

# Paradigmas alternativos de computación: paralelismo y simplicidad inspirados en la biología

Sergio A. Rojas

## RESUMEN

A pesar que la computación secuencial ha alcanzado avances extraordinarios en nuestra historia reciente, no ha podido demostrar su aplicación exitosa en problemas complejos donde se requiera máquinas que funcionen de manera autónoma, robusta y eficiente, dentro de ambientes dinámicos y amenazantes. Existen otros paradigmas alternativos de cómputo, agrupados en lo que se ha denominado bioinformática y computación celular, los cuales tienen una motivación fundamentalmente biológica, y ciertamente han mostrado resultados prometedores. En este artículo se abordan sus principales características y se mencionan algunos avances.

**Palabras claves:** computación secuencial, paralelismo, computación celular.

## ALTERNATIVE PARADIGMS FOR COMPUTING: PARALLELISM AND SIMPLICITY INSPIRED IN BIOLOGY

## ABSTRACT

While sequential computers have achieved astonishing progress in our recent history, it can't demonstrate successful application in complex problems where autonomous, robustness and efficient machines are required to operate in a threatening and dynamic environment. There are another alternative computing models that become as cellular computing, motivated primary in biologics. And certainly, they had shown good promises. This paper is a reviewing of this issue, its features and some advances.

**Keywords:** sequential computing, parallelism, cellular computing.

## INTRODUCCIÓN

La computación ha sido uno de los avances más significativos en toda la historia de la civilización, a pesar de ser un desarrollo relativamente reciente. Aunque los intentos por construir máquinas capaces de computar (o al menos calcular) son longevos y numerosos, solo a partir de mediados del siglo XX, se logró establecer un modelo claro y preciso de máquina de cómputo que aprovechara los adelantos alcanzados en las ciencias de la informática y de la electrónica: el computador de arquitectura secuencial (es decir que ejecuta algoritmos paso por paso para resolver problemas). Invento fabuloso para especialistas y laicos que estupefactos pudieron presenciar como se instituyó como un pilar fundamental del mundo moderno dentro de la era de la información y el conocimiento. Inclusive muchos lo catalogan de invención del siglo [1], y como hito dentro de la evolución cultural de la humanidad, al lado del fuego y la escritura [2].

Bajo la promesa de insuperables cualidades de velocidad, precisión, capacidad de almacenamiento y economía de insumos, el modelo secuencial facilitó la aplicación de la informática a la resolución de problemas básicos y medianamente complejos en una amplia gama (¿toda?) de oficios realizables por los seres humanos; sin embargo, persisten muchos problemas, la mayoría de ellos complejos, que no pueden ser resueltos, o pueden serlo, pero de manera mediocre, utilizando el más potente computador secuencial disponible en nuestra humanidad: El movimiento perfecto de una extremidad como su mano, la identificación del rostro de sus padres dentro de una multitud, la autorreparación de un tejido de células, la autonomía para tomar una decisión acertada, la capacidad de extraer

La computación ha sido uno de los avances más significativos en toda la historia de la civilización.

conocimiento a partir de la experiencia. Esta es razón suficiente para cuestionarse por caminos alternativos que mejoren aún mas nuestras herramientas tecnológicas.

## I. EL MODELO DE CÓMPUTO SECUENCIAL

Turing [3] fue el visionario que propuso el modelo mediante el cual una máquina compuesta de una cabeza lectora, una cinta infinitamente larga y un conjunto finito de estados, podía resolver un problema lógico, de manera algorítmica. ¡Eureka! Nacimiento del computador secuencial, ese que está en la oficina del jefe, en la estación lunar, en el horno microondas y en uno que otro bolsillo de nuestros ejecutivos.

El mismo Turing la bautizó elegantemente: Máquina de Estado Finito (MEF). Para entender que su funcionamiento, corresponde a lo que ahora se conoce como programa o algoritmo de computador, se puede revisar el siguiente ejemplo (adaptado de [4]): Calcular la suma  $2+1$ . Los números se escriben en la cinta como una cadena de  $n+1$  unos (o sea que 2 se escribe 11). La MEF se comporta de acuerdo con unas reglas de transiciones entre estados y acciones (mostrada en la Tabla I), que consulta cada vez que la cabeza lectora lee un estado de la cinta infinita.

