y el relleno, como aceite silicón. Los microfluidos son controlados de manera independiente mediante variaciones de potencia eléctrica a lo largo de un arreglo lineal de electrodos, esta técnica es conocida como electrohumectación. El bioensayo en un DMFB consiste en varias operaciones que son aplicadas a los microfluidos. Debido a la flexibilidad de las operaciones y a las propiedades de los microfluidos, su manipulación se vuelve un problema en el

diseño físico de la ruta de encaminamiento [2] en las operaciones de transporte. El principal reto en la planificación de las rutas es asegurar que el bioensayo sea correcto, evitando mezclas inesperadas entre las gotas analizadas, por lo que el procesamiento de un gran número de gotas simultáneamente en un microchip integrado DMFB puede llegar a convertirse en un problema NP-Completo [6].

En este trabajo se tratan los problemas relacionados al transporte de las gotas, y se supone un modelo de dispositivo que se abstrae de los detalles de implementación física. Se propone un algoritmo utilizando la meta-heurística de colonia de hormigas multiobjetivos, basado en el ordenamiento lexicográfico. La propuesta se basa en el algoritmo Max-Min Ant System (MMAS) [1] de colonia de hormigas.

El objetivo principal es mejorar el tiempo total y el número total de celdas que son activadas durante el enrutamiento de los microfluidos, tratando al mismo tiempo de disminuir las interferencias entre las mismas.

El resto de este trabajo está organizado de la siguiente manera. En la sección II se describe el enrutamiento de los microfluidos en un DMFB, y también son discutidos los trabajos relacionados. En la sección III se presenta la propuesta de trabajo haciendo una introducción a las colonias de hormigas y a los diferentes algoritmos derivados. En la sección IV se analiza y se comparan los resultados del trabajo, para luego culminar con el análisis de los resultados en la sección V y una conclusión en la sección VI.

## II. ENRUTAMIENTO DE MICRO-FLUIDOS

### A. Formulación del Problema

El DMFB consiste en una matriz  $M$  de  $m \times n$  celdas. Cada celda de la matriz puede estar vacía, bloqueada u ocupada por una gota. Se considera  $G$  como el conjunto de gotas del sistema, tal que  $G = \{g^1, \dots, g^K\}$  donde  $K$  es el número de gotas en el sistema, y  $g^k$  representa a la  $k$ -ésima gota.

El conjunto  $C$  es la configuración del estado de una celda, tal que  $C = \{\emptyset, G, B\}$  donde  $\emptyset$  es una celda vacía,

$G$  es una gota ocupando la celda, y  $B$  es una celda bloqueada. Por lo tanto el DMFB se puede describir como  $M(x, y) = C(x, y)$ , tal que  $C(x, y) = \{\exists \emptyset \vee \exists G \vee \exists B\}$ .

Se asume el tiempo como un contador discreto,  $T = \{t | t \in \mathbb{Z}\}$ . Las transiciones en la matriz DMFB se producen en pasos de unidades de tiempo entero de  $t$  a  $t + \Delta t$ , donde  $\Delta t = 1$ .

El conjunto solución está definido por  $S = \{s^1, \dots, s^K\}$ , donde:

$$s^k = \{g_{x,y}^{k,t} \mid g_{x,y}^{k,t} \neq g_{x,y}^{k,t+1} \vee g_{x,y}^{k,t} = g_{x,y}^{k,t+1}\} \quad (1)$$

La expresión (1) representa la ruta conformada por celdas  $(x, y)$  ocupadas por la gota  $k$  en el tiempo  $t$ . Se toma  $x$  como la posición horizontal, e  $y$  como la posición vertical en la matriz. Por ejemplo, si la gota  $g^1$  representa "agua desionizada", entonces  $g_{2,3}^{1,1}$  es la gota desionizada en la posición 2, 3 de la matriz y en el instante de tiempo 1.

