

La mitosis celular: una sorprendente frontera de la eficiencia

Mario de J. Pérez Jiménez

Resumen. En este trabajo se presenta una frontera entre la eficiencia e ineficiencia de problemas abstractos en un marco computacional inspirado en la estructura y funcionamiento de las células de los organismos vivos. Más concretamente, se justifica que una regla de reescritura, abstracción de la mitosis celular, proporciona un límite de la tratabilidad de problemas, bajo el supuesto de que se verifique la conjetura $P \neq NP$. Como consecuencia relevante se obtienen nuevas herramientas que permiten atacar la resolución de la citada conjetura en un marco absolutamente novedoso.

14.1. Introducción

La mejora de la calidad de vida del hombre ha sido una constante aspiración de la sociedad. Es claro que ese deseo de mejora suele ser ilimitado; es decir, no existe propiamente una *meta* tras la cual hay un paraíso terrenal. A lo largo del camino de la Vida van surgiendo muchas dificultades cuya superación representa un pequeño logro colectivo. Para ello, es necesario: (a) abordar el asunto de manera sistemática, esto es, detectar con claridad cuál es el escollo que se trata de salvar; (b) formular de manera precisa la problemática que se ha presentado; y (c) tratar de buscar una solución *satisfactoria*.

Con frecuencia sucede que a la hora de materializar una solución del problema es necesario realizar una enorme cantidad de operaciones simples o procesos. Esta tarea suele ser extraordinariamente tediosa e, incluso, puede requerir una importante cantidad de tiempo si es el hombre el encargado de materializarlas. Por ello, ha sido un anhelo constante de la humanidad la búsqueda de herramientas mecánicas capaces de asistir al hombre en la ejecución de esas tareas elementales pero tediosas. Es la perseverante lucha del hombre por diseñar instrumentos de tipo mecánico que pudieran ayudarle en la resolución de los problemas que le van surgiendo. Hacia el año 1000 antes de Cristo aparece el ábaco (un instrumento de cálculo que permite realizar operaciones aritméticas simples), las varillas o huesos de Neiper (siglo XVII), la regla de cálculo de W. Oughtred (1621), la máquina de B. Pascal (diseñada entre 1642 y 1644), la máquina de Leibniz (1673), las tarjetas de Jacquard (1801), la máquina de diferencias (1822) y la máquina

analítica (1834) de Ch. Babbage (verdadero precursor de la disciplina que hoy se conoce por el término *Informática*), y la aparición de las primeras máquinas de propósito general hacia mediados de la década de los cuarenta del siglo XX (Colussus, 1943).

Por tanto, a la hora de resolver un problema de la vida real el hombre trata de hallar soluciones a través de un método especial que sea susceptible de ser *adaptado* convenientemente a fin de que pueda ser debidamente *interpretado* por una máquina de propósito general, capaz de reconocer y ejecutar una solución mecánica descrita en un lenguaje específico.

En este punto nos podríamos hacer una pregunta interesante ¿cualquier problema puede ser resoluble mediante un método del tipo antes descrito y que denominaremos *método* o *procedimiento mecánico*? (un procedimiento mecánico que resuelve un problema se denomina una *solución mecánica*). Curiosamente, esta pregunta no ha sido respondida de manera satisfactoria (y *negativa*) hasta 1936, primero por A. Church y, en el mismo año y de manera independiente, por A. Turing. Concretamente, probaron que un cierto problema relacionado con la *Lógica de Primer Orden* no podría ser resuelto mediante *ningún* procedimiento mecánico. Para ello, fue necesario definir rigurosamente ese concepto con el fin de demostrar que *ningún procedimiento mecánico*, de acuerdo con esa definición formal, resolvía dicho problema. De esta forma, entre 1931 y 1936, aparecieron los primeros *modelos formales de computación*: el modelo de las *funciones recursivas* (K. Göel, 1931), el modelo de las *funciones λ -calculables* (A. Church y S. Kleene, 1934) y el modelo de las *máquinas de Turing* (A. Turing, 1936). Los problemas que eran resolubles por procedimientos mecánicos de esos modelos se denominaron *problemas decidibles*.