Tabla I. Transiciones entre estados de la MEF sumadora

Estado	Lectura	Acción	Nuevo estado
Dormido	Blanco (" ")	Saltar derecha	Dormido
	Uno("1")	Saltar derecha	Empiece
Empiece	Blanco (" ")	Escriba "1"	Siga
	Uno("1")	Saltar derecha	Empiece
Siga	Blanco (" ")	Saltar izquierda	Borre
	Uno("1")	Saltar derecha	Siga
Borre	Blanco (" ")	-	-
	Uno("1")	Escriba " "	Retro-borre
Retro-borre	Blanco (" ")	Saltar izquierda	Retro-borre
	Uno("1")	Escriba " "	Acabe
Acabe	Blanco (" ")	Pare	Acabe
	Uno("1")	Pare	Acabe

Tabla I. Transiciones entre estados de la MEF sumadora

El funcionamiento de la MEF sumadora se observa en la figura 1. En la secuencia 1) la MEF pasa de un estado D (Dormido) a E (Empiece) y la cabeza comienza a leer "1" hacia la derecha, hasta que llega al primer espacio en blanco. En ese momento, secuencia 2), la MEF cambia al estado S (Siga) y escribe "1" en la cinta y continua leyendo hacia la derecha hasta encontrar otro blanco, momento en

que la máquina cambia al estado B (Borre). En 3) se ve que la cabeza se mueve a la izquierda una posición, escribe un blanco y la MEF cambia al estado R (Retro-Borre) y continua leyendo hacia la izquierda. La secuencia 4) muestra que la cabeza escribe un blanco y cambia al estado A (Acabe). Por último, ya en 5) la MEF ha terminado la operación y se mantiene en el estado de quietud; resultado:  $2+1 = 3$  (o sea 111). Cualquier problema que pueda resolverse de manera algorítmica, puede ser implementado en una MEF con un adecuado conjunto de estados y transiciones.

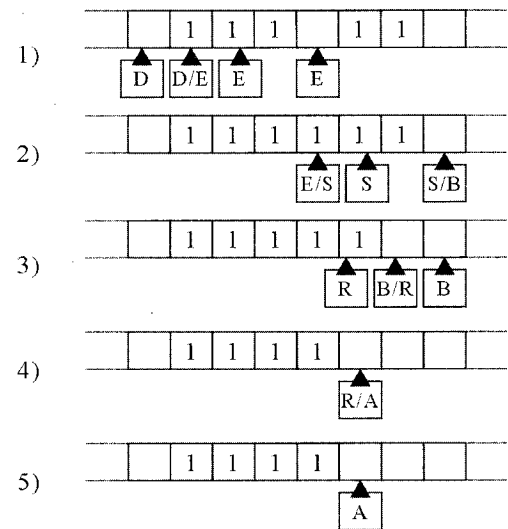


Figura 1. Secuencia seguida por la MEF para sumar  $2+1$

## II. PARADIGMAS ALTERNATIVOS DE CÓMPUTO

El paradigma secuencial inspiró el diseño de los populares computadores personales y empresariales, con el aditamento que el procesador se hizo más complejo y más rápido. Pero no sobra mencionar, que casi al mismo tiempo de su nacimiento, se propusieron otros modelos válidos para hacer computación (de ahí que sean señalados como "alternativos", en vez de "nuevos" o "novedosos"), aunque no tuvieron hasta ahora la fortuna de ser tan famosos ni utilizados como su hermanito secuencial. El hecho es que el modelo secuencial no es hijo único, y se puede ubicar en uno de los vértices del cubo (tomado de [5]) que se muestra en la figura 2, cuyas dimensiones son: son la secuencialidad de las operaciones (S), la complejidad de los procesadores (C) y la disposición de los datos (G).

Cualquier problema que pueda resolverse de manera algorítmica, puede ser implementado en una MEF con un adecuado conjunto de estados y transiciones.

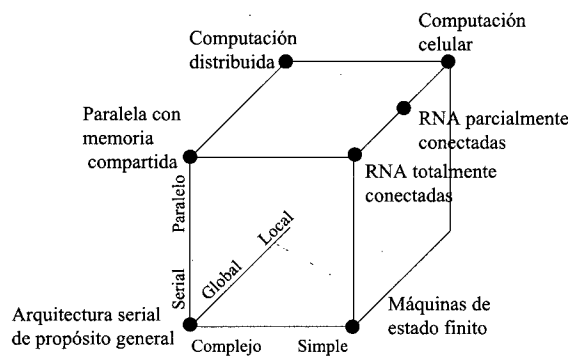


Figura 2. Taxonomía de los paradigmas de computación

El famosísimo computador de escritorio, una máquina serial, compleja y con acceso global a toda la memoria del sistema, está ubicado en el vértice inferior izquierdo. Al subir un poco por el eje S, se encontraría las máquinas modernas que tienen varios procesadores especializados (para operaciones matemáticas, para procesamiento de multimedia, para comunicaciones, etc.).

Al extremo de este eje están ubicados los supercomputadores masivamente paralelos, aquellos basados en arquitecturas SIMD y MIMD, que ciertamente han demostrado ser muy buenos para resolver complicados cálculos aritméticos (establecer la fecha y hora en la que el cometa Halley nos volverá residuo espacial o encontrar la próxima jugada con la que se puede dejar boquiabierto a Kasparov). A este mismo nivel, pero profundizando en el eje G, se encuentran los nuevos sistemas de computación distribuidos (*clusters*), que tienen estaciones complejas de cómputo con acceso local a los datos y comunicados mediante protocolos de red de alto y bajo nivel; aún no son muy conocidas sus ventajas potenciales, se está investigando en ello.

En la esquina inferior derecha se encuentran las teóricas MEF de Turing. Subiendo por el eje S está una arquitectura interesante: las redes neuronales artificiales (RNA). Estos computadores simulan el comportamiento del cerebro humano: muchas neuronas interconectadas entre sí (total o parcialmente), con un programa celosamente almacenado y codificado en las sinapsis de dichas conexiones, y que reproducen cualidades complejas de los hombres como el reconocimiento de patrones visuales y sonoros, la extracción de rasgos fundamentales y el aprendizaje en tiempo real. Este es uno de los primeros aportes de la biología a la teoría computacional (después de von Bertalanffy y su TGS).

La naturaleza, a lo largo de la evolución, ha podido encontrar soluciones efectivas para los problemas complejos que ha enfrentado (la mejor prueba la encontramos en la máquina cuasiperfecta que llamamos "ser humano").

Desde ese punto, yendo en profundidad por el eje G, se encuentra el paradigma de interés para este artículo: la computación celular. Como se puede observar, sus tres características claves son: simplicidad, paralelismo extremo y localidad.

### III. COMPUTACIÓN CELULAR

La naturaleza, a lo largo de la evolución, ha podido encontrar soluciones efectivas para los problemas complejos que ha enfrentado (la mejor prueba la encontramos en la máquina cuasiperfecta que llamamos "ser humano"). ¿Por qué no aprovechar estas herramientas para reproducirlas en un medio artificial, orientado al desarrollo de nuevas aplicaciones que contribuyan al avance de nuestra civilización? Esta nueva inspiración biológica aparece como un camino distinto para el diseño de sistemas complejos [6].

Y esta es una característica adicional de la computación celular, quizás la que la diferencia de los demás paradigmas.

La vida ha logrado resolver muchos problemas complejos utilizando procesos relativamente simples, con un grado de paralelismo extremadamente alto. Algunos ejemplos saltan a la vista:

- El cerebro analiza problemas secuenciales mediante una plataforma vastamente paralela y conexionista; no es posible reducir ese potente dispositivo a una simple MEF como lo plantea el modelo secuencial. O sea, sería tremendamente complicado hacer un cerebro con una MEF.
- Una especie animal de hoy en día, ha sobrevivido gracias a que miles de individuos han estado y están experimentando, aprendiendo y multiplicándose, todos al mismo tiempo, explorando múltiples "soluciones" de su problema (el cual es nada más y nada menos que sobrevivir).
- Un hormiguero es una organización maravillosa que puede ser vista como un superorganismo que opera mediante pequeñas contribuciones dadas por componente elementales (las hormigas). Al ser considerados como un todo, alcanzan una complejidad increíble. Por esta razón, eliminar una hormiga es insignificante

para el hormiguero, es como quitarle una hoja a un árbol.

- El ADN ha logrado codificar el algoritmo de construcción de un ser vivo a punta de millones y millones de ensayos, errores, aciertos y uno que otro chiripazo a lo largo de una cadena evolutiva.

Todos estos son elementos que se aprovechan en el paradigma de la computación celular y que se han conjugado en forma de diversas herramientas. Su estudio ha tenido un inusitado entusiasmo y un vertiginoso desarrollo en los últimos años: RNA [7], algoritmos genéticos (AG) [8], programación genética [9], autómatas celulares [10], computación con ADN [11], animales artificiales [12], hardware evolutivo [13].

Nombres extraños dirán algunos, atractivos o escandalosos dirán otros, pero todos prometedores y extrañamente efectivos. Las características que comparten estos enfoques se enmarcan en:

- **Simplicidad:** La unidad de procesamiento es sencilla, ejecuta algún conjunto de reglas simples, operaciones elementales o recombinaciones.
- **Paralelismo:** En contraste con los computadores distribuidos o los masivamente paralelos, que a lo sumo incluyen docenas o cientos de procesadores, en la computación celular se parte del principio "más es mejor" y el número de células se puede contar en potencias de diez. Por esta razón se ha establecido como un paradigma de cómputo *vastamente* paralelo.
- **Localidad:** Cada procesador se conecta usualmente a nivel local, solo con sus vecinos más cercanos, no necesitan conocer datos de toda la configuración. Sin embargo, el comportamiento global resultante posee una complejidad asombrosa.

Concebido inicialmente por vonNeumann en 1940, un autómata celular es un arreglo bidimensional de celdas, que se encuentran en un conjunto finito de estados.

## IV. AUTÓMATAS CELULARES

Quizás uno de las herramientas más representativas de este paradigma, porque expone claramente las características mencionadas anteriormente, son los autómatas celulares. Concebido inicialmente por vonNeumann en 1940, un autómata celular es un arreglo bidimensional de celdas, que se encuentran en un conjunto finito de estados.

Las celdas se actualizan de manera sincrónica en pasos discretos de tiempo de acuerdo con una serie de reglas que operan sobre el estado de sus vecinas.

Para introducir algunos de sus conceptos, un ejemplo divertido es el Juego de la Vida [14]. Se trata de establecer, para un conjunto de células ubicadas en una rejilla cuadrículada, si cada una está viva o muerta.

Para ello, se examinan los estados de ocho vecinas más cercanas y se siguen las siguientes reglas:

- Una célula muerta regresa a la vida si tiene tres vecinos vivos alrededor (aunque cada célula es asexual, se requiere un trío para procrear, ¿liberadas?)
- Una célula viva con cuatro o más vecinos vivos, muere por hacinamiento.
- Una célula viva con solo un vecino vivo, muere por abandono.

Dada una configuración inicial aleatoria de la población, su evolución durante varias generaciones (en cada generación se actualiza el estado de todas las células), conduce a encontrar patrones complejos y muchas veces estables.

Los experimentos con este juego han encontrado patrones que representan relojes, patines móviles, pistolas, trenes y ¡hasta computas capaces de

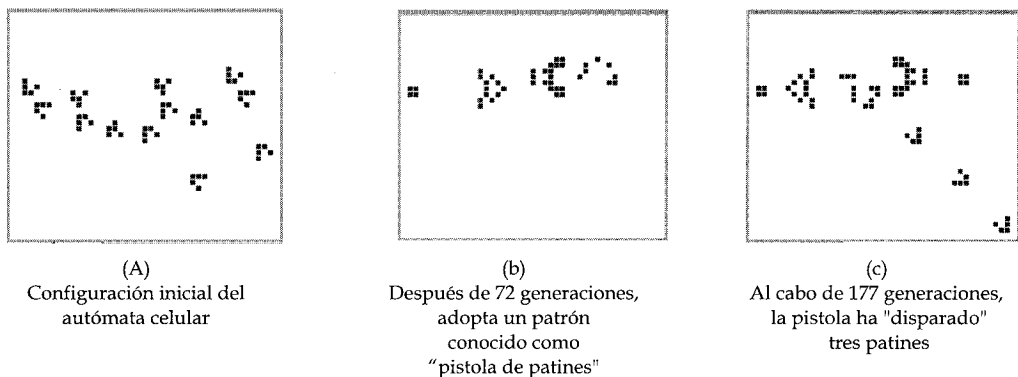


Figura 3. Autómata del Juego de la vida que exhibe un patrón oscilante

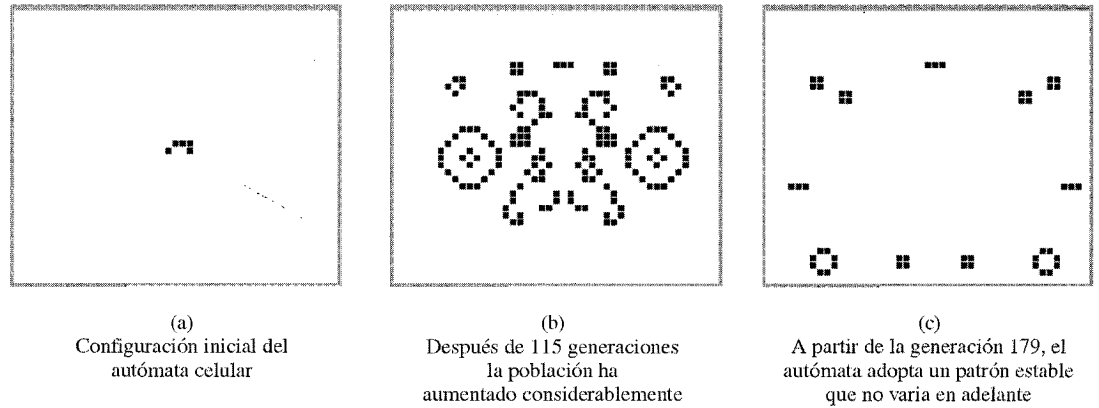


Figura 4. Autómata que sigue un patrón explosivo al comienzo y estable al final

simular las operaciones AND y NOT!

En las figuras 3 y 4 se observan ejemplos de la evolución del Juego de la Vida con patrones oscilantes y estables.

## V. CARACTERÍSTICAS DEL PARADIGMA

Dentro del paradigma de computación celular se pueden destacar algunos aspectos que caracterizan a las herramientas utilizadas para la construcción de sistemas bajo este enfoque:

- El funcionamiento es paralelo y no depende de una unidad central de procesamiento. Cada procesador puede estar representado por una celda/célula, una neurona, un circuito, un agente, un genoma o un programa.
- Los procesadores pueden procesar valores discretos (estados) o funciones analógicas con valores continuos.
- El comportamiento puede darse mediante una tabla de búsqueda exhaustiva de reglas estado-acción, mediante un algoritmo o una función parametrizada.
- Las células pueden estar inmóviles procesando información o pueden moverse dentro de su ambiente y/o hasta reproducirse.
- Pueden comunicarse mediante rejillas regulares (cuadriculadas, hexagonales e inclusive en 3D) o irregulares, expresadas mediante grafos.
- La información transmitida entre procesadores es simple, no se requieren complicados protocolos de comunicación.
- Los procesadores pueden actuar al tiempo o

Las células pueden estar inmóviles procesando información o pueden moverse dentro de su ambiente y/o hasta reproducirse.

asincrónicamente.

- Las celdas no son todas homogéneas. Pueden tener diferentes estados o transiciones.
- Los programas que ejecutan pueden ser determinísticos y orientados a la solución, mediante algún estudio preliminar, o pueden ser simples reglas de convivencia local que permitan alguna forma de evolución y que permitan que el sistema se adapte para encontrar una solución global emergente.
- La implementación puede ser en software de simulación, con hardware de propósito especial, con procesadores configurables y adaptativos, computación en tubos de laboratorio, o inclusive a nivel cuántico, jugando con los estados de los electrones.
- Su arquitectura es escalable y altamente robusta.

## VI. DESARROLLOS Y AVANCES

Encriptamiento de mensajes; modelos físicos de difusión, transmisión de enfermedades, y movimiento de gases, se han logrado desarrollar con autómatas celulares. Otras aplicaciones incluyen la extracción de líneas de contorno en imágenes digitalizadas mediante redes neuronales celulares; sumadores de números representados como poblaciones iniciales de un autómata celular posiblemente implementado a escala nanométrica; generadores celulares de números aleatorios; solución a problemas NP como el agente viajero, mediante hormigas artificiales; solución al problema de satisfacibilidad (encontrar los valores que deben tomar variables booleanas para hacer válido un

predicado lógico) mediante bucles de moléculas de ADN que se propagan y reproducen en un tubo de laboratorio; búsqueda de relaciones insospechadas de gusto por gastar de clientes en una bodega de datos utilizando AG; solución al problema del camino Hamiltoniano (similar al del agente viajero) mediante computación molecular; elaboración de robots autónomos que aprenden mediante la experiencia sin supervisión; asignación y programación óptima de recursos con AG; prótesis de extremidades artificiales que se adaptan a la actividad muscular del paciente en vez de ser al contrario; por mencionar someramente algunas.

## CONCLUSIONES

Un paradigma diferente muchas veces no surge como contrapuesto con los existentes, sino como coadyuvante. Este es un caso. No podemos suponer que la computación celular pueda reemplazar a la computación secuencial; seguro que una red neuronal no puede construir un reporte de ventas tan bien como lo puede hacer un motor de base de datos. Pero si puede servir como complemento; y si el objetivo es el avance de nuestra civilización, los diferentes esfuerzos son bienvenidos.

En virtud de las necesidades cada vez más exigentes de nuestra sociedad del conocimiento, nuestras propuestas deben ser más ingeniosas, explotando los desarrollos que hemos alcanzado hasta el momento, y explorando nuevos caminos promisorios.

Lo importante es, que a pesar de no ser un paradigma reciente, ni tampoco novedoso, la computación celular nos ofrece una alternativa totalmente diferente al estilo clásico que nos ha acompañado en las últimas décadas.

La naturaleza nos enseña que existen soluciones maravillosas para problemas extremadamente difíciles. Poder aprender de ella, y utilizar ese conocimiento para aplicarlo a medios artificiales es nuestro gran reto. Es el momento justo para comenzar a afrontarlo en nuestras universidades y centros de investigación. Todavía nos espera bastante tierra por arar, pero el clima promete una cosecha abundante.

Quizás lo más interesante es que con un poco de dedicación y disciplina, estamos en capacidad de

hacer aportes al desarrollo de este paradigma. ¿Podemos hacer parte del derrocamiento del "señor dictador computador secuencial"! Y establecer un mecanismo más participativo, en cuanto computación se refiere.

## REFERENCIAS

- [1] Lucky, R. (2000) "Greatest engineering accomplishments". En: *Spectrum*, vol. 37, 3, Marzo.
- [2] Moravec, H. (1988). *Mind Children*. Harvard University Press. Citado en: Stan Franklin, *Artificial Minds* (1995). Cambridge, MA: MIT Press.
- [3] Turing, A. (1950). "Computing Machinery and Intelligence". En: *Mind*, 59.
- [4] Martínez, J. (1999). "Vida Artificial, el Nuevo Paradigma". En: *Ingeniería e investigación*, 40.
- [5] Sipper, M. (1999). "The emergence of Cellular Computing". En: *Computer*, Junio.
- [6] Martínez, J.J.; Rojas, S.A. (1999). *Introducción a la Informática Evolutiva: Un nuevo enfoque para resolver problemas de Ingeniería*. Publicaciones de Ingeniería de la Universidad Nacional.
- [7] Haykin, S. (1994). *Neural Networks, a Comprehensive Foundation*. Prentice-Hall.
- [8] Goldberg, D. (1989). *Genetic Algorithms*. Cambridge, MA: Addison-Wesley.
- [9] Koza, J. (1992). *Genetic Programming*. Cambridge, MA: MIT Press.
- [10] Toffoli, T; Margolus, N. (1987). *Cellular Automata Machines*. Cambridge, MA: MIT Press.
- [11] Adleman, L. (1994). "Molecular computation of solutions to combinatorial problems". En: *Science*, 266.
- [12] Cliff, D.; Husbands, P.; Meyer, J.; Wilson, S. (1994). *From Animals to Animats 3: Proceedings of the Thurd International Conference on Simulation of Adaptive Behavior*. Cambridge, MA: MIT Press.
- [13] Sipper, M.; Ronald, E. (2000). "A new species of hardware". En: *Spectrum*, vol. 37, 3, Marzo.
- [14] Conway, J. *Winning Ways*. Academic Press, 1982.



### Sergio Andrés Rojas Galeano

Ingeniero de Sistemas, U. Nacional de Colombia. Especialista en Ingeniería de Software, U. Distrital. Profesor de la Facultad de Ingeniería, Universidad Distrital FJC. Correo electrónico: sar@colombiamix.com

La naturaleza nos enseña que existen soluciones maravillosas para problemas extremadamente difíciles. Poder aprender de ella, y utilizar ese conocimiento para aplicarlo a medios artificiales es nuestro gran reto.