Una gota cuando se mueve siempre lo hace a una celda adyacente de manera vertical u horizontal, tal que si  $g_{x,y}^{k,t}$  se mueve en el siguiente tiempo  $g_{x',y'}^{k,t+1}$  se cumple que  $|x - x'| + |y - y'| = 1$ , es decir,  $M(x, y)$  y  $M(x', y')$  son directamente adyacentes. En el tiempo  $t$ ,  $M(x, y) = G$  y  $M(x', y') = \emptyset$ , luego en el tiempo  $t + \Delta t$ ,  $M(x, y) = \emptyset$  y  $M(x', y') = G$ .

Se escribe  $M_t$  para referirse al estado de la matriz en un momento específico  $t$ .

Las restricciones del fluido pueden ser estáticas o dinámicas. Dadas dos gotas  $g_{x,y}^{k,t}$  y  $g_{x',y'}^{k',t}$  independientes en el tiempo  $t$ , entonces las restricciones del fluido que deben ser satisfechas en cada intervalo de tiempo  $t$  durante el ruteo, se describe como sigue:

- Restricción estática:  $|x_k^t - x_{k'}^t| > 1 \vee |y_k^t - y_{k'}^t| > 1$
- Restricción dinámica:  $|x_k^{t+1} - x_{k'}^t| > 1 \vee |y_k^{t+1} - y_{k'}^t| > 1 \vee |x_k^t - x_{k'}^{t+1}| > 1 \vee |y_k^t - y_{k'}^{t+1}| > 1$

Para resolver el problema propuesto se deben minimizar el tiempo máximo de llegada de la última gota y la cantidad de celdas utilizadas durante el enrutamiento. Minimizar el tiempo máximo  $T_{max} = \omega\{s^k(T_k)\}$ , donde  $\omega$  es la función de minimización y  $T_k$  es el tiempo máximo empleado por la gota  $k$  para completar el enrutamiento a su celda destino. Y, minimizar la cantidad de celdas utilizadas durante el enrutamiento  $A = \{\omega[M(t, x, y)] \mid t \in \{1, \dots, T_{max}\} \wedge M(t, x, y) = G\}$ . Se cuentan las celdas utilizadas como las celdas que se activaron para que la gota pueda realizar el movimiento, tal que una gota que permanece en el tiempo no cuenta sino solo la vez que pasó a la celda actual y se cumple que  $\{\forall g_{x,y}^{k,t} \in M_t \cup M_{t+1} \Rightarrow g_{x,y}^{k,t} \neq g_{x,y}^{k,t+1}\}$ .

### B. Trabajos Relacionados

Böhringer en [7] propone el algoritmo A\*, el cual calcula la solución óptima pero no es escalable cuando el problema crece en tamaño.

En [6], nuevamente Böhringer adopta un enfoque que permite a la gota elegir libremente los caminos (a excepción de las restricciones estáticas). Su enfoque asigna

prioridades a cada gota y genera rutas sucesivamente, comenzando con la gota de más alta prioridad. El algoritmo no asegura completitud, y las soluciones generadas dependen de la clasificación de prioridad de las gotas y puede no ser óptima.

El algoritmo High-Performance propuesto por M. Cho y D. Z. Pan en [8] usa una métrica denominada *bypassability* para estimar la degradación de la ruta mientras la gota está siendo enrutada. También usa el concepto de zona de concesión, en donde alguna gota puede migrar para romper un estancamiento cuando exista un bloqueo entre gotas. Sólo la gota elegida por la métrica *bypassability* se encamina mientras que las otras son congeladas.

En [4], Huang y Ho proponen un algoritmo eficaz, Fast-Routability, con las siguientes características: un análisis global del vector de movimiento para la construcción de pistas de enrutamiento para minimizar el número de celdas utilizadas, una ecuación basada en la entropía para determinar el orden de enrutamiento de las gotas, y una técnica de compactación de rutas basada en programación dinámica para reducir al mínimo el tiempo de llegada de la última gota.

En [11] se propone la utilización de Ant Colony Optimization (ACO). El algoritmo funciona en dos fases. En la primera traza las probables rutas para todas las gotas utilizando la distancia de Manhattan, y en la segunda fase resuelve los problemas de concurrencia y las limitaciones de los fluidos. Para generar las soluciones con menor tiempo de enrutamiento se aplica una técnica de movimiento restringido limitando la optimalidad de las soluciones en problemas de mediano y gran tamaño.

En [12] proponen una estrategia de optimización multi objetivo utilizando un algoritmo genético, específicamente el NSGA-II. El tiempo máximo de completamiento y el número total de celdas utilizadas son minimizadas simultáneamente.

Todos estos algoritmos no aplican ninguna restricción durante la selección de celdas para la ruta de cada gota. Basados en las explicaciones anteriores, en este trabajo se propone un algoritmo utilizando la meta-heurística de colonia de hormigas multiobjetivos, basado en el ordenamiento lexicográfico. La propuesta se basa en el algoritmo Max-Min Ant System (MMAS) [1] de colonia de hormigas aplicando las restricciones durante el enrutamiento a través de las diversas visibilidades.

## III. PROPUESTA

### A. Ordenamiento por entropía

La definición más elemental de entropía es el grado o medida de desorden que tiene un sistema. Se trata de una medida de la información que es necesaria para reducir o eliminar la incertidumbre.

Según la descripción en [4] una de las claves del enrutamiento de las gotas es el orden en que estas se enrutan. Si las gotas se enrutan sin orden alguno se producen

problemas de ruteo que aumentan la complejidad del problema.

Para resolver esto, en [4] se propone una ecuación basada en la Entropía la cual considera la congestión de la región de ruteo. La ecuación de la variación de Entropía de una gota está dada por:

$$\Delta Q_{g^k} = ES_{g^k} - E_{g^k}^{neg} + E_{g^k}^{pos} \quad (2)$$

La entropía (2) básicamente describe la energía de la región enrutable de una gota, donde  $ES_{g^k}$  es la energía del sistema de la gota  $g^k$ .  $E_{g^k}^{neg}$  es la energía negativa en  $ES_{g^k}$ , y está dada por la sumatoria de las celdas bloqueadas y cantidad de celdas fuentes (incluidas sus celdas adyacentes) en esa región.  $E_{g^k}^{pos}$  es la energía positiva en  $ES_{g^k}$  dada por la sumatoria de la energía de las celdas destino (cantidad de celdas destino y sus adyacentes) en la región de la gota  $g^k$ .

### B. Colonia de Hormigas

La teoría de colonias de hormigas está basada en el comportamiento de las hormigas reales, las cuales en busca de un camino óptimo entre su colonia y el alimento encuentran siempre el camino más corto [9].

Los algoritmos basados en colonias de hormigas artificiales son programas constructivos que aplican políticas estocásticas: en cada iteración cada hormiga construye una solución al problema y tiene asociada dos parámetros: la información de los rastros de feromonas y la información heurística o visibilidad.

La información de los rastros de feromonas ( $\tau$ ) mide la deseabilidad aprendida en el movimiento de un estado a otro, lo cual busca imitar la feromona real que depositan las hormigas naturales, esta es modificada por cada hormiga mientras que se ejecuta el algoritmo dependiendo de las soluciones encontradas; y el parámetro de información heurística o visibilidad ( $\eta$ ) mide la preferencia heurística que tienen las hormigas para moverse de un estado a otro.

### C. Propuesta basada en MMAS para DMFB

Este trabajo propone la meta-heurística ACO-DMFB basada en el algoritmo MMAS, descrito en [1] para el enrutamiento de microfluidos en biochips. En ACO-DMFB la restricción de adyacencia de gotas es tratada como un objetivo del problema, de forma a darle mayor libertad de movimiento a la hormiga y ampliando la exploración del espacio solución.

En este contexto, los objetivos del problema son minimizar:

- 1) La cantidad de adyacencias entre las gotas  $F_1$
- 2) El tiempo máximo de enrutamiento  $F_2$
- 3) La cantidad total de celdas utilizadas  $F_3$

Note que ACO-DMFB busca la solución considerando una prioridad para el desempate de soluciones entre los objetivos.

Para el orden se estableció como primer objetivo el

número de adyacencias ( $F_1$ ), el segundo objetivo es el tiempo total de ruteo ( $F_2$ ) y el tercer objetivo es el número total de celdas activadas ( $F_3$ ). De esta manera se prioriza que no deben existir adyacencias, para luego priorizar los otros dos objetivos.

### D. Visibilidad

Para darle información local a las hormigas se utiliza la visibilidad y en este trabajo se proponen dos. La primera visibilidad ( $\eta_1$ ) informa cual es la celda que se encuentra más cerca a la celda destino; mientras que, la segunda visibilidad ( $\eta_2$ ) informa cual es la celda que tiene menor posibilidad de caer en una adyacencia. Cabe destacar que en el ACO convencional se utiliza una sola visibilidad debido a que se resuelven problemas de optimización mono objetivo. El problema planteado es de naturaleza multiobjetivo por lo que cada función objetivo tiene asociada una visibilidad en los ACOs multiobjetivos.

Entonces  $\eta_1$  está dado por:

$$\eta_{1,i,j}^k = \frac{1}{\text{dist}(c_j, O_k)} \quad (3)$$

La visibilidad dada en (3) indica, para la gota  $k$ , que minimiza la distancia entre el siguiente estado ( $c_j$ ) y la celda destino ( $O_k$ ) de la gota  $k$ .

Para  $\eta_2$  se aplican pesos a los estados según ciertas condiciones que evalúan el estado actual de las gotas. Cuanto más peso posee un estado, es menos probable que sea elegido.

Evaluando el estado actual de la gota ( $g^{k,t}$ ) contra el estado de las gotas de orden mayor ( $g^{k',t}$ ), tal que  $g^{k',t}$  son las gotas que todavía no han ejecutado su movimiento, el peso que se le asigna está dado por (4).

$$\eta_2 = \begin{cases} 1 & \text{si } \text{dist}(g^{k',t}) > \text{dist}(g^{k,t}) \\ & \text{y } \text{ady}(g^{k,t}, g^{k',t+1}) \\ 2 & \text{si } \text{dist}(g^{k',t}) > \text{dist}(g^{k,t}) \\ & \text{y } \text{ady}(g^{k,t}, g^{k',t+1}) \\ & \text{y } g^{k,t} = g^{k,t-1} \\ & \text{y } \text{ady}(g^{k,t}, g^{k',t+2}) \end{cases} \quad (4)$$

Evaluando el estado actual de la gota ( $g^{k,t}$ ) contra el estado de las gotas de orden menor ( $g^{k'}$ ), tal que  $g^{k'}$  son las gotas que ya han ejecutado su movimiento, el peso que se le asigna está dado por (5).

$$\eta_2 = \begin{cases} 1 & \text{si } \text{ady}(g^{k,t}, g^{k',t}) \\ 2 & \text{si } \text{ady}(g^{k,t}, g^{k',t}) \\ & \text{y } \text{ady}(g^{k,t}, g^{k',t+1}) \\ 3 & \text{y } \text{ady}(g^{k,t}, g^{k',t}) \\ & \text{y } \text{ady}(g^{k,t}, g^{k',t+1}) \\ & \text{y } g^{k,t} = g^{k,t-1} \\ & \text{y } \text{ady}(g^{k,t}, g^{k',t+2}) \\ & \text{si } \text{dist}(g^{k',t}) > \text{dist}(g^{k,t}) \end{cases} \quad (5)$$

La función  $\text{ady}(g^{k,t}, g^{k',t})$  determina si la gota  $g^{k,t}$  es adyacente a la gota  $g^{k',t}$ . La condición  $g^{k,t} = g^{k,t-1}$

especifica que el siguiente estado de la gota  $g^{k,t}$  es su estado actual, es decir que se está evaluando permanecer en ese estado. Y la función  $dist(g^{k',t}) > dist(g^{k,t})$  evalúa la distancia entre las gotas y sus celdas destino respectivos, dando prioridad a la gota que tiene mayor distancia a su celda destino.

#### E. Actualización de feromonas

La actualización y evaporación de feromonas se realiza manteniendo el rango especificado en [1] según la siguiente fórmula:

$$\tau_{ij} = \begin{cases} \rho \cdot \tau_{ij} + \Delta\tau & \text{si } \tau_{min} < \tau_{ij} < \tau_{max} \\ \tau_{max} & \text{si } > \tau_{max} \\ \tau_{min} & \text{si } < \tau_{min} \end{cases} \quad (6)$$

donde,

$$\tau_{max} = \frac{\Delta\tau}{1 - \rho} \quad (7)$$

$$\tau_{min} = \frac{\Delta\tau}{2 \cdot m \cdot (1 - \rho)} \quad (8)$$

donde,  $\rho$  es el coeficiente de evaporación,  $m$  es la cantidad de hormigas, y

$$\Delta\tau = \frac{1}{\sum_{i=1}^3 \vec{f}_i(\vec{S})} \quad (9)$$

En la fórmula (6) el estado  $i$  equivale a la posición en la celda  $(x_i, y_i)$  en el tiempo  $t$ , y el estado  $j$  equivale a la posición en la celda  $(x_j, y_j)$  en el tiempo  $t + 1$ . Y la variación de feromonas descrita en la fórmula (9) está dada por la sumatoria de las funciones objetivo en un contexto de minimización. Las funciones objetivos se normalizan previamente dividiéndolas por un valor predefinido.

#### F. Probabilidad de selección de estado

En este trabajo se da a la hormiga la posibilidad de seleccionar un estado anterior o el estado actual sin caer en ciclos o estancamiento mediante la utilización de la visibilidad; además, cada sub colonia tiene su propia tabla de feromonas. Por lo tanto la función de probabilidad para la selección de estados de la gota  $k$  en la transición  $i \rightarrow j$  y en el tiempo  $t$  es la siguiente:

$$P_{i,j}^k = \frac{[\tau_{i,j}^k]^\alpha \cdot [\eta_{1,i,j}^k \cdot \eta_{2,i,j}^k]^\beta}{\sum_{b \in J} [\tau_{b,j}^k]^\alpha \cdot [\eta_{1,b,j}^k \cdot \eta_{2,b,j}^k]^\beta} \quad (10)$$

#### G. Suavizamiento y Reinicialización de Feromonas

Cuando MMAS converge o está cerca de hacerlo en [1] se propone un suavizamiento de las feromonas, el cual busca la exploración de otros caminos proporcionando más feromonas en los estados con bajo nivel de feromonas. Esto se hace según la siguiente fórmula:

$$\tau_{ij} = \tau_{ij} + fs \cdot (\tau_{max} - \tau_{ij}) \quad (11)$$

La variable  $fs$  es el factor de suavizamiento con  $0 \leq fs \leq 1$ . Con un  $fs < 1$  la información de

las feromonas no se pierde completamente, sino que disminuye un tanto. Con  $fs = 1$  se logra la reinicialización de las feromonas; mientras que con  $fs = 0$  se desactiva el mecanismo de suavizamiento.

El mecanismo de suavizamiento permite mayor exploración sin pérdida completa de la información de feromonas hasta ese instante, mientras que la reinicialización actúa como un evaporador completo de las feromonas depositadas hasta ese momento, con la diferencia que la hormiga ya tiene información de la solución global encontrada hasta el momento.

#### H. Pseudocódigo

El algoritmo ACO-DMFB se desarrolla en cuatro pasos, presentando varios métodos descritos en [13]. El pseudocódigo 1 siguiente presenta la propuesta. En el pseudocódigo 1 el método que construye la solución

---

#### Pseudocódigo 1 ACO-DMFB

---

**Entrada:** Matriz  $M(m \times n)$ , vector de celdas origen, vector de celdas destino, vector de celdas bloqueadas

- 1: inicializarParametros()
- 2: dividirColonia(cantidad\_gotas)
- 3: ordenarGotas()
- 4: **repetir**
- 5:   **desde**  $ant = 1$  **hasta** *hormigas* **hacer**
- 6:     solucion = Ant-DMFB()
- 7:     actualizarFeromonas(mejorIteracion) //con exp. 6
- 8:   **fin desde**
- 9:   actualizarFeromonas(MejorGlobal) //con exp. 6
- 10:   **si** suavizar **entonces**
- 11:     suavizar(feromonas) //con exp. III-G
- 12:   **fin si**
- 13:   **si** reinicializar **entonces**
- 14:     reinicializar(feromonas) //con exp. 11
- 15:   **fin si**
- 16: **hasta** condicionParadaAlgoritmo
- 17: **Salida:** mejor solución encontrada

---

(línea 6) es el que mueve cada gota en el tiempo  $t$ , y se muestra en el pseudocódigo 2.

La actualización de las feromonas se hace en las líneas 7 y 9 con la expresión (6), utilizándose la mejor solución de la iteración y la global respectivamente. El suavizamiento (línea 11) y la reinicialización (línea 14) de feromonas se realizan según una calendarización establecida en la inicialización del algoritmo.

La condición de parada del algoritmo (línea 16) está dada por la cantidad de generaciones o por el límite de tiempo de cómputo.

En el pseudocódigo 2 se presenta la hormiga artificial que construye una solución, la cual es denominada Ant-DMFB.

En la línea 3 la hormiga itera por cada sub colonia

## Pseudocódigo 2 Ant-DMFB

**Entrada:** sub colonia

```

1: repetir
2:    $t \leftarrow 0$ 
3:   desde  $k = 1$  hasta  $K$  hacer
4:     vecinos = obtener_vecinos()
5:     mientras siguienteVecino() hacer
6:       feromonas = getFeromonas(vecino,t)
7:        $\eta_1 = \text{getDistancia}()$  //con exp. 3
8:        $\eta_2 = \text{getPeso}()$  //con exp. 4 y 5
9:        $P = \text{probabilidad}()$  //con exp. 10
10:    fin mientras
11:    siguiente_estado = aleatorio( $P_1, \dots, P_{\text{vecinos}}$ )
12:    subSolucion(s) = subSolucion  $\cup$  {siguiente_estado}
13:  fin desde
14:   $t = t + 1$ 
15: hasta condicionParadaHormiga
16: solucion = {subSolucion(1), ..., subSolucion(K)}
17: Salida: solución de la hormiga

```

tomando los parámetros propios de cada una, convirtiéndose de esta manera en una hormiga propia de la sub colonia en la iteración actual. Durante el tiempo  $t$  cada sub colonia mueve su gota según el orden establecido por su valor de entropía.

La condición de parada (línea 17) de la hormiga está dada por el máximo tiempo de ruteo establecido a priori, o por la llegada de todas las gotas a sus celdas destino.

## IV. PRUEBAS EXPERIMENTALES

Las pruebas experimentales se han realizado en un computador con procesador Intel i3 de tercera generación, doble núcleo, para cuatro subprocesos y de 2.40 GHz; 6 GB de memoria RAM. Sistema operativo Windows 8.1 de 64 bits, lenguaje de programación Java (jdk7), paquete JGraph para construcción de grafos y algoritmo para hallar el camino más corto de Dijkstra.

El esquema de pruebas se realizó con 1000 iteraciones y una población de 200 hormigas cada una. Cada esquema se ha ejecutado 10 veces para cada prueba con un límite de tiempo computacional establecido en una hora. Como resultado de las pruebas se toma siempre la mejor solución considerando el orden lexicográfico.

No se ha encontrado en la literatura una medida que represente la capacidad de ruteo de las pruebas, por lo que en este trabajo se ha establecido el nivel de capacidad teniendo en cuenta las celdas libres, celdas bloqueadas y el número de gotas como dicha medida.

La capacidad está acotada en un intervalo de  $[1, 0]$ , donde 1 indica mayor capacidad de ruteo y 0 indica una menor capacidad de ruteo. La expresión (12) muestra cómo se realiza el cálculo de la capacidad de ruteo de una matriz.

$$comp = \frac{cel_{libres} - (gotas \times |8 \times disp|)}{cel_{totales}} \quad (12)$$

En este trabajo se clasifican como ensayos sintéticos a las instancias que no forman parte de un ensayo biológico completo, y tienen como objetivo verificar el comportamiento del ACO-DMFB en distintas configuraciones.

Para los ensayos biológicos reales se ha tomado una instancia particular del trabajo de [10]. El ensayo seleccionado es llamado ensayo in vitro para diagnósticos sobre fluidos fisiológicos humanos.

Los ensayos biológicos generan múltiples sub problemas que se toman individualmente para el enrutamiento.

Las instancias de prueba se han tomado del trabajo de Huang y HO en [4]. Sin embargo se han generado nuevas instancias realizando modificaciones de configuración en relación a la posición de las gotas, celdas libres y disponibilidad de espacio para maniobrabilidad de la gota para evitar conflictos.