A lo largo de este artículo, el término problema se entiende como sinónimo de *problema abstracto* que es un conjunto (usualmente, infinito numerable) de problemas concretos, denominadas *instancias* o datos de entrada del problema abstracto. Por ejemplo, un problema abstracto sería: *para cada número natural n , determinar un número impar que sea primo y estrictamente mayor que n* ; una instancia de ese problema abstracto sería la siguiente: *hallar un número impar que sea primo y estrictamente mayor que 8796540213*. Es claro que las distintas instancias de un problema abstracto tienen asociadas, de manera natural e intuitiva, un concepto de *tamaño*. Por ejemplo, en el problema antes citado el conjunto de las instancias del problema coincide con el conjunto de los números naturales y es fácil asignar a cada número natural un concepto *razonable* de tamaño (el número de dígitos significativos necesarios para representar un número natural en base 2).

El problema abstracto $X \equiv$ “*para cada número natural n , determinar un número impar que sea primo y estrictamente mayor que n* ”, es decidible ya que un procedimiento que resuelve el problema sería el siguiente:

- Entrada: un número natural n
 - Si $n \geq 2$ entonces devolver 3.
 - Si no
 - Considero una variable p y le asigno el valor $n + 1$.
 - Chequeo si p es primo e impar.
 - ◇ En caso afirmativo, devolver p y finalizar.
 - ◇ En caso negativo, aumentar en 1 el valor de la variable p y reiterar el proceso.

Es bien conocido que existen infinitos números primos. Por tanto, el procedimiento anterior siempre para; es decir, para cualquier número natural que se introduzca como entrada, el procedimiento se detendrá al cabo de un *número finito de pasos*.

Consideremos ahora el siguiente problema: *para cada número natural n , determinar un número impar p tal que p y $p + 2$ sean primos y estrictamente mayores que n* . En este caso, el panorama cambia sustancialmente ya que se desconoce si es infinito el conjunto de los números impares consecutivos que sean primos. Por tanto, si se considera un procedimiento mecánico parecido al antes descrito pero en el que la *instrucción* “Chequeo si p es primo e impar” se sustituye por esta otra “Chequeo si p y $p + 2$ son primos”, entonces no se puede asegurar que ese procedimiento se detenga cuando se introduce un número natural arbitrario y , por tanto, no proporcionaría una solución del problema propuesto. Así pues, no podemos asegurar que ese problema sea decidible ¿será indecidible? Este tipo de respuesta suele ser muy complicada y requiere un desarrollo teórico importante que está fuera del objetivo de este artículo.

Conviene observar que dado un problema decidible no sólo existe un procedimiento mecánico que lo resuelve sino que, en realidad, existen *infinitos* procedimientos mecánicos distintos que lo resuelven. Entonces, surge de manera natural una cuestión ¿cuál de los infinitos procedimientos mecánicos que resuelven un problema decidible es *el mejor*? Si bien esta cuestión suele ser extremadamente compleja de resolver, sí podemos aproximarnos un poco a la casuística que suele darse cuando se trabaja con problemas decidibles.

Analicemos el problema X descrito anteriormente, así como la solución mecánica propuesta. Tratemos de estudiar si es posible determinar, aunque sea de manera aproximada, el número de instrucciones que tendría que ejecutar ese procedimiento mecánico cuando se introduce como entrada un número natural n . Teniendo presente un resultado debido a Euclides (*para cualquier número natural $n \geq 1$, existe un número primo que es estrictamente mayor que n y menor o igual que $1 + n!$*) se deduce que los valores de p que tendríamos que chequear serían, en el peor de los casos, $p, p + 1, \dots, 1 + p!$; es decir, *grosso modo* el número de instrucciones a ejecutar sería, a lo sumo, del orden $p!$ ¿Qué significa este dato? Sencillamente que si considero un número natural con 50 cifras significativas en base 2, suponiendo que cada instrucción se ejecutara en un microsegundo, el procedimiento podría tardar miles de años en dar una respuesta. Por tanto, si bien el problema es decidible, el procedimiento mecánico antes diseñado no es *eficiente* en el sentido de que para instancias del problema de tamaño mediano podría tardar un tiempo desorbitado. En tales circunstancias se trataría de encontrar otros procedimientos mecánicos más eficientes ¿Y si no se encuentran?

El concepto de *eficiencia* de un procedimiento mecánico puede ser formalizado de la siguiente manera: los recursos computacionales (tiempo y/o espacio) que necesita un procedimiento mecánico para ser ejecutado, deben ser de tipo polinomial en el tamaño del dato de entrada. Esta definición es adecuada debido a que las funciones polinómicas suelen crecer de manera moderada, a diferencia de las funciones exponenciales en las que, a partir de un determinado valor relativamente pequeño se produce una especie de *cambio de fase*, creciendo de manera desmesurada. Un problema se dice que es *tratable* si existe un procedimiento mecánico que lo resuelve en tiempo polinomial. La clase de los problemas tratables se representa por \mathbf{P} .

Veamos un ejemplo ilustrativo de este concepto. Sea X_1 un problema tratable y A_1 un procedimiento mecánico que lo resuelve en tiempo cuadrático (es decir, para cada

instancia de tamaño n el procedimiento tarda, salvo un factor constante, n^2 unidades de tiempo en dar una respuesta). Sea X_2 un problema intratable y sea A_2 un procedimiento mecánico que lo resuelve en tiempo exponencial (por ejemplo, para una instancia de tamaño n el procedimiento tarda, salvo un factor constante, 2^n unidades de tiempo en dar una respuesta). Si ejecutamos los procedimientos A_1 y A_2 sobre una máquina que es capaz de realizar un millón de operaciones por segundo, entonces para un dato de entrada de tamaño 20, el procedimiento A_1 tardaría cuatro diezmilésimas de segundo, mientras que el algoritmo A_2 tardaría aproximadamente un segundo. En cambio, para un dato de entrada de tamaño 100, A_1 tardaría una centésima de segundo en dar la respuesta, mientras que A_2 tardaría, aproximadamente, trescientos noventa mil millones de siglos (para hacernos una idea de la cifra tan enorme que representa ese número: se estima que el Big Bang sucedió entre hace doce y quince millones de siglos).

Se conocen problemas que son tratables y, también, problemas que son intratables. No obstante, existe una cantidad importante de problemas de los que no se sabe si son tratables o intratables; es decir, problemas tales que todos los procedimientos mecánicos *conocidos* que los resuelven son de tipo exponencial pero no se ha conseguido probar que *ninguna* solución mecánica de dichos problemas tenga coste en tiempo polinomial. Entre los problemas presuntamente intratables destacan una clase importante conocida como problemas **NP**-completos.

Una de las cuestiones más importantes que tiene planteada la Ciencia hoy día es si la clase **P** y la clase **NP** son iguales, lo que puede expresarse equivalentemente de la siguiente manera: “¿*existe algún problema NP-completo que sea tratable?*”. Una respuesta afirmativa a esta cuestión tendría unas consecuencias funestas, en lo que respecta a la seguridad de la transmisión de información (base del comercio electrónico) que, en la actualidad, se asienta en los sistemas de encriptación. Sería posible disponer de un procedimiento mecánico que en tiempo razonable descifraría cualquier tipo de mensajes. No obstante, también tendría sus consecuencias positivas. Por ejemplo, sería posible diseñar programas que permitieran a un ordenador electrónico convencional encontrar demostraciones de teoremas que tengan pruebas de *longitud razonable*. Eso sí, sucedería que muchas de las pruebas no serían entendidas por los humanos. Entre otras consecuencias *positivas*, se conseguirían los siete millones de dólares que garantizan los premios del Instituto Clay de Massachussets (con tal de guardar cuidadosamente la prueba de que **P** = **NP** durante el tiempo necesario para obtener las soluciones de los restantes seis problemas). Lo que no estaría nada mal.

La relevancia de esa cuestión hace que desde 1971 toda una pléyade de eminentes científicos estén tratando de encontrar alguna respuesta. Para ello, se está trabajando en la obtención de herramientas que permitan atacar dicha cuestión desde diferentes puntos de vista. Ahora bien, es creencia generalizada que ha de verificarse que **P** \neq **NP** y, por tanto, una forma de atacar la cuestión consistiría en buscar fronteras (lo más *fin*a posible) de la tratabilidad de problemas, bajo ese supuesto. Imaginemos que en un determinado marco computacional si se admite una cierta propiedad θ entonces se puede resolver de manera eficiente un problema X presuntamente intratable y que, en caso contrario, sólo se pueden resolver problemas tratables. En tal situación tendríamos lo siguiente:

- Para probar que **P** \neq **NP** bastaría demostrar que el problema X no se puede resolver eficientemente en dicho marco computacional si se prescinde de la propiedad θ .
- Para probar que **P** = **NP** bastaría demostrar que el problema X se puede resolver

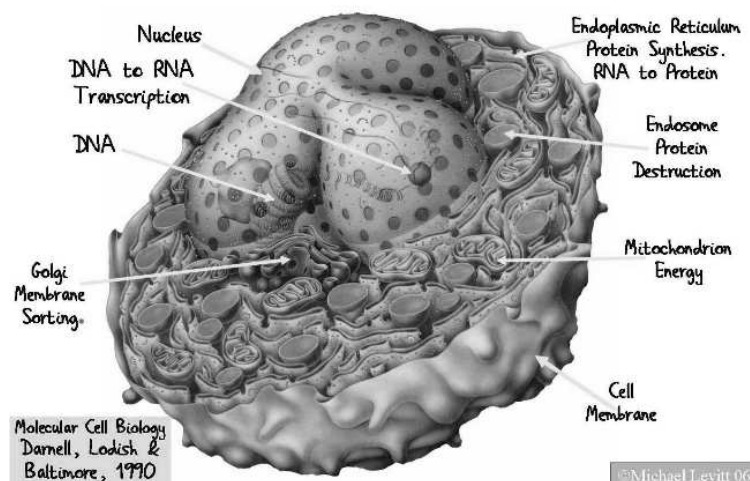
eficientemente en dicho marco computacional, incluso en el caso que se prescindiera de la propiedad θ .

A este respecto, la aparición de los modelos de computación bio–inspirados y no convencionales proporcionan un nuevo marco para abordar esta importante cuestión.

14.2. Las células de los organismos vivos

La célula es la unidad fundamental de todo organismo vivo. Posee una estructura compleja y, a la vez, muy organizada que permite la ejecución simultánea de un gran número de reacciones químicas. Existen, básicamente, dos tipos de células: las *procariotas*, que carecen de un núcleo bien definido y son propias de los organismos unicelulares (bacterias), y las *eucariotas*, que poseen un núcleo rodeado por una doble membrana y son específicas de los animales y plantas. Ambas realizan de manera similar una serie de procesos complejos que son esenciales para la vida: la replicación del ADN, la producción de energía, la síntesis de proteínas y los procesos metabólicos.

Existen tres partes fundamentales en una célula: (a) una especie de piel (*membrana plásmica*) que delimita a la célula de su entorno, permitiendo diferenciar el interior del exterior de la célula; (b) el corazón de la célula (*núcleo*), que almacena la información genética a través de moléculas de ADN y de ARN; y (c) el resto de la célula (*citoplasma*). A su vez, el citoplasma consta de: *la mitocondria*, encargada de la producción de energía, el *aparato de Golgi*, que viene a ser una fábrica de proteínas y juega un papel esencial en el metabolismo de la célula, el *retículo endoplásmico*, que es una red de membranas interconectadas que facilita el transporte de moléculas y la comunicación entre las distintas componentes, y los *lisosomas*, que constituyen el *estómago* de la célula.



Las *membranas biológicas* son estructuras dinámicas que juegan un papel crucial en las células vivas. Se estima que el noventa por ciento de las actividades celulares se realizan a través de ellas, y, por tanto, es un concepto esencial a la hora de definir el fenómeno que usualmente denominamos como *vida*. El premio Nobel de química del año 2003, se otorgó a los científicos estadounidenses Peter Agre y Roderick MacKinnon por sus descubrimientos relacionados con los canales de las membranas celulares.

Las membranas no sólo sirven para delimitar el interior del exterior de un determinado cuerpo, sino que también juegan un papel fundamental desde el punto de vista bioquímico, en el sentido que describimos a continuación. Una membrana (denominada *plasma*) separa y protege el espacio interno de la célula de su entorno. Las restantes membranas (que denominaremos *internas*) proporcionan la estructura propia de la célula ofreciendo la adecuada protección del *núcleo*, que es el depositario de la información genética de la célula. Se comportan como barreras semipermeables que o bien permiten que algunas sustancias químicas atraviesen la membrana en cualquiera de los dos sentidos (dentro–fuera o fuera–dentro), o bien, bloquea el paso de sustancias a través de ella. Es decir, las membranas se comportan como unos canales selectivos de comunicación para la transferencia de compuestos químicos entre distintos compartimentos de una célula, así como entre la propia célula y su entorno, controlando de esa manera un flujo de datos (es decir, de información).

A lo largo de su vida, la célula desarrolla básicamente cuatro procesos vitales desde su nacimiento que globalmente se denomina *ciclo celular*: crecimiento (fase G1), replicación del ADN (fase S), nuevo crecimiento (fase G2) para, finalmente, desencadenar la mitosis celular (fase M) que permite la producción de dos nuevas células que podríamos denominar hijas de la anterior.

14.3. Computación Celular con Membranas

Durante la última década, el campo de investigación denominado *Computación Natural* ha experimentado un enorme desarrollo. Se trata de una disciplina cuyo objetivo es el análisis, simulación e implementación de procesos dinámicos que se dan en la naturaleza y que son susceptibles de ser interpretados como procedimientos de cálculo. Esta disciplina surge de la investigación de modelos matemáticos y de las herramientas tecnológicas necesarias para la implementación de la computación bio–inspirada. Se trata de un nuevo paradigma computacional que proporciona modelos formales susceptibles de ser implementados en soporte físico no electrónico (de ahí que los modelos computacionales se denominen *no convencionales*).

La *Computación Celular con Membranas* (*Membrane Computing*) es un área de la Computación Natural, introducida por Gh. Păun [3] en 1998, que está inspirada en la estructura y el funcionamiento de las células de los organismos vivos, en cuanto a su capacidad para procesar y generar información. En octubre de 2003, el Institute for Scientific Information (ISI, USA), ha designado a la *computación celular con membranas* como *Fast Emerging Research Front* en el área de *Computer Science*.

Por otra parte, la *Bioinformática* es una disciplina que estudia la aplicación de herramientas y técnicas de la información para la manipulación de datos biológicos. Entre sus objetivos destacamos el desarrollo de una base de datos específica que permita almacenar y actualizar la gran cantidad de datos que han sido y están siendo generados de manera continua e ininterrumpida; el diseño de algoritmos eficientes para la comparación de sucesiones de cadenas; la investigación de métodos de predicción de la estructura tridimensional de las moléculas orgánicas, en general, y de las proteínas, en particular; y, finalmente, la definición de modelos físicos y/o matemáticos de sistemas biológicos.

Teniendo presente que a lo largo de la breve historia de la informática, muchos desarrollos tanto en el plano teórico como en el de aplicaciones prácticas, se han producido

bajo la inspiración de los procesos que se dan en la naturaleza, algunas cuestiones que se podría plantear un buen científico serían las siguientes: ¿Qué puede proporcionarnos la célula como fuente de inspiración para la informática? ¿Podemos abstraer una especie de modelo de computación a partir de la estructura y del funcionamiento de las células vivas? ¿Es la célula un *computador de propósito general*? ¿Es posible implementar computaciones a través de las células vivas, con la esperanza de disponer algún día de un *computador celular*?

Cada compartimento de la célula puede considerarse como una unidad de computación (como un *procesador*) con sus propios datos (compuestos químicos) y su programa local (reacciones químicas), de tal manera que el conjunto de compartimentos, vistos como una unidad global (la célula), puede ser considerado como un *modelo de computación no convencional*.

Conviene observar que en las ideas originales del paradigma computacional (*Membrane Computing*) introducido por Gh. Păun, el objetivo inicial no era modelizar la estructura y funcionamiento de una célula, sino más bien en estudiar y analizar algunos hechos relevantes desde el punto de vista computacional que pueden ser abstraídos de las células. Se consideran algunos aspectos de su funcionamiento y de su estructura, así como ciertas formas de comunicación de sustancias bioquímicas a través de sus membranas. De esta forma, se definirán variantes de sistemas P como dispositivos computacionales, tan realistas como sea posible desde un punto de vista biológico. Una posible implementación biológica de este modelo es, quizás, uno de los retos más importantes que tiene planteado este nuevo paradigma computacional.

14.4. Una descripción informal de los sistemas P

Esta interpretación de las células vivas tiene su formalización matemática en los *sistemas celulares con membranas* o *sistemas P*. En esta sección vamos a dar una descripción informal de los sistemas celulares básicos de transición (una formalización de los sistemas P puede encontrarse en [4]).

Una *estructura de membranas* consta de un conjunto finito y jerarquizado de membranas que se encuentran en el interior de una membrana principal (que se denomina *piel*) y delimita una serie de *regiones*. Cada región está formada por el espacio intercelular comprendido entre una membrana y las colocadas inmediatamente dentro de ella (si las hay). Una membrana se dice que es *elemental* si no tiene ninguna membrana dentro de ella. Más concretamente, una estructura de membranas puede ser descrita gráficamente como un diagrama de Venn (mediante una familia de subconjuntos de un conjunto base tal que dos conjuntos cualesquiera de la familia o son disjuntos, o uno de ellos está contenido en el otro), o bien formalmente a través de un árbol (grafo no dirigido que carece de ciclos) enraizado (no un nodo distinguido) tal que la raíz es la membrana piel y las hojas son las membranas elementales.

Las sustancias químicas que están contenidas en el interior de las membranas celulares (concretamente, en las distintas regiones) son representadas mediante símbolos sobre un alfabeto finito. Teniendo presente que las diferentes sustancias químicas pueden aparecer en una región (compartimento) en diferentes cantidades, representaremos el contenido de cada región a través de *multiconjuntos* de objetos; es decir, conjuntos de objetos con multiplicidades asociadas a cada uno de ellos.

Las reacciones químicas que se producen en las regiones se representan a través de las denominadas *reglas de evolución* que, básicamente, son reglas de reescritura de un sistema formal y que, como tal, actúan sobre los objetos modificándolos y/o trasladándolos de un compartimento a otro adyacente de acuerdo con una determinada semántica. Así pues, cada compartimento celular puede ser considerado como una unidad de cálculo (es decir, como un procesador) ya que tiene sus propios datos de trabajo (materializado por las sustancias químicas que contiene) y su propio programa local (implementado a través de las reglas de evolución). En consecuencia, la propia célula como entidad global puede ser vista como un *dispositivo de cálculo no convencional*.

Una *configuración* de un sistema P consta de una estructura de membranas y una familia de multiconjuntos de objetos asociados a cada región. En la descripción formal de un sistema celular, aparece siempre una configuración denominada *inicial*. En algunas variantes de sistemas celulares, existe una *membrana de entrada* que permite añadir un nuevo multiconjunto de objetos antes de que se produzca evoluciones, dando lugar a lo que se denomina *configuración inicial asociada a un dato de entrada*.

