

# Computación interactiva y emergente en colonias de hormigas

**Nelson Alfonso Gómez-Cruz**

Centro de Innovación, Escuela de Administración, Universidad del Rosario  
Laboratorio de Sistemas Inteligentes, Facultad de Ingeniería, Universidad Nacional de Colombia

[nelson.gomez@urosario.edu.co](mailto:nelson.gomez@urosario.edu.co)

Orcid: 0000-0001-9594-1441

Recepción: 29/10/2020

Aceptación: 18/11/2020

## Resumen

Los sistemas vivos procesan información; esto es lo que hacen para vivir. La capacidad de procesar información es, de hecho, uno de los rasgos más sobresalientes de los sistemas complejos adaptativos en general. Se suele aceptar, desde la biología y las ciencias de la computación, que el modelo estándar de computación algorítmica, representado por la máquina de Turing, establece los límites teóricos de lo que la vida puede computar y lo que no. Pese a ello, el tipo de computaciones que realizan los sistemas vivos difiere en maneras fundamentales de las soportadas por la máquina de Turing. En este artículo se demuestra que el sistema de navegación que emplean individualmente las hormigas y las estrategias colectivas para la recolección óptima de alimentos que usa la colonia no se pueden reducir a formas de computación algorítmica. Esta idea justifica la necesidad de desarrollar nuevos modelos de computación que nos permitan desentrañar la lógica computacional y la complejidad de la vida.

## Palabras clave

Sistemas complejos; computación interactiva; computación emergente; colonias de hormigas; forrajeo; sistema de navegación en hormigas

## Introducción: complejidad y cognición en las colonias de hormigas

Actualmente, se estima que existen entre 14.000 y 20.000 especies de hormigas que se extienden a través de todos los hábitats terrestres del planeta (Gordon, 2016a; Hölldobler & Wilson, 2014). Algunas de ellas llegan a conformar colonias de hasta 300 millones de individuos. Su comportamiento social está soportado en sistemas de comunicación sofisticados que involucran señales químicas y otras señales complejas en las que intervienen el olfato, el gusto, las vibraciones (sonidos) y el tacto. Desde el punto de vista ecológico, el comportamiento colectivo de una colonia es equiparable al de un organismo (Gordon, 2010): nace, crece, se reproduce y muere. La expectativa de vida de una colonia oscila entre 25 y 30 años (Gordon, 2016a). Debido a su organización coherente y cooperativa, a las colonias de hormigas se les denomina *superorganismos* (Hölldobler & Wilson, 2014).

Las colonias de hormigas han sido reconocidas, desde hace algunas décadas, como

*sistemas complejos adaptativos* (SCA) (Gordon, 2010; Mitchell, 2009; Dorigo & Stützle, 2004; Bonabeau, 1998; Hölldobler & Wilson, 1990). Un SCA es aquel cuya organización, dinámica y función resultan de la acción autónoma, paralela, distribuida, asíncrona e interactiva entre sus múltiples componentes. Los

SCA exhiben, de manera fundamental, procesamiento sofisticado de información y adaptación mediante aprendizaje o evolución (Gómez-Cruz, 2013; Mitchell, 2009). "Para comprender cómo la acción coordinada de los componentes logra desplegar patrones colectivos de organización espaciotemporal, es necesario considerar por lo menos dos niveles diferentes de observación. La escala local (o microescala), donde operan e interactúan los componentes del sistema. Y la escala global (o macroescala), donde aparecen los *patrones agregados*, que pueden ser estructurales, comportamentales o funcionales" (Gómez-Cruz, 2018).

En la microescala, las capacidades funcionales de cada hormiga son restringidas con respecto a las demandas del ambiente; su capacidad para acceder a información del entorno es incompleta, y su interacción, que puede ser directa o mediada por el entorno, es estrictamente local. En la macroescala, por su parte, surge una red de procesamiento de información que opera de manera *autoorganizada*, es decir, sin la acción de un líder o de un controlador externo. De la red *emergen* patrones agregados complejos que ninguna hormiga podría llevar a cabo por sí sola. Algunos patrones comunes incluyen la construcción, reparación o extensión del nido; la búsqueda y recolección de alimento; la clasificación, alimentación y cuidado de los huevos, las larvas y las pupas; la asignación dinámica de tareas, entre otros (Gordon, 2016a; 2016b; Bonabeau, Dorigo & Theraulaz, 1999).

Adicionalmente, algunas especies llevan a cabo funciones especializadas que han evolucionado a partir de contextos ecológicos y fisiológicos particulares (Gordon, 2016b). Las hormigas cortadoras (*Atta*), por ejemplo, cortan y transportan las hojas de las plantas para cultivar hongos dentro del nido que luego consumen y brindan como alimento a las crías. Las hormigas legionarias (*Eciton*) construyen puentes con sus propios cuerpos con el fin de superar precipicios y obstáculos de manera eficiente (Bonabeau et al., 1999). Las hormigas tejedoras (*Oecophylla*) elaboran sus nidos a partir de las hojas vivas de los árboles, empleando seda para unirlos. También son capaces de atrapar y transportar animales de gran tamaño, como escorpiones, u ordeñar comunidades de áfidos por periodos extensos como si fueran "ganado". Las hormigas negras (*Lasius*), por su parte, agrupan los cadáveres en pilas a la ma-

nera de "cementorios". Todos estos patrones agregados involucran capacidades computacionales y cognitivas tanto individuales como colectivas (Feinerman & Korman, 2017; Dornhaus & Franks, 2008). Desde el punto de vista cognitivo, las colonias de hormigas pueden ser entendidas como *cerebros líquidos* (Solé, Moses & Forrest, 2019). Para este tipo de sistemas no hay una arquitectura fija ni conexiones persistentes, como en el cerebro humano. En el caso de las hormigas, la movilidad incrementa los flujos de información y la estructura física del entorno se convierte en un componente fundamental de su capacidad para procesar información (ibíd.).

A pesar de la sofisticación y de la complejidad en el procesamiento de información que despliegan las hormigas, los investigadores en campos como las ciencias de la complejidad (Mitchell, 2009), la inteligencia colectiva (Bonabeau et al., 1999) o la inteligencia computacional (Dorigo & Stützle, 2004) suelen representar y modelar a las hormigas como *autómatas reflejos* (agentes simples), cuyo comportamiento es predefinido e inflexible. Entomólogos de la talla de E. O. Wilson, incluso, han considerado a las hormigas como componentes genéticamente programados para cumplir su tarea (Gordon, 2010). Esta hipótesis de trabajo, útil en algunos casos, no siempre hace justicia a las hormigas. Como Gordon (2010) apunta, el comportamiento individual no se reduce simplemente a un conjunto de respuestas fijas a señales específicas. Por el contrario, se ha mostrado que dicho comportamiento es altamente flexible e involucra aprendizaje (Dornhaus & Franks, 2008). Se sabe, por ejemplo, que las hormigas pueden modificar su esquema de comportamiento de acuerdo con el patrón de interacción que perciben en su entorno (Gordon, 2010). Tal flexibilidad requiere habilidades cognitivas elaboradas. La cognición se define como la habilidad para adquirir información del entorno (mediante los sentidos), retenerla, procesarla y tomar decisiones con base en ella (Shettleworth, 2001). En este sentido, la cognición es un fenómeno computacional.

En este artículo se introducen dos modelos de procesamiento de información en colonias de hormigas y se demuestra que, en ambos casos, la teoría de la computación algorítmica, representada por la Máquina de Turing (MT), es insuficiente para capturar la expresividad (o el poder) computacional de las hormigas. En el primer caso, se analiza el sistema de navegación que emplean a nivel individual las hormigas del desierto. Para ello se utiliza un modelo de computación interactiva secuencial conocido como MT Persistente. En el segundo caso, haciendo uso de una simulación basada en agentes, se explora la expresividad computacional que exhibe la colonia cuando busca y recolecta alimento. Los resul-

tados presentados tienen implicaciones profundas para la comprensión de lo que significa computar y para el estudio del procesamiento de información en sistemas vivos<sup>1</sup>.

## Computación individual y colectiva en las hormigas

En el nivel individual, las hormigas despliegan un amplio repertorio de capacidades cognitivas y computacionales. Cada hormiga es capaz de percibir una gran cantidad de entradas sensoriales, modular su comportamiento de acuerdo con múltiples estímulos y tomar decisiones con base en grandes cantidades de información. Se conoce, por ejemplo, que las hormigas pueden distinguir entre cientos de químicos distintos (Gordon, 2016a). Con su diminuto cerebro, que pesa alrededor de 0.1 miligramos (Ronacher, 2008), las hormigas aprenden, planean para el futuro, enseñan, toman decisiones sofisticadas y emplean herramientas (Dornhaus & Franks, 2008). El uso de herramientas, particularmente, se ha detectado en el transporte de líquidos y en la construcción de nidos. La figura 1 (izquierda) muestra a una hormiga tejedora australiana adulta (*Oecophylla*) sosteniendo una larva entre sus mandíbulas y golpeándola suavemente con sus antenas como indicación para que comience a secretar seda de sus glándulas salivares (National Geographic, 2011). La seda es empleada para unir las hojas que formarán parte del nido (derecha).



Figura 1. Uso de herramientas en hormigas tejedoras y la arquitectura de su nido. Fuente: National Geographic (2011)

Cada una de estas habilidades involucra diversos mecanismos de procesamiento de información que operan en las escalas molecular, celular, cerebral y organizacional. No obstante, la complejidad individual no suele explicar los patrones emergentes que exhibe la colonia (Bonabeau, Theraulaz, Deneubourg, Aron & Camazine, 1997). Un caso bastante estudiado es la formación de senderos para la recolección de alimento (Camazine, Deneubourg, Franks, Sneyd, Theraulaz & Bonabeau, 2001; Bonabeau et al., 1999).

Durante el proceso de búsqueda y recolección de alimento, el comportamiento funcional de las hormigas se puede considerar "simple" (Franks, 1989). El proceso opera más o menos como se indica a continuación (Wilensky & Rand, 2015). Las hormigas forrajeras abandonan el nido, caminando de forma aleatoria, en busca de alimento. Cuando una de ellas lo encuentra, toma un poco y regresa a depositarlo en el nido. Mientras retorna, la hormiga deposita una sustancia química, denominada feromona, en el camino. El rastro de feromonas se difunde en el entorno cercano con un efecto atractivo sobre las demás hormigas. Cada vez que otra hormiga que deambule sin alimento detecte el rastro, existe una mayor probabilidad de que module su comportamiento y decida seguirlo. Al hacerlo, depositará más feromonas sobre el rastro existente y este será reforzado, atrayendo más y más hormigas al nuevo sendero.

El mecanismo de "reclutamiento" de hormigas mediante feromonas genera un bucle de retroalimentación positiva. Entre más hormigas participen en la recolección, más feromonas serán depositadas y más atractivo será el sendero. De esta manera, el número de hormigas en el sendero se incrementa exponencialmente en el tiempo (Moussaid, Garnier, Theraulaz & Helbing, 2009). Sin embargo, el proceso es contrabalanceado por un bucle de retroalimentación negativa que se desencadena cuando las feromonas se evaporan, cuando el número total de hormigas disponibles es alcanzado o cuando el alimento de la fuente ha sido recolectado por completo. La figura 2 muestra la dinámica de reclutamiento para hormigas de la especie *Linepithema humile*. En ella se ilustra la emergencia del rastro de feromonas entre el nido y la fuente de alimento. Mediante este proceso, la colonia decide colectivamente qué fuente de alimento explotar y conforma un sendero para hacerlo. La información necesaria para realizar la recolección es difundida a través del entorno y no puede ser contenida o generada por una sola hormiga. Las feromonas constituyen, en la escala global, la *memoria de la colonia*.

En general, la colonia produce, por un lado, nueva información (información cogenerada) que las hormigas

<sup>1</sup> Al respecto, véase Gómez-Cruz y Niño (2020).

asumen individualmente y, por el otro, nuevos mecanismos para procesar colectivamente dicha información. La integración de la información parcial que las hormigas adquieren, procesan y comunican (en la microescala), y los mecanismos de procesamiento de información que emergen (en la macroescala), le confieren a la colonia capacidades computacionales que no están presentes en los individuos (Mitchell, 2009). Tales capacidades le permiten a la colonia, como un todo, interactuar eficientemente con el entorno y resolver problemas que individualmente serían intratables o, en el peor de los casos, indecibles. Algunos ejemplos son la búsqueda de la ruta más corta entre el nido y la fuente de alimento; la selección de la fuente de alimento más cercana entre dos o más opciones igualmente atractivas; la elección de la fuente más atractiva (por ejemplo, más abundante y nutritiva) entre dos o más alternativas; o la explotación (sub)óptima de varias fuentes de alimento disponibles (Camazine et al., 2001).

Algunas aproximaciones al procesamiento de información en colonias de hormigas se han reportado en la literatura. En el contexto de la optimización por colonias de hormigas, la capacidad de procesamiento de información de la colonia se ha sobresimplificado y reducido a su aplicación práctica en problemas de optimización y búsqueda. El enfoque en este caso ha sido ingenieril, no científico. La manera como se analizan los modelos resultantes es en términos de su complejidad computacional, es decir, de acuerdo con la cantidad de recursos computacionales (de tiempo y memoria) requeridos para resolver problemas matemáticos o ingenieriles -por ejemplo, de optimización combinatoria- que son intratables por métodos exactos (Dorigo & Stützle, 2004). El problema de la expresividad computacional no es considerado.

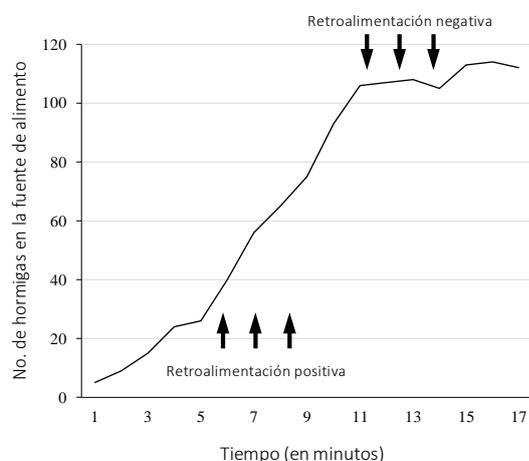


Figura 2. Dinámica de reclutamiento en hormigas de la especie *Linepithema humile*  
Fuente: Adaptada de Moussaid et al. (2009)

Otros trabajos ofrecen aproximaciones que involucran

técnicas computacionales (como la simulación) o el uso de un lenguaje computacional para comprender o explicar procesos específicos, como la formación de senderos o la emergencia de carriles (Moussaid et al., 2009). Sin embargo, en estos casos tampoco se ha estudiado de una manera explícita el problema de la expresividad computacional.

Más allá de estos casos, estudiar el hormiguero (o cualquier otro sistema vivo) como una red de procesamiento de información distribuida que opera en distintas escalas espaciotemporales sigue siendo un reto (Solé et al., 2019; Mitchell, 2012). El hormiguero comprende múltiples mecanismos que funcionan de manera simultánea, dinámica, interactiva y coordinada para que la vida misma de la colonia sea posible. Como se infiere de lo anterior, la colonia no computa ninguna función matemática específica ni se espera que su procesamiento se detenga. En general, sugiere Dodig-Crnkovic (2011), una red de procesamiento de información que funciona en un entorno dinámico no puede ser serializada ni modelada, de ninguna forma interesante o útil, con una MT.

Más aún, los modelos de computación disponibles no pueden, de momento, capturar la complejidad global de la colonia, involucrando sus múltiples escalas de emergencia y la coevolución de la miríada de mecanismos de computación presentes. Nuevos modelos de computación siguen siendo necesarios. Sin embargo, como se verá en la siguiente sección, el paradigma de la computación interactiva, aunque incipiente, permite aportar evidencia específica acerca de la expresividad computacional de las hormigas y las colonias que conforman.

## El problema de la expresividad computacional en los sistemas vivos

La máquina de Turing (MT) representa el modelo estándar de computación (Siegelmann, 2013; Cleland, 2006). Una computación en una MT se lleva a cabo a la manera de una caja negra. La computación consiste en la transformación de una entrada finita (por ejemplo, una cadena *-string*) en una salida finita, haciendo uso de una cantidad finita de tiempo. La entrada debe ser determinada antes de iniciar la computación y la salida solo estará disponible una vez que la computación haya culminado. La transformación está definida por una función matemática recursiva y se expresa mediante un algoritmo. La MT es, por lo tanto, un modelo de *computación algorítmica secuencial* (Eberbach, Goldin & Wegner, 2004). Durante décadas, muchos autores han sostenido que la MT es capaz de computar cualquier cosa computable. A este hecho se le conoce como Tesis fuerte de Church-Turing.

En el plano de la biología, la MT ha sido empleada como un modelo de computación para los sistemas vivos y como una analogía para aproximar la comprensión de la vida y sus procesos. Brenner (2012), por ejemplo, opina que las células y los organismos son MT "especiales". Para Forbes (2004), la lógica de los sistemas biológicos está acotada por las capacidades de la MT. Solé y Macia (2011), por su parte, plantean la posibilidad de modelar los ribosomas como nanomáquinas de Turing capaces de (i) leer una cinta definida por el ARN mensajero, (ii) crear una cadena de aminoácidos de salida e (iii) iniciar y terminar el proceso mediante la detección de secuencias específicas. Asimismo, distintas aproximaciones a los procesos cognitivos suponen que el pensamiento humano es equiparable a un proceso algorítmico de manipulación de símbolos, y que el cerebro es una MT natural (Barrett, 2011; Syropoulos, 2008; Newell & Simon, 1976).

Sin embargo, al contrastar los rasgos de la MT con los de los sistemas vivos, es difícil encontrar algún aspecto relevante en común (Gómez-Cruz & Maldonado, 2011). La MT es un dispositivo de procesamiento secuencial, determinístico, cerrado (durante la computación no entra ni sale información) y terminante (debe detenerse para que la computación sea exitosa). Para los sistemas vivos, en contraste, el procesamiento de información es concurrente, no determinístico, abierto, interactivo y autoorganizado. Mientras que una MT debe finalizar el procesamiento para poder decir que hubo una computación, el éxito para los procesos computacionales biológicos no es su terminación, sino la capacidad de respuesta al mundo exterior, así como la velocidad, la flexibilidad, la robustez y la adaptabilidad con la que se genera esa respuesta (Dodig-Crnkovic, 2011a). En un sistema vivo dejar de computar equivale a la muerte del sistema. Puesto en términos positivos, vivir es procesar información. Antes que comportarse como MT, los sistemas vivos se asemejan a sistemas abiertos e interactivos.

La computación interactiva (CI) es un nuevo paradigma de computación que captura los aspectos fundamentales de aquellas computaciones en las que, a diferencia de las que soporta la MT, existe comunicación (o interacción) con el mundo exterior durante el proceso de cómputo (Dodig-Crnkovic, 2010; Wegner, 1997). Fue introducida por Wegner (1997) y sus fundamentos teóricos fueron esbozados por el mismo autor en 1998. En la visión interactiva, la computación es vista como un proceso permanente (*ongoing*) de interacción con el entorno, antes que como la transformación, basada en funciones, de una entrada en una salida. La CI permite, en consecuencia, expresar sistemas abiertos, comportamientos autónomos dependientes de la historia y dinámicas autoorganizadas y emergentes. Tales aspectos

no pueden ser capturados por la lógica de primer orden que caracteriza a la computación algorítmica. De esta manera, la CI es fundamentalmente incompleta (da Costa & Doria, 2013). Nuevos formalismos lógicos son necesarios. Una alternativa sensata son las lógicas paraconsistentes. Es interesante notar que la incompletitud de los sistemas interactivos contribuye a su expresividad computacional, pero limita la posibilidad de formalizarlos (Goldin & Wegner, 2008).

Goldin, Smolka, Attie y Sonderegger (2004) introdujeron las *máquinas de Turing persistentes* (MTP) como una manera de capturar los rasgos sobresalientes de la CI considerando el mínimo número de modificaciones a una MT estándar. Las MTP fueron la primera prueba formal de que la interacción es más expresiva que los algoritmos. En esencia, las MTP logran capturar la noción intuitiva de *computación interactiva secuencial* (Ebebach, Goldin & Wegner, 2004). Una MTP no solamente extiende la sintaxis de la MT (como muchas de las variantes clásicas que resultaron ser Turing-equivalentes), sino que además extiende de manera simple, pero profunda, su semántica. La forma de hacerlo es incorporando flujos dinámicos (*dynamic streams*) y persistencia.

Una MTP es un modelo no determinístico compuesto por tres cintas semiinfinitas: una de entrada, una de lectura/escritura, denominada cinta de trabajo, y una de salida. La MTP puede ser entendida como un agente que interactúa con su entorno durante la computación a medida que procesa un flujo (*stream*) de entradas (*tokens* o *strings*) y genera un flujo de salidas. El flujo de entradas es generado dinámicamente por el entorno. El flujo de salidas, a medida que es procesado por la MTP, es emitido al entorno. Cada vez que la MTP recibe un *token* de entrada lo procesa y produce un *token* de salida. De este modo, salidas tempranas pueden afectar la estructura de entradas futuras. Este proceso se realiza sin cesar. El comportamiento de tales sistemas no es reproducible ni expresable mediante funciones recursivas.

La interacción secuencial es más expresiva que la MT. No obstante, la interacción secuencial no captura de forma apropiada la noción de emergencia, la cual sucede, por definición, en sistemas compuestos por múltiples componentes que interactúan de manera no lineal entre sí y con el entorno (Gómez-Cruz, 2018). De acuerdo con Goldin et al. (2004), la computación interactiva secuencial no admite comportamientos no serializables, como la coordinación de entradas provenientes de múltiples fuentes concurrentes. Estos autores conjeturan que un modelo de CI que considere MTP concurrentes podría ser más expresivo a una MTP secuencial. Similarmen- te, Eberbach et al. (2004) sostienen que cuando

un sistema interactivo está conformado por múltiples componentes autónomos, asíncronos y concurrentes, el sistema no puede ser simulado mediante la serialización de los comportamientos de sus componentes. Los sistemas multicomponente, específicamente los sistemas complejos basados en agentes (Grimm et al., 2005), exhiben patrones de comportamiento más ricos que aquellos que soporta un agente interactivo secuencial.

La simulación basada en agentes (Gómez-Cruz, 2018) y el modelo Actor de Carl Hewitt (2013) son modelos de computación distribuida e interactiva que modelan la emergencia en sistemas complejos. A pesar de ello, ninguno de los dos enfatiza el problema de la expresividad computacional. No existe un modelo análogo a la MTP que capture la expresividad de los sistemas interactivos distribuidos. A pesar de ello, la evidencia empírica que soporta la idea de una mayor expresividad sobre los sistemas de interacción secuencial es amplia. Eberbach et al. (2004) lo plantean en los siguientes términos:

*La interacción distribuida es más expresiva que la interacción secuencial, de la misma manera en que la interacción secuencial es más expresiva que la computación algorítmica (énfasis original) (pág. 175).*

La CI expande la noción de problema computacional e incluye dinámicas no terminantes, persistencia, dependencia de la historia, procesos no funcionales, mecanismos no algorítmicos, sistemas distribuidos y patrones emergentes. Todos estos son aspectos relevantes para los sistemas vivos. En la CI, la capacidad de un sistema para resolver problemas es sinónimo de expresividad computacional (Eberbach et al., 2004). La visión interactiva de la computación ya no se reduce a los conjuntos de funciones matemáticas que un modelo puede computar. Cuando el marco de relevancia es la biología, entonces, otros criterios de expresividad deben ser considerados. La computación natural (Kari & Rozenberg, 2008) y la computación biológica (Gómez-Cruz & Niño, 2020; Mitchell, 2012) requieren de modelos de computación que admitan nociones ortogonales de expresividad computacional. MacLennan (2004) denomina a tales modelos *computación no-Turing*.

Para un sistema vivo, la computación establece el límite entre la vida y la muerte. Computar "bien" significa vivir y la posibilidad participar en el proceso de la evolución. Computar "mal", por su parte, pone en riesgo su continuidad y lo ubica al borde del peligro y la extinción (Maldonado & Gómez-Cruz, 2015). Los criterios de expresividad deben dar cuenta de este hecho. MacLennan (2004) propone los siguientes criterios:

- Respuesta en tiempo real.

- Velocidad de respuesta.
- Generalidad de la respuesta.
- Flexibilidad en la respuesta a la novedad.
- Adaptabilidad.
- Tolerancia al ruido, el error, las fallas y los daños.

La hipótesis que se sugiere en este punto es que la complejidad de un sistema (vivo) es directamente proporcional a la sofisticación (expresividad computacional) del procesamiento de información que es capaz de llevar a cabo. En las siguientes secciones demostraremos que el procesamiento de información que realizan las hormigas, tanto individual como colectivamente, no se pueden reducir al funcionamiento de una MT.

## El sistema de navegación de las hormigas

Las hormigas y otros insectos sociales se caracterizan por ser forrajeros centrales. Esto significa que, de manera rutinaria, deben regresar al nido tras deambular por el entorno en busca de alimentos u otros recursos (Hartmann & Wehner, 1996). En muchas especies, cuando una fuente de alimento ha sido descubierta por una hormiga o un grupo pequeño de ellas, la colonia forma senderos temporales que favorecen una explotación eficiente de los recursos. El reto individual consiste en saber cómo regresar al nido tras haber encontrado por primera vez el alimento, luego de recorrer largas distancias.

Se han descrito tres estrategias empleadas en la naturaleza para resolver el problema de retornar al nido tras el forrajeo (Ronacher, 2008). Los gasterópodos (como los caracoles), por ejemplo, son capaces aplicar marcas durante el forrajeo que luego siguen de forma inversa para retornar al punto de origen (figura 3, izquierda). Varias especies de vertebrados, por otro lado, utilizan mapas cognitivos a través de los cuales logran establecer relaciones espaciales entre las estructuras que encuentran en el entorno (figura 3, derecha). Las hormigas, por su parte, emplean de manera fundamental un mecanismo denominado *integración de rutas (path integration)* (figura 3, centro). Este mecanismo consiste en la estimación, por parte de la hormiga, de las distancias recorridas y las variaciones en la orientación del desplazamiento. La información calculada es integrada en un vector (dirección y distancia) de regreso. Dicho vector será actualizado constantemente en la memoria de la hormiga y estará disponible en todo momento durante el forrajeo. De esta manera, la hormiga logra retornar al nido prácticamente en línea recta.

La integración de rutas tiene ventajas selectivas para las hormigas frente a las otras estrategias descritas. Aunque el seguimiento de ruta no resulta del todo im-

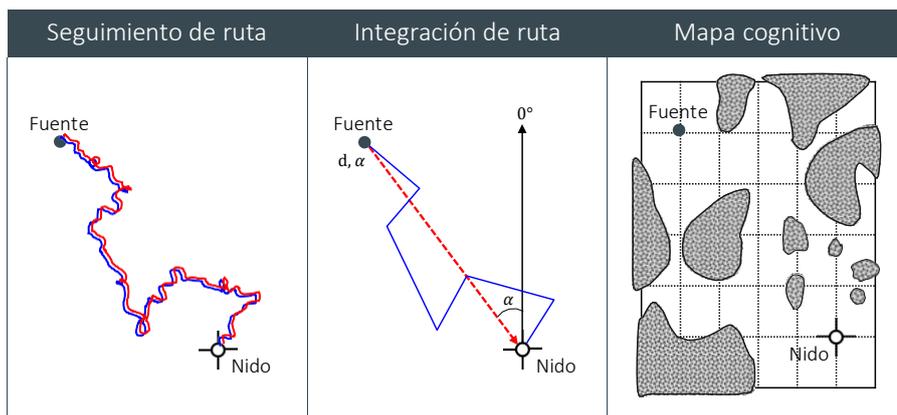


Figura 3. Tres estrategias de orientación empleadas por forrajeros centrales  
Fuente: Adaptado de Ronacher (2008)

plausible, ya que las hormigas podrían dejar rastros de feromonas para retornar tras el forrajeo, esta estrategia tiene dos desventajas manifiestas. Primero, que el viento o las temperaturas elevadas podrían alterar el rastro depositado, dejando a la hormiga con vacíos de información y seguramente extraviada. Segundo, que retornar por la misma ruta que se usó para la búsqueda incrementa considerablemente la distancia total recorrida y, por tanto, también el consumo energético por parte de la hormiga. En el caso de las hormigas del desierto (*Cataglyphis fortis*), una situación semejante aumentaría el riesgo de morir por agotamiento o desecadas por el sol. Por otro lado, el uso de mapas mentales requeriría almacenar grandes volúmenes de información sobre las relaciones espaciales de los objetos que se extienden por decenas, incluso cientos, de metros recorridos. Una hormiga, con su cerebro diminuto, tal vez no estaría en condiciones de lograr una proeza semejante, sobre todo si se considera que el cerebro es uno de los órganos que demandan mayor consumo energético. Adicionalmente, en ambientes desérticos, es infrecuente la presencia de marcas en el entorno. Ronacher (2008) plantea que la complejidad de los mecanismos neuronales que componen el sistema de navegación de las hormigas, probablemente se encuentre a medio camino entre el seguimiento de rutas y la elaboración de mapas mentales. Se ha demostrado que la integración de rutas, junto con otros mecanismos asociados, constituyen el sistema de navegación de las hormigas del desierto (Andel & Wehner, 2004; Müller & Wehner, 1988).

Diversos mecanismos neurofisiológicos están involucrados en el sistema de navegación que emplean las hormigas (Wehner, 2003): (i) Una brújula, basada en la polarización de la luz y en gradientes espectrales, que permite la estimación de los ángulos de rotación. (ii) Un odómetro para la estimación de las distancias recorridas (Ronacher, 2008). Este mecanismo se soporta en el flujo óptico y en las señales propioceptivas.

vas que autogenera la hormiga. Su lógica opera, en líneas generales, como un "contador" de pasos (Wolf, 2011). (iii) Un integrador de rutas, que calcula el vector de regreso con base en los ángulos y las distancias estimadas. (iv) Una memoria que le permite a la hormiga almacenar el vector actual de regreso e información sobre diversas marcas encontradas en el entorno (cuando están disponibles). El uso de marcas puede ser determinante, puesto que el integrador de rutas es propenso a la acumulación de errores. (v) Un

generador de búsqueda que es usado cuando la hormiga no encuentra el nido tras emplear los mecanismos previos. (vi) Un módulo global a través del cual la hormiga orquesta de forma flexible los demás mecanismos que componen el sistema de navegación (figura 4).

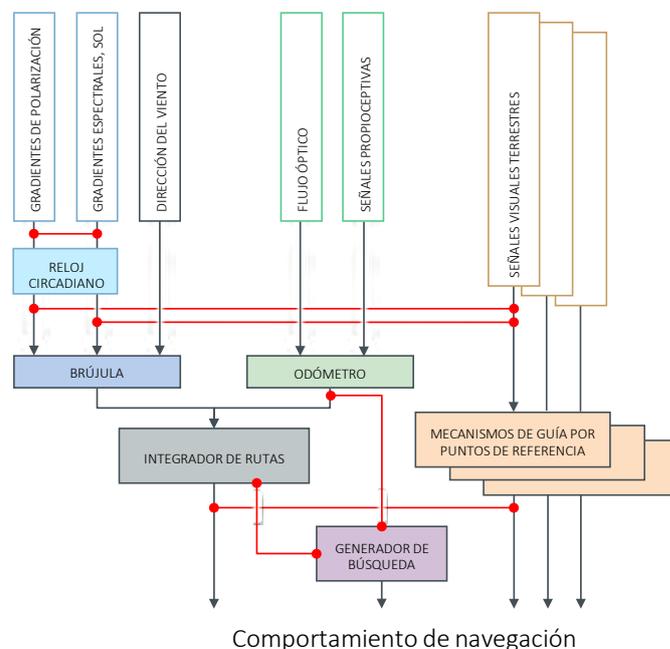


Figura 4. Sistema de navegación de las hormigas del desierto  
Fuente: Adaptado de Wehner (2009)

La navegación en las hormigas es un fenómeno cognitivo en el que complejas redes de procesamiento de información interactúan y se integran a través de múltiples escalas (Gordon, 2016; Almér, Dodig-Crnkovic & von Haugwitz, 2015). Tales redes han evolucionado, mediante selección natural, con el fin de satisfacer las condiciones ecológicas de las diversas especies. Para una hormiga, integrar rutas o hacer uso de puntos de referencia que la orienten en su regreso al nido requiere, entre otras cosas, habilidades de percepción, almacenamiento, recuperación, estimación, transformación y síntesis de información. Comprender los mecanismos

de procesamiento de información que emplean las hormigas para resolver sus necesidades de navegación es, de hecho, un problema computacional de interés científico, particularmente desde el punto de vista de la neurociencia y la ciencia cognitiva (Wehner, 2003).

### Integración de rutas en las hormigas del desierto

La integración de rutas es, entonces, el principal mecanismo de navegación que emplean las hormigas del desierto de la especie *Cataglyphis fortis* (figura 5) (Ronacher, 2008). La razón radica en que, en ambientes desérticos, como los salares del norte de África, es infrecuente la existencia de puntos de referencia que las hormigas puedan memorizar y seguir (Andel & Wehner, 2004). Puesto que en el desierto las fuentes de alimento son escasas, las excursiones de forrajeo se pueden extender por cientos de metros. Para no deshidratarse, las hormigas deben desplazarse tan rápido como les resulte posible y retornar al nido por la vía más corta disponible. Las hormigas del desierto se desplazan a la sorprendente velocidad promedio de 50 cm/seg (Wehner & Whener, 1990).



Figura 5. Hormiga del desierto, *Cataglyphis fortis* (derecha) y visión frontal de su cerebro (izquierda)  
Fuente: tomado de Wittlinger, Wolf y Whener (2007) y Stieb (2010), respectivamente

El retorno al nido, que no es menos sorprendente, se logra empleando la integración de rutas. La figura 6 muestra el mapeo experimental de la trayectoria saliente de una hormiga del desierto cuando busca el alimento (línea roja). Este recorrido cubre 354,5 metros. La trayectoria entrante (línea verde), por su parte,

mide solo 113,2 metros (Müller & Wehner, 1988).

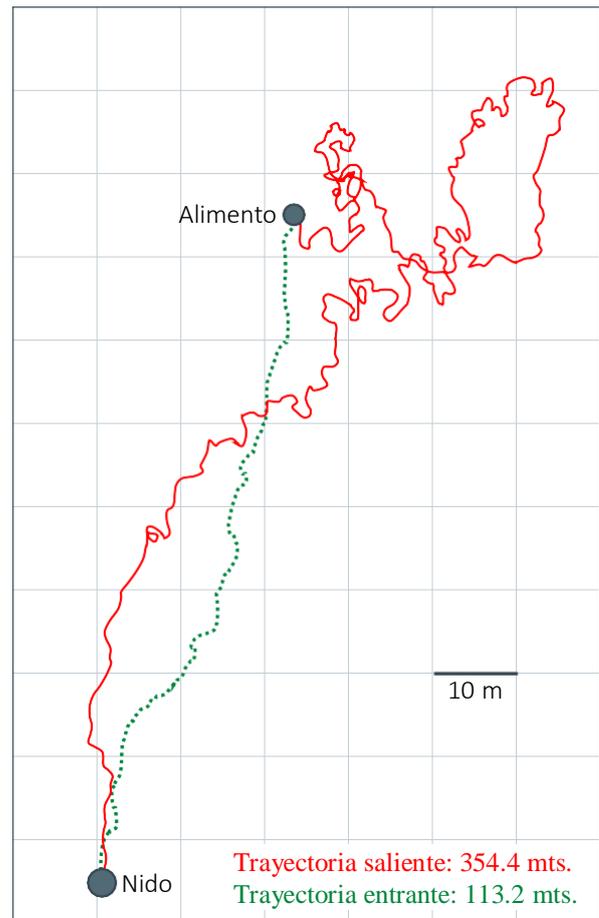


Figura 6. Viaje de forrajeo de una hormiga de la especie *Cataglyphis fortis*  
Fuente: Adaptado de Heinze, Narendra y Cheung (2018)

Como se indicó en la sección anterior, la integración de rutas significa que el animal es capaz de computar continuamente su posición actual con respecto a un punto de origen a medida que se desplaza. Esa información es empleada, posteriormente, para regresar a dicho punto de forma directa. Las hormigas del desierto continuamente memorizan los componentes lineales y angulares de su movimiento e integran esa información, de forma autogenerada, en un vector de regreso (Wehner & Srinivasan, 2003). Cada hormiga está equipada con una brújula que determina la orientación y un odómetro que mide las distancias (Hartmann & Wehner, 1995).

Para determinar la dirección, las hormigas usan los patrones de polarización del cielo y, en menor medida, la posición del sol, gradientes espectrales y la dirección del viento (Ronacher, 2008). Las aproximaciones experimentales soportan la idea de que las hormigas usan una plantilla interna simplificada que abarca los patrones de polarización que ocurren en distintos momentos del día y del año. Este mecanismo les permite a las hormigas determinar su dirección de desplazamiento con bastante precisión (Horváth & Varjú, 2004). Adicio-

nalmente, cuando una hormiga viaja a sitios comunes varias veces durante el día, las variaciones en los patrones de polarización de la luz son compensadas con información acerca del curso del sol, la cual es obtenida gracias a su reloj circadiano interno (Ronacher, 2008). Por otro lado, