Elección en un sistema de agentes móviles[1]

Lali Barrière[2] Paola Flocchini[3] Pierre Fraigniaud[4] Nicola Santoro[5]

Resumen

En este artículo se estudia el problema de la elección de un líder en un conjunto de agentes dispersos en una red. Se considera una red anónima arbitraria de orden n y un conjunto de r agentes móviles también anónimos, que ejecutan su protocolo de forma asíncrona. Suponiendo n y r desconocidos se demuestran los resultados siguientes: si las etiquetas de los enlaces no son consistentes, es decir, no se dispone de *sense of direction*, no se puede resolver el problema; en una red con *sense of direction* se demuestra que hay solución si y solo si n y r son primos entre sí. La demostración de la posibilidad de la solución es constructiva y el protocolo presentado es nuevo y eficiente.

1 Introducción

Este trabajo trata sobre cuestiones computacionales que surgen en entornos de red con soporte de agentes móviles autónomos y asíncronos. En un nivel abstracto estos entornos, que llamaremos *sistemas móviles distribuidos*, pueden ser descritos como una colección $\mathcal{E}$ de entidades móviles autónomas dispersas en un grafo G. Estas entidades tienen las mismas capacidades de almacenaje y procesado, se comportan exactamente de la misma forma, es decir, ejecutan el mismo protocolo, y son capaces de moverse en G de un nodo a sus vecinos. En este contexto interesa determinar qué tareas pueden ser ejecutadas, bajo qué condiciones, y a qué coste. En particular, una cuestión central es determinar las hipótesis mínimas que permiten resolver un problema $\mathcal{P}$ dado. Ejemplos de este tipo de investigaciones son los estudios de reconstrucción de la topología: $\mathcal{E}$ es normalmente una única entidad, a veces dos, G es desconocido para esta entidad, y el problema $\mathcal{P}$ es la reconstrucción del mapa del grafo (ver por ejemplo [2, 6, 7]). Otros ejemplos son la exploración de un grafo [8, 9, 12], el problema del despertar o *wake-up* [1, 16], la búsqueda de un agujero negro [10, 11], la búsqueda de un intruso [4, 18], etc.

En este artículo, nos centramos en un problema fundamental en computación móvil distribuida: la *elección*, es decir el proceso a partir del cual un grupo de agentes móviles, inicialmente en el mismo estado y dispersos en una red G, son capaces de seleccionar uno de ellos como *líder*. Nos interesan soluciones *genéricas*, independientes de la topología, es decir, soluciones que funcionen independientemente de la estructura de G y del número de agentes. Suponemos, por lo tanto, que el tamaño del grafo, n, y el número de agentes, r, no son conocidos *a priori*. Al igual que en el entorno distribuido tradicional, la resolubilidad del problema de la elección en un sistema móvil distribuido está relacionado con las propiedades de las *funciones de etiquetado*, es decir, de la información de que se dispone para distinguir o identificar los nodos, los puertos (o enlaces incidentes en los nodos), y los agentes. Estas etiquetas se llaman, respectivamente "identificador de nodo", "número de puerto" y "nombre". Claramente, el problema de la elección se puede resolver si los nodos, los agentes o ambos tienen etiquetas distintas. En este caso, el problema interesante sería el de minimizar el coste de la solución óptima. En este trabajo nos interesamos por el problema de la elección de un líder en sistemas *totalmente anónimos*, es decir, los agentes y los nodos son, ambos, anónimos.

Una solución al problema es un protocolo determinista que permite, en un tiempo finito, elegir un líder en un sistema móvil distribuido, independi-

[1]Trabajo parcialmente subvencionado por la Junta de Andalucía.

[2]Dpto. de Matemática Aplicada 4. Universitat Politècnica de Catalunya. `lali@mat.upc.es`.

[3]School of Information Technology and Engineering, *University of Ottawa*, `flocchin@site.uottawa.ca`.

[4]CNRS, Laboratoire de Recherche en Informatique, *Univ. Paris-Sud*, `pierre@lri.fr`.

[5]School of Computer Science, *Carleton University*, `santoro@scs.carleton.ca`.

entemente de la topología de la red y de la posición inicial de los agentes. En primer lugar, se demuestra que si el sistema es totalmente anónimo y el etiquetado de los enlaces es arbitrario, el problema de la elección no es resoluble. En segundo lugar, consideramos un sistema totalmente anónimo en el cual se dispone de *sense of direction*, es decir, el etiquetado local de las aristas permite determinar si dos caminos que empiezan en un mismo nodo acaban o no en el mismo nodo [13]. En este caso se demuestra que si $\mathrm{Mcd}(r,n) \neq 1$, el problema es también irresoluble. Sin embargo, si $\mathrm{Mcd}(r,n) = 1$, se da un protocolo eficiente de elección. En otras palabras, cuando $\mathrm{Mcd}(r,n) = 1$, el disponer de *sense of direction* permite superar la anonimidad del sistema.