### A. Pruebas con ensayos sintéticos

La Tabla I muestra los resultados en comparación con las soluciones del enfoque de fuerza bruta y ACO-DMFB. Vemos que las primeras tres pruebas tienen la misma capacidad pero diferentes resultados debido a la configuración de las gotas. En esas tres primeras pruebas se ha modificado solo el lugar de las celdas destino agrupando algunas de ellas de tal forma a que todas las gotas se dirijan hacia el mismo sector ocasionando posibles estados de conflicto.

TABLE I  
RESULTADOS EXPERIMENTALES

Benchmark				Fuerza Bruta			ACO-DMFB		
Nombre	Tamaño	Gotas	Capacidad	F1	F2	F3	F1	F2	F3
Test 1 [4]	13x13	5	0.5266	0	13	41	0	13	41
Test 2	13x13	5	0.5266	0	13	45	0	13	45
Test 3	13x13	5	0.5266	0	17	55	0	17	55
Test 4 [4]	13x13	6	0.4260	0	15	62	0	16	64
				Estimación			ACO-DMFB		
Test 5 [4]	16x16	9	0.3438	0	22	133	8	28	159
Test 6 [4]	24x24	24	0.4531	0	58	585	207	134	1385

### B. Pruebas con ensayo biológico in vitro

La configuración de este ensayo está basada en [10]. El protocolo del ensayo se modela por un grafo de secuencia descomponiéndose en sub problemas que deben resolverse individualmente para luego calendarizar y ejecutar concurrentemente. En la tabla II se muestran los resultados obtenidos con ACO-DMFB en cada sub problema. Mientras que en la tabla III se comparan estos resultados con un trabajo del estado del arte basado en ACO [11].

Se incluye en la tabla III el número de celdas (#Celdas) que se utilizaron durante todo el ensayo, que agrupa a todas las celdas activadas sin diferenciar el sub problema en el cual fueron activadas.

TABLE II  
RESULTADOS EXPERIMENTALES PARA EL BIOENSAYO IN VITRO EN  
CADA SUB PROBLEMA

Sub problema	ACO [11]			ACO-DMFB OPT2		
	$f_2$	$f_3$	$T_{min}$	$f_2$	$f_3$	$T_{min}$
1	18	59	0,0018	18	56	4,542
2	19	49	0,0089	18	48	1,222
3	18	65	0,0063	18	60	3,516
4	5	5	0,0012	5	5	0,21
5	17	44	0,0057	13	44	329,136
6	8	14	0,0015	8	14	0,4
7	15	25	0,0014	15	25	0,601
8	6	6	0,0012	6	6	0,224
9	8	8	0,0012	8	8	0,245
10	13	13	0,0015	13	13	0,301
11	10	10	0,0014	10	10	0,277

TABLE III  
RESULTADO DEL ENSAYO IN VITRO COMPARADO CON ACO [11].

ACO			ACO-DMFB		
Max. T	Avg. T	#Celdas	Max T	Avg. T	#Cedas
18	12.47	200	18	12	197

## V. DISCUSIÓN DE LOS RESULTADOS

En los casos donde la capacidad de ruteo es de complejidad media, las soluciones encontradas responden a las exigencias del tiempo de ruteo pero se debilitan un poco en la cantidad de celdas utilizadas. Las causas son que el algoritmo tiende siempre a encontrar la solución con el mejor tiempo por la utilización de la heurística del camino más corto, aplicado en una de las visibilidades; mientras que, las reglas aplicadas en la visibilidad  $\eta_2$  tratan de optimizar tanto la cantidad de celdas utilizadas como la cantidad de adyacencias, pero al suponer posibles movimientos de las demás gotas trata de evitar una adyacencia a costa de utilizar más celdas incrementando el uso de las mismas.

En los casos de capacidad de ruteo de complejidad alta, las hormigas presentan un problema con relación a las decisiones a largo plazo. En este trabajo la hormiga maneja información de hasta dos unidades de tiempo futuras, y además a veces la información es estimada ya que la hormiga solo supone el posible movimiento de las demás gotas; por tanto, es incapaz de tomar decisiones que podrían beneficiarla más adelante. La hormiga debería ser capaz de ver mucho más que dos unidades de tiempo y saber que una espera mayor evitaría varios conflictos.

## VI. CONCLUSIÓN Y TRABAJOS FUTUROS

En este trabajo se propone ACO-DMFB basado en MMAS [1] para tratar el problema DMFB. Se caracteriza por la utilización de una tabla de feromonas que guarda información de la transición en el tiempo. Además, se implementa la capacidad de selección de estados ya visitados en las hormigas.

En las pruebas realizadas, el algoritmo de este trabajo

halla la solución de la cota de tiempo de ruteo mínima para los casos de complejidad sencilla, donde las hormigas tienen mayor capacidad de ruteo. La propuesta del ordenamiento por entropía mejora el tiempo computacional y la calidad de la solución.

Con la experiencia obtenida, se ve que la complejidad del problema depende de muchos factores además de las celdas libres, por lo que se propone como trabajo futuro el estudio de propuestas ACO multiobjetivos en Pareto, además de otros tipos de visibilidades para el ACO-DMFB. También se sugiere el uso de alguna metodología de clústeres de gotas para minimizar el número de matrices de feromonas.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Thomas Stützle y Holger H. Hoos, "MAX-MIN Ant System," Future Generation Computer Systems, 16, pp. 889-914, 2000.
- [2] F. Su, K. Chakrabarty, and R. B. Fair, "Microfluidics -based biochips: Technology issues, implementation platforms, and design-automation challenges," IEEE Trans. on CAD, vol. 25, no. 2, pp. 211-223, Feb. 2006.
- [3] T. Xu and K. Chakrabarty, "Automated design of digital microfluidic lab-on-chip under pin-count constraints," Proc. ACM ISPD, pp. 90-98, Apr. 2008.
- [4] Tsung-Wei Huang, Tsung-Yi Ho "A Fast Routability- and Performance-Driven Droplet Routing Algorithm for Digital Microfluidic Biochips", TY Ho Computer Design, pp. 447-448, Oct. 2009.
- [5] T. Mukherjee, "Design automation issues for biofluidic microchips," Proc. IEEE/ACM ICCAD, pp. 463-470, Nov. 2005.
- [6] Karl F. Böhringer, "Modeling and Controlling Parallel Tasks in Droplet-Based Microfluidic Systems" Senior Member, IEEE, pp 329-336, Feb. 2006.
- [7] K. F. Böhringer, "Optimal strategies for moving droplets in digital microfluidic systems," presented at the 7th Int. Conf. Miniaturized Chemical and Biochemical Analysis Systems (MicroTAS), Squaw Valley, CA, 2003.
- [8] M. Cho and D. Z. Pan, "A High-Performance Droplet Routing Algorithm for Digital Microfluidic Biochips," IEEE Trans. on CAD, vol. 27, no. 10, pp. 1714-1724, Oct. 2008.
- [9] M. Dorigo y G. Di Caro. "The Ant Colony Optimization Meta-Heuristic". In D. Corne, M. Dorigo, and F. Glover (Eds.), New Ideas in Optimization, McGraw Hill, London, UK, 11-32. 1999.
- [10] Xu, T., and Chakrabarty, K. (2008). A droplet-manipulation method for achieving high-throughput in cross-referencing-based digital microfluidic biochips. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 27(11), 1905-1917.
- [11] Pan, I., Dasgupta, P., Rahaman, H., and Samantam, T. (2011, December). Ant colony optimization based droplet routing technique in digital microfluidic biochip. In Electronic System Design (ISED), 2011 International Symposium on (pp. 223-229). IEEE.
- [12] Juárez, J., Brizuela, C., Martínez, I., Velázquez, K., and Lozano, E. (2014). A genetic algorithm for the routing of droplets in DMFB: preliminary results. IEEE International Conference on Systems, Man, and Cybernetics 2014.
- [13] Mendoza, C., Szaran, E., (2014). Enrutamiento de microfluidos en biochips digitales. Un enfoque basado en colonias de hormigas multiobjetivas. Proyecto final de grado.