Las reglas de evolución serán aplicadas de manera *no determinista, paralela y maximal*; es decir, en cada paso de transición las reglas a ser usadas y los objetos involucrados en la misma deben verificar una condición esencial: tras la ejecución de las reglas no puede quedar ningún objeto por evolucionar al que se le pueda aplicar alguna regla. Los objetos pueden pasar de una región a otra a través de las membranas y éstas pueden ser disueltas, creadas o divididas. De esta manera se produce una *transición* de una configuración del sistema a otra.

Una *computación* en un sistema será una sucesión (finita o no) de configuraciones tal que la primera de ellas sea una configuración inicial y, además, toda configuración de la sucesión que no sea la inicial se obtiene de la anterior mediante un paso de transición. Se dice que una configuración es de *parada* si no existe ninguna regla de evolución que sea aplicable a dicha configuración. Una *computación de parada* es una computación que alcanza una configuración de parada. Generalmente, a cada computación de parada se le asocia un *resultado* que suele estar codificado en una membrana distinguida (denominada de *salida*) a través del contenido de la misma. En el caso de sistemas P denominados *con salida externa* el resultado de una computación de parada está codificado en el entorno exterior de la célula. Así pues, una computación de un sistema celular parte de una configuración inicial a partir de cual y mediante sucesivas transiciones o bien se llega a una situación de parada (en tanto que el sistema no puede seguir evolucionando) y devuelve un resultado, o bien, el sistema nunca se detiene en tanto que sigue evolucionando de manera indefinida.

Admitiremos que el proceso llevado a cabo en una computación está sincronizado; es decir, que existe una especie de reloj global que marca las unidades de tiempo a todo el sistema.

14.5. Clases de Complejidad en el marco de los sistemas P

En Teoría de la Complejidad suele trabajarse fundamentalmente con problemas abstractos que se denominan *de decisión*; es decir, problemas tales que sus respuestas son *Sí* o

No. Recuérdese que esto no implica ninguna restricción importante ya que todo problema abstracto de optimización tiene una versión de decisión tal que para cada solución de la versión de decisión puede construirse, con una cantidad pequeña de recursos computacionales, una solución del problema de optimización.

En el marco de los sistemas P existe una variante especial para atacar la resolución de los problemas de decisión: son los sistemas P que se denominan *reconocedores*.

Diremos que un sistema P es *reconocedor* si se satisfacen las siguientes condiciones: (a) el alfabeto que representa los símbolos del sistema posee dos símbolos distinguidos *Yes* y *No*; (b) existe una membrana distinguida (denominada membrana de *entrada*) que permite codificar los datos de entrada del sistema P; (c) todas las computaciones del sistema son de parada; y (d) un objeto *Yes* o un objeto *No* es enviado al exterior y, además, únicamente en el último paso de la computación (ese símbolo codifica la salida). En los sistemas reconocedores, una computación se dice que es de *aceptación* si el objeto *Yes* es enviado al entorno en el último paso; en caso contrario, la computación se dice que es de *rechazo*.

Diremos que un sistema P reconocedor es *confluyente* si para cada dato de entrada del sistema, todas las computaciones del sistema devuelven el mismo resultado; es decir, o bien todas son de aceptación (en cuyo caso, se dice que el sistema acepta la entrada), o bien todas son de rechazo (en cuyo caso, se dice que el sistema rechaza la entrada).

A partir de estas definiciones se introduce la clase $\mathbf{PMC}_{\mathcal{R}}$ de todos los problemas de decisión que son resolubles en tiempo polinomial usando familias de sistemas P reconocedores de la clase \mathcal{R} (una definición formal se puede encontrar en [5]). Si se considera la clase \mathcal{T} de los sistemas P reconocedores básicos de transición, en los que sólo se admiten reglas que no incrementan el tamaño de la estructura inicial de membranas (por ejemplo, reglas de evolución, de comunicación o de disolución), entonces se ha demostrado que la clase de complejidad $\mathbf{PMC}_{\mathcal{T}}$ coincide con la clase \mathbf{P} de los problemas tratables. Es decir, usando familias de sistemas reconocedores básicos de transición únicamente es posible resolver en tiempo polinomial problemas de decisión que sean tratables (para más detalle, ver [2]).

Supongamos ahora que en un sistema P reconocedor se admite una regla (denominada de *división*) que es una abstracción del proceso de mitosis celular. Dicha regla tiene el formato genérico $[a]_h \rightarrow [b]_h [c]_h$ y se podrá aplicar en una membrana identificada por la etique h si en ella está presente el objeto a . Su aplicación genera dos nuevas membranas que tienen la misma etiqueta y de tal manera que en una de ellas el objeto a es reemplazado por b y en la otra a es reemplazado por c . Además, todos los objetos restantes que estuvieran en la membrana inicial se replicarían en las dos membranas nuevas. Así pues, en tales sistemas P será posible generar una cantidad de espacio exponencial (expresado en términos de membranas y objetos) en tiempo lineal.

Un sistema P reconocedor con membranas activas (sin cargas eléctricas) está caracterizado por el tipo de reglas que pueden ser usadas: (a) reglas de evolución $[a]_h \rightarrow [u]_h$; (b) reglas de comunicación $[a]_h \rightarrow b []_h$ y $a []_h \rightarrow [b]_h$; reglas de disolución $[a]_h \rightarrow b$; y (d) reglas de división $[a]_h \rightarrow [b]_h [c]_h$; en donde a, b, c son símbolos del alfabeto y u una cadena o palabra del mismo. Se ha demostrado que:

- Existen problemas \mathbf{NP} -completos que pueden ser resueltos de manera eficiente por familias de sistemas P reconocedores con membranas activas y sin polarizaciones (para más detalles, ver [1]).

- Las familias de sistemas P reconocedores con membranas activas y sin polarizaciones en las que se *prohiben las reglas de división* únicamente pueden resolver problemas tratables de manera eficiente (para más detalles, ver [2]).

De ambos resultados se deduce que en el marco de los sistemas P reconocedores con membranas activas y sin polarizaciones la regla de división proporciona una frontera entre la eficiencia y la no eficiencia, bajo el supuesto de que $\mathbf{P} \neq \mathbf{NP}$. En consecuencia, podemos afirmar que en el marco computacional antes citado una abstracción de la mitosis celular constituye una frontera de la tratabilidad de problemas y, por tanto, proporciona una herramienta interesante para poder atacar la resolución de la conjetura $\mathbf{P} \neq \mathbf{NP}$ en un marco computacional bio-inspirado y no convencional.

Agradecimiento

Este trabajo se ha realizado en el marco de la ejecución del proyecto de investigación I+D+I del Ministerio de Ciencia e Innovación TIN2009-13192, cofinanciado por fondos FEDER, y del proyecto de excelencia con investigador de reconocida valía P08-TIC-04200 de la Junta de Andalucía.

Bibliografía

- [1] A. Alhazov, M.J. Pérez-Jiménez. Uniform solution of QSAT using polarizationless active membranes. In J. Durand-Lose and M. Margenstern (eds.) *Machines, Computations, and Universality. Lecture Notes in Computer Science*, **4664** (2007), 122–133.
- [2] M.A. Gutiérrez, M.J. Pérez-Jiménez, A. Riscos, F.J. Romero, A. Romero. Characterizing tractability by cell-like membrane systems. In K.G. Subramanian, K. Rangarajan, M. Mukund (eds.) *Formal models, languages and applications*, World Scientific, Series in Machine Perception and Artificial Intelligence - Vol. 66, 2006, chapter 9, pp. 137–154.
- [3] Gh. Păun, Computing with membranes, *Journal of Computer and System Sciences*, **61**, 1 (2000), 108–143, and *Turku Center for Computer Science-TUCS Report No. 208*, 1998 (www.tucs.fi).
- [4] M.J. Pérez-Jiménez; F. Sancho-Caparrini. A formalization of transition P systems. *Fundamenta Informaticae*, **49** (2002), 261–272.
- [5] M.J. Pérez Jiménez, A. Romero Jiménez, F. Sancho Caparrini. Complexity classes in models of cellular computing with membranes. *Natural Computing*, **2**, 3 (2003), 265–285.