Debido a las limitaciones de espacio, no se dan aquí las demostraciones de los lemas.

2 Definiciones

Sean $G = (V,E)$ un grafo simple no dirigido y $E(u)$ el conjunto de aristas incidentes al vértice u. El etiquetado de G es una función inyectiva $\lambda_u : E(u) \to \Sigma$ que asocia a cada arista incidente a u una etiqueta o número de puerto. A cada arista (u,v) se asocian dos etiquetas, $\lambda_u(u,v)$ y $\lambda_v(u,v)$, que pueden ser distintas. El conjunto $\lambda = \{\lambda_u : u \in V\}$ es un *etiquetado* de G y denotamos por (G,λ) el correspondiente grafo etiquetado.

Si u y v son vértices de G, $P[u]$ es el conjunto de los caminos no vacíos de origen u y $P[u,v]$ el de caminos de u a v, se pueden considerar las extensiones de λ_u y λ de aristas a caminos, $\Lambda_u : P[u] \to \Sigma^+$ y $\Lambda = \{\Lambda_u : u \in V\}$. Dado un grafo etiquetado (G,λ), una *función de codificación* $\mathbf{c}$ para λ es cualquier función con dominio Σ^+ tal que dos caminos con el mismo vértice inicial se aplican al mismo valor si y solo si su vértice final es el mismo, es decir, $\forall u,v,w \in V$, $\forall \pi_1 \in P[u,v]$, $\pi_2 \in P[u,w]$, $\mathbf{c}(\Lambda_u(\pi_1)) = \mathbf{c}(\Lambda_u(\pi_2)) \Leftrightarrow v = w$. Una *función de decodificación* $\mathbf{d}$ para $\mathbf{c}$ es una función tal que dada una etiqueta $\lambda_u(u,v)$ de una arista (u,v) y la codificación de un camino π de v a otro vértice w, devuelve la codificación del camino de u a v obtenido al concatenar la arista (u,v) al camino π, es decir, $\forall (u,v) \in E$, $\forall \alpha = \Lambda_v(\pi)$, $\mathbf{d}(\lambda_u(u,v), \mathbf{c}(\alpha)) = \mathbf{c}(\lambda_u(u,v) \circ \alpha)$. Dada una función de codificación $\mathbf{c}$ de (G,λ) y una función de decodificación $\mathbf{d}$ para $\mathbf{c}$, el par $(\mathbf{c},\mathbf{d})$ se llama un *sense of direction* para (G,λ) [13].

Un *sistema totalmente anónimo* de agentes móviles se constituye de una red anónima, (G,λ), es decir, un grafo en que los vértices no están etiquetados, en el cual se halla disperso un conjunto de agentes móviles anónimos (y por lo tanto indistinguibles) y autónomos, $\mathcal{E}$. Si la localización inicial de los agentes en G viene dada por la función inyectiva $p : \mathcal{E} \to V(G)$, decimos que el vértice $p(a)$ es la *casa* del agente $a \in \mathcal{E}$. Los agentes tienen capacidades de cálculo y memoria limitadas, ejecutan el mismo protocolo y pueden moverse en G de un nodo a sus vecinos. Al desplazarse de u a v, un agente dispone de la etiqueta $\lambda_u(u,v)$ del enlace por el cual ha salido de u, así como de la etiqueta $\lambda_v(v,u)$ del enlace por el cual ha llegado a v.

Los agentes se comportan de manera *asíncrona*, en el sentido de que cada acción ejecutada puede tomar un tiempo impredecible, que se supone acotado. No se dispone de conocimiento *a priori* sobre la topología de G, ni tampoco de su tamaño, ni del número de agentes presentes en el sistema. Cada nodo está provisto de una *pizarra*, una memoria local donde los agentes pueden escribir, leer y borrar información. Los accesos a la pizarra se efectúan bajo el supuesto de exclusión mutua. Inicialmente, las casas de los agentes están marcadas, aunque debido al anonimato, este marcador denota únicamente que el nodo es la casa de algún agente y no puede ser utilizado para romper la simetría del sistema. Todas las pizarras contienen además las etiquetas de los puertos incidentes. El comportamiento de un agente es el siguiente: basándose en el valor de su variable de estado y en el contenido de la pizarra del nodo en que se encuentra actualmente, efectúa los cálculos que le permitirán decidir sobre su evolución posterior. Una vez completados estos cálculos, el agente *cambia su estado* y utiliza el puerto de salida determinado por dicho cálculo para *desplazarse*. (Se puede añadir un puerto *nulo* para describir una decisión en la cual el agente no se mueve.) Inicialmente, el valor de la variable de estado es *disponible* para todos los agentes. El problema de la *elección* consiste que los agentes seleccionen de manera unánime a uno de ellos, cuya variable de estado tomará el valor *líder*, mientras que las variables de estado de todos los demás tomarán el valor *seguidor*.

3 Imposibilidad de la elección con etiqueta arbitrarias

En esta sección se demuestra que, si el etiquetdo de los enlaces es arbitrario, es decir, sirve únicamente para distinguirlos, entonces el problema de la elección no se puede resolver.

Teorema 1 *En un sistema totalmente anónimo con etiquetado de enlaces arbitrario, el problema de la elección no es determinísticamente resoluble.*

Suponemos que P es un protocolo de elección correcto, y consideramos ejecuciones síncronas de P, en las que todos los agentes empiezan de forma simultánea. Es decir, los cálculos se efectuan de forma instantánea, los movimientos requieren una unidad de tiempo, y los agentes empiezan simultáneamente. Además, suponemos que cada nodo puede ser visitado por un único agente por unidad de tiempo.

Notamos por $s(a,t)$ el estado del agente a, por $p(a,t)$ la posición del agente a, por $w(v,t)$ el contenido de la pizarra del nodo v y por $e(a,t)$ la etiqueta del puerto de entrada por el cual a llegó a $p(a,t)$, todos ellos en el instante t. Así, el sistema evoluciona de acuerdo con las reglas siguientes.

1) El contenido de la pizarra puede cambiar únicamente cuando el nodo correspondiente es visitado por un agente. El nuevo contenido viene dado por una función f que sólo depende del contenido actual, el puerto de entrada y el estado del agente visitante, si lo hay. En otras palabras, si ningún agente está en v en el instante $t+1$, entonces $w(v,t+1) = w(v,t)$; en caso contrario, $w(v,t+1) = f(w(v,t), s(a,t), e(a,t))$, donde $p(a,t) = v$.

2) El enlace a utilizar por un agente viene dado por una función g que sólo depende del estado actual del agente, el puerto de entrada y el contenido de la pizarra (recordemos que ésta contiene las etiquetas de los enlaces incidentes). En otras palabras, si $p(a,t) = v$, entonces $\ell(v,t+1) = g(w(v,t), s(a,t), e(a,t))$.

3) El nuevo estado de un agente viene dado por una función γ que sólo depende del estado actual del agente, el puerto de entrada y el contenido de la pizarra. En otras palabras, si $p(a,t) = v$, entonces $s(a,t+1) = \gamma(w(v,t), s(a,t), e(a,t))$.

En todos los sistemas que consideramos a continuación, el grafo es un ciclo con la orientación consistente "izquierda/derecha". Esta orientación no implica *sense of direction*, ya que sin conocer n no se puede disponer de una función de codificación consistente [13].

El sistema $\mathcal{A}$ tiene tres nodos (y_0, y_1, y_2), y un único agente a situado inicialmente en y_0. Después de un número finito de pasos, la ejecución de cualquier protocolo correcto debe terminar con el agente *líder*. Llamamos $T(\mathcal{A})$ al tiempo de la ejecución síncrona de P en $\mathcal{A}$. El sistema $\mathcal{B}$ tiene $n = 3q$ nodos, $(x_0, x_1, \ldots, x_{n-1})$ donde $q \geq \frac{4}{3}(T(\mathcal{A})+1)$, y un agente en cada una de las posiciones x_{3j}, $0 \leq j \leq q-1$. (Se cumple $\mathrm{Mcd}(r,n) = q \neq 1$.) La simetría inicial del sistema en $\mathcal{B}$ es total: todos los agentes "ven" un sistema idéntico, que además no puede ser distinguido del sistema que percibe el agente único de $\mathcal{A}$. En un entorno distribuido tradicional, es decir, con elementos *estáticos*, esto sería suficiente para garantizar que la relación entre los dos sistemas es invariante en el tiempo, en el supuesto de sincronía (ver por ejemplo [19]). Todos los agentes de $\mathcal{B}$ serían elegidos como líder al mismo tiempo, contradiciendo la corrección del protocolo P. Veamos que, también en un entorno móvil, la correspondencia inicial entre los dos sistemas puede preservarse con el paso del tiempo. La ejecución síncrona de P en $\mathcal{B}$, donde todos los agentes empiezan simultáneamente tiene importantes propiedades. Notamos por a_j el agente con casa x_j y, por simplicidad en las notaciones, escribimos $x(t) = p(a_0,t)$ y $x(t)+h = p(a_h,t)$ las posiciones de a_0 y de a_h en el instante t, respectivamente.

Lema 2 *En cualquier instante de tiempo $t \geq 0$*
(1) por cada $i = 0,1,2$ los contenidos de las pizarras $w(x_{i+3j},t)$ son idénticos para todo j, $0 \leq j \leq q-1$,
(2) los estados $s(a_{3j},t)$, son idénticos para todo j, $0 \leq j \leq q$,
(3) los puertos de entrada $e(a_{3j},t)$, son idénticos para todo j, $0 \leq j \leq q$, y
(4) $p(a_{3j},t) = x(t)+3j$, para todo j, $1 \leq j \leq q-1$, donde las operaciones son módulo n.

Lema 3 *En cualquier instante t,*
(1) $w(x_{i+3j},t) = w(y_i,t)$ para todo i,j, $0 \leq j \leq q-1$ y $0 \leq i \leq 3$,
(2) $s(a_0,t) = s(a,t)$,
(3) $e(a_0,t) = e(a,t)$, y

(4) si $p(a_0, t) = u$ *y* $p(a, t) = v$, *entonces* $\ell(u, t+1) = \ell(v, t+1)$.

Para ver que si $\mathrm{Mcd}(r, n) = 1$ el problema es también irresoluble, podemos suponer $k = T(\mathcal{A})$ múltiplo de 3, $k = 3d$ (en otro caso, tomaríamos $k' = 3\lceil k/3 \rceil$ en lugar de k). El sistema $\mathcal{C}$ tiene ahora $n = 4k + 2$ nodos, $(z_0, z_1, \ldots, z_{n-1})$, con un agente en el nodo z_0 y en cada una de las posiciones z_{3j} y z_{-3j}, $1 \leq j \leq 2d$, con las operaciones sobre los índices efectuadas módulo n. Tenemos pues $r = 4d + 1$, y $\mathrm{Mcd}(r, n) = 1$. Denotamos por b_j el agente cuya casa es el nodo z_j. El siguiente lema nos de la comparación de los k primeros pasos de las ejecuciones de P en $\mathcal{C}$ y en $\mathcal{B}$

Lema 4 *En cualquier instante* $t \leq k$,
(1) para todo $i = 0, 1, 2$, $w(z_{\pm(i+3j)}, t) = w(x_i, t)$, $0 \leq j \leq d$, *y*
(2) $s(b_0, t) = s(b_3, t) = s(b_{-3}, t) = s(a_0, t)$.

Se deduce del Lema 3 que en el instante de tiempo $T(\mathcal{A})$, el agente a_0 se convierte en *líder* en $\mathcal{B}$. Por lo tanto, por el Lema 2, todos los agentes se convierten en *líder* en $\mathcal{B}$ en el instante $T(\mathcal{A})$, contradiciendo la corrección del protocolo P para el caso $\mathrm{Mcd}(r, n) \neq 1$. Además, el Lemma 4 implica que, en el instante $T(\mathcal{A})$, los agentes b_0, b_3 y b_{-3} se convierten en *líder* en $\mathcal{C}$, contradiciendo la corrección de P en el caso $\mathrm{Mcd}(r, n) = 1$. Esto demuestra la imosibilidad de la resolución.

4 Elección con *sense of direction*

En esta sección consideramos un sistema totalmente anónimo en el que los agentes disponen, sin embargo, de *sense of direction*. En primer lugar se muestra cómo emplear el *sense of direction* de forma que los agentes, aun siendo anónimos, puedan asociar una única *firma* a su escritura. En segundo lugar se demuestra que, incluso con *sense of direction*, el problema de la elección sigue sin poderse resolver en el caso $\mathrm{Mcd}(r, n) \neq 1$. Para terminar, presentamos un protocolo que elegirá uno de los agentes como *líder* siempre que $\mathrm{Mcd}(r, n) = 1$.

4.1 Nombres y firmas

Inicialmente cada agente elige su propio nombre basándose en las etiquetas de los enlaces incidentes a su casa. A cada movimiento que el agente realiza en el grafo, el nombre es modificado, de forma que es siempre relativo a la actual posición del agente. La dificultad principal es modificar los nombres de forma que en cada posición v, dos nombres sean distintos si y solo si se refieren a agentes distintos. Esto asegurará que los mensajes escritos en la pizarra de v por agentes distintos tienen firmas distintas. Otra dificultad relacionada con el problema es asegurar que un agente es capaz de reconocer como propio cualquier mensaje que ha escrito en visitas previas.

ESTRATEGIA *Dynamic Name Mutation*
(1) Para determinar su nombre inicial, un agente a con casa $p(a) = u$, escoge arbitrariamente un nodo vecino $v \in E(u)$ y determina la etiqueta $\lambda_v(v, u)$. Entonces su nombre es $Myname := \mathbf{c}(\lambda_u(u, v) \circ \lambda_v(v, u))$.
(2) Cuando un agente con nombre $Myname$ en el nodo u se mueve a un nodo vecino v, modifica su nombre de acuerdo con la regla siguiente: $Myname := \mathbf{d}(\lambda_v(v, u), Myname)$.

Lema 5 *El nombre de a en v es* $\mathbf{c}(\alpha)$ *donde* α *es la secuencia de etiquetas correspondiente a un camino cualquiera de* v *a* h_a*, la casa de a.*

Lema 6 *En cualquiera de los nodos de G, dos nombres son distintos si y solo si se refieren a agentes distintos.*

4.2 Imposibilidad si $\mathrm{Mcd}(r, n) \neq 1$

Al disponer de *sense of direction*, un agente que llega a una casa, puede determinar mediante *Dynamic Name Mutation* si es la propia casa o la de otro agente, incluso con n y r desconocidos. Por lo tanto, los sistemas $\mathcal{A}$ y $\mathcal{B}$ del apartado anterior pueden ser distinguidos por los agentes. A pesar de ello, si $\mathrm{Mcd}(r, n) \neq 1$, no podemos tampoco resolver el problema de la elección.

Teorema 7 *En un sistema totalmente anónimo con* sense of direction, *el problema de la elección no es determinísticamente resoluble si* $Mcd(r, n) \neq 1$.

La demostración se basa también en el ciclo $(x_0, x_1, \ldots, x_{n-1})$, con el *sense of direction* inducido por la orientación consistente "izquierda/derecha". Supongamos $\mathrm{Mcd}(r, n) = d \neq 1$, con una situación inicial simétrica, con los agentes separados por $d-1$ nodos vacíos. La simetría inicial del sistema es total, el sistema es homónimo, y todos los agentes ven el sistema de la misma forma. La ejecución síncrona mantiene la simetría a lo largo del tiempo: a cada instante, los agentes estarán exactamente en el miso estado, reaccionando de la misma forma y efectuando las mismas acciones, leyendo y escribiendo la misma información en las pizarras y seleccionando el mismo puerto de salida para el movimiento siguiente. Si un agente se convierte en líder, todos los demás lo harán también, simultáneamente.

4.3 Protocolo de elección para $\mathbf{Mcd}(r,n) = 1$

Presentamos un protocolo que, siempre que $\mathrm{Mcd}(r,n) = 1$, selecciona uno de los agentes como líder. Este protocolo se ejecuta en una secuencia de *fases electorales.* Cada fase se compone de dos operaciones, que son ejecutadas por el conjunto de agentes *activos*: (1) conquista de territorio, y (2) una secuencia de etapas de partición y emparejamiento. Al final de una fase, al menos la mitad de los agentes que la empezaron como *activos* la finalizarán como *pasivos*, de tal forma que, además, el número de agentes que sobrevivan de una fase a la siguiente sea siempre primo con n. A continuación se describen las acciones de los r_i agentes que empiezan la Fase i (inicialmente $r_1 = r$).

4.3.1 Conquista de territorio.

La primera operación efectuada por un agente *activo* es "conquistar" tantos nodos como le sea posible. Al comienzo de una fase se suponen todos los nodos *disponibles*, excepto las casas de los agentes activos. Un agente activo empieza entonces un recorrido del grafo, marcando como *conquistado* cualquier nodo disponible que visite. El marcado se efectua escribiendo una información adecuadamente *firmada* en la pizarra. Durante el recorrido, un agente guarda en memoria el número de nodos conquistados por otros agentes. Los nombres de los agentes en la lista se decodifican a cada movimiento, de forma que serán siempre consistentes.

OPERACION *Conquista*

(1) El agente a escribe en $p(a)$ el número de fase actual y empieza el recorrido de G.

(2) Durante el recorrido, a guarda la lista L de nombres de agentes, con sus contadores asociados. Al llegar al nodo u por el enlace (v,u):

- a modifica su nombre: $Myname := \mathbf{d}(\lambda_v(v,u), Myname)$, y todos los nombres de su lista L: $\forall m \in L$, $m := \mathbf{d}(\lambda_v(v,u), m)$;
- si u ya ha sido conquistado durante esta fase, a modifica L y su contador asociado;
- si u está todavía disponible, a marca u como *conquistado*, incrementa su propio contador y prosigue el recorrido.

(Si u es la casa de un agente todavía activo, b, pero el número de fase no ha sido actualizado, a espera la llegada de b antes de proseguir el recorrido del grafo.)

4.3.2 Etapas de partición y emparejamiento.

La segunda operación que un agente activo debe realizar en una fase es una secuencia de *etapas de partición y emparejamiento.* En cada Etapa j, los agentes *activos* se reparten en dos grupos, $W^{(j)}$ y $S^{(j)}$. Cada agente de $S^{(j)}$ intentará emparejarse con un agente de $W^{(j)}$. Dependiendo del resultado del emparejamiento, algunos agentes se convierten en *pasivos* al final de la etapa, mientras que otros siguen activos y se reparten en dos nuevos grupos, $W^{(j+1)}$ y $S^{(j+1)}$. Cuando uno de los dos grupos está vacío, se da por terminada la fase actual. En este último caso se empieza una nueva fase, excepto cuando sólo queda un agente *activo*, que pasa a ser el *líder* y termina el protocolo.

Primera etapa ($j = 1$): Partición

Al finalizar la conquista de territorio, cada agente activo conoce una secuencia de enteros, $n_1, \ldots, n_k$, y una partición $S_1, \ldots, S_k$ del conjunto de agentes, tales que para cada ℓ, S_ℓ es el conjunto de agentes que han conquistado exatamente n_ℓ nodos. Basándose en esta información, cada agente determina los siguientes dos conjuntos: A = conjunto de agentes con más de n/r_i nodos conquistados, y B = conjunto de agentes con menos de n/r_i nodos conquistados. Observemos que, puesto que $\mathrm{Mcd}(r_i,n) = 1$, n/r_i no es entero. Sea $W^{(1)}$ el mayor de estos dos conjuntos y $S^{(1)}$ el más pequeño (en caso de igualdad $W^{(1)} = A$). Los agentes en W se llaman *waiting*, mientras que los de S se llaman *searching.*

Primera etapa ($j = 1$): Emparejamiento
Agentes waiting.
(1) Un agente $a \in W^{(1)}$, inicialmente *soltero*, debe escribir el número de etapa actual en su casa, y esperar la visita de todos los agentes de $S^{(1)}$.
(2) Cuando un agente $s \in S^{(1)}$ llega a $p(a)$:

- a guarda el nombre local de s
- si s se empareja con a, a pasa a ser *casado*.

(3) Una agente a *casado* que ha sido visitado por todos los agentes de $S^{(1)}$, pasa a ser *pasivo*.
Agentes searching.
(1) Un agente $s \in S^{(1)}$ debe recorrer el grafo, y visitar a todos los agentes de $W^{(1)}$.
(2) Durante el recorrido, cuando s llega a $p(a)$, $a \in W^{(1)}$, s comprueba si a ha regresado ya a su casa.

- Si a no está en $p(a)$, s le espera.
- Cuando a llega a $p(a)$, si ambos están solteros, s y a se emparejan; entonces s prosigue el recorrido de G. (Si hay varios agentes en $p(a)$, sólo uno puede emparejarse con a.)

Agentes pasivos.
Un agente pasivo debe permanecer en su casa, esperando la notificación de la teminación del protocolo. Denotamos por $P^{(1)}$ el conjunto de agentes que pasan a ser *pasivos* durante la Etapa 1.
Etapas siguientes ($j \geq 2$)

Los conjuntos $S^{(j)}$ y $W^{(j)}$ se definen de acuerdo con la regla siguiente: (1) si $|W^{(j-1)}| - |S^{(j-1)}| \geq |S^{(j-1)}|$ entonces $S^{(j)} = S^{(j-1)}$ y $W^{(j)} = W^{(j-1)} \setminus P^{(j-1)}$; y (2) si $|W^{(j-1)}| - |S^{(j-1)}| < |S^{(j-1)}|$ entonces $W^{(j)} = S^{(j-1)}$ y $S^{(j)} = W^{(j-1)} \setminus P^{(j-1)}$. ($P^{(j-1)}$ es el conjunto de agentes que pasan a ser *pasivos* durante la Etapa $j-1$.)

Cuando todos los agentes de $S^{(j)}$ han finalizado el recorrido, un agente activo a es capaz de calcular $|S^{(j)}|$ y $|W^{(j)}|$, y también de determinar a cual de los dos conjuntos pertence.
(1) Si $|S^{(j)}| \neq 0$, a empieza la Etapa $j+1$.
(2) Si $|S^{(j)}| = 0$, entonces $r_{i+1} = |W^{(j)}|$ es el número de agentes todavía activos. Si $r_{i+1} = 1$, a se convierte en *líder* y empieza la fase final del protocolo; si $r_{i+1} > 1$, a empieza la Fase $i+1$.

El emparejamiento en la Etapa j se realiza de forma análoga al emparejamiento en las etapas anteriores, con las puntualizaciones siguientes.
Caso 1. $S^{(j)} = S^{(j-1)}$ y $W^{(j)} = W^{(j-1)} \setminus P^{(j-1)}$. En este caso, los actuales *waiting* eran ya *waiting* en la etapa anterior. Por lo tanto, se encuentran en su casa, a la espera de ser visitados por los agentes *searching*. Si un agente *searching* llega a una casa de un agente *waiting* que se encuentra todavía en la etapa anterior, debe esperar a que éste actualice su contador de etapa o se convierta en *pasivo*.
Caso 2. $S^{(j)} = W^{(j-1)} \setminus P^{(j-1)}$ y $W^{(j)} = S^{(j-1)}$. Los actuales agentes *searching* eran agentes *waiting* en la etapa anterior. Por lo tanto, han sido visitados por todos los agentes que actualmente son *waiting*. Por lo tanto, todos los agentes de $S^{(j)}$ conocen los nombres (locales) de los agentes de $W^{(j)}$. Esto es útil en caso de la llegada de un agente *searching* a la casa de un agent *waiting* que no ha finalizado todavía la etapa anterior.

4.4 Corrección

Mediante la siguiente serie de lemas se demuestra que el protocolo propuesto permite elegir un líder entre r agentes anónimos dispersos en una red anónima de n nodos con *sense of direction*, siempre que r y n sean primos entre sí.

Lema 8 *Si un agente activo completa la Conquista de territorio de la Fase i, $i \geq 1$, entonces cualquier otro agente activo ha empezado la Fase i.*

Lema 9 *Si el número de agentes activos al comienzo de la Fase i no divide a n, entonces $S^{(1)}$ y $W^{(1)}$ forman una partición del conjunto de agentes activos en dos partes no vacías.*

Lema 10 *Suponemos que al comienzo de la Fase $i \geq 1$, r_i no divide a n. Entonces, al final de la Etapa $j-1$, $j \geq 1$, de la Fase i, se cumple: (1) cada agente activo conoce los cardinales de $S^{(j)}$ y $W^{(j)}$, (2) cada agente activo conoce a cual de estos dos conjuntos pertenece, y (3) cada agente en $S^{(j)}$ es capaz de reconocer a los agentes de $W^{(j)}$.*

Lema 11 *Suponemos que al comienzo de la Fase $i \geq 1$, r_i no divide a n. Suponemos también que un agente $a \in S^{(j)}$, actualmente en la Etapa $j \geq 1$ de la Fase i, llega a $p(b)$, $b \in W^{(j)}$. Si a y b no están en la misma etapa, entonces b está en la Etapa $j-1$ de la Fase i.*

Lema 12 *Para todo $i \geq 1$, si r_i no divide a n entonces en alguna Etapa $k \geq 1$ se satisface la condición $S^{(k+1)} = \varnothing$.*

Lema 13 *Suponemos que al comienzo de la Fase $i \geq 1$, r_i no divide a n. Entonces todo agente activo en esta fase completará en algún momento, la etapa actual.*

Lema 14 *Si $Mcd(r,n) = 1$, entonces para todo i, $Mcd(n,r_i) = 1$ y el protocolo propuesto termina después de como mucho $\log_2 r$ fases.*

Para finalizar, podemos pues afirmar la validez del teorema siguiente.

Teorema 15 *En un sistema totalmente anónimo con* sense of direction, *hay una solución determinista para el problema de la elección si y solo si $Mcd(n,r) = 1$.*

En este trabajo nos hemos preocupado de encontrar las hipótesis mínimas que permiten resolver el problema de le elección en un sistema móvil distribuido. El coste de esta solución no ha sido para nosotros una cuestión relevante. Sin embargo, el teorema siguiente muestra que el protocolo encontrado es eficiente, en el sentido de que el coste en número total de movimientos de agentes es mínimo.

Teorema 16 *El número total de movimientos de agentes en el protocolo propuesto es $O(rn)$ en el peor de los casos.*

Referencias

[1] E. Arkin, M. Bender, S. Fekete, and J. Mitchell. The freeze-tag problem: how to wake up a swarm of robots. In 13th ACM-SIAM Symposium on Discrete Algorithms (SODA '02), pages 568–577, 2002.

[2] B. Awerbuch, M. Betke, and M. Singh. Piecemeal graph learning by a mobile robot. Information and Computation, 152:155–172, 1999.

[3] L. Barrière and S. Dobrev. Leader election in abelian Cayley graphs. In 8th Colloquium on Structural Information and Communication Complexity (SIROCCO '01), Carleton Scientific, pages 5–20, 2001.

[4] L. Barrière, P. Flocchini, P. Fraigniaud, and N. Santoro. Capture of an Intruder by Mobile Agents. In *Proceedings of SPAA02*, 2002, to appear.

[5] L. Barrière, P. Flocchini, P. Fraigniaud, and N. Santoro. Distributed Mobile Computing with Incomparable Labels. Technical Report LRI-1309, Université Paris-Sud, France, 2002.

[6] M. Bender, A. Fernandez, D. Ron, A. Sahai, and S. Vadhan. The power of a pebble: Exploring and mapping directed graphs. In 30th ACM Symp. on Theory of Computing (STOC '98), pages 269–278, 1998.

[7] X. Deng, T. Kameda and C. H. Papadimitriou. How to learn an unknown environment I: the rectilinear case. Journal of the ACM, 45:215-245, 1998.

[8] X. Deng and C. H. Papadimitriou. Exploring an unknown graph. Journal of Graph Theory, 32:265-297, 1999.

[9] K. Diks, P. Fraigniaud, E. Kranakis and A. Pelc. Tree exploration with little memory. In 13th ACM-SIAM Symposium on Discrete Algorithms (SODA '02), pages 588–597, 2002.

[10] S. Dobrev, P. Flocchini, G. Prencipe and N. Santoro. Mobile agents searching for a black hole in an anonymous ring. In 15th Int. Symposium on Distributed Computing (DISC 2001), pages 166–179, 2001.

[11] S. Dobrev, P. Flocchini, G. Prencipe and N. Santoro. Searching for a black hole in arbitrary networks. In 21st ACM Symposium on Principles of Distributed Computing (PODC 2002), to appear.

[12] C. Duncan, S. Kobourov and V. Kumar. Optimal constrained graph exploration. In 12th ACM-SIAM Symp. on Discrete Algorithms (SODA '01) pages 807-814, 2001.

[13] P. Flocchini, B. Mans and N. Santoro. Sense of direction: definition, properties and classes, Networks 32: 165-180, 1998.

[14] P. Flocchini, B. Mans and N. Santoro. Sense of direction in distributed computing, In 12th International Symposium on Distributed Computing (DISC 98), pages 1-15, 1998. To appear in Theoretical Computer Science.

[15] P. Fraigniaud, C. Gavoille and B. Mans. Interval routing schemes allow broadcasting with linear message-complexity. In 19th ACM Symposium on Principles of Distributed Computing (PODC '00), pages 11-20, 2000.

[16] P. Fraigniaud, A. Pelc, D. Peleg and S. Pérennes. Assigning labels in unknown anonymous networks In 19th ACM Symposium on Principles of Distributed Computing (PODC '00), pages 101-112, 2000.

[17] N. Lynch. Distributed Algorithms. Morgan Kaufmann, Inc. San Francisco, California, 1996.

[18] I. Suzuki and M. Yamashita. Searching for a mobile intruder in a polygonal region. SIAM Journal on Computing, 21(5):863–888, 1992.

[19] M. Yamashita and T. Kameda. Computing on anonymous networks, part I: characterizing the solvable cases. IEEE Transaction on Parallel and Distributed Computing, 7(1):69–89, 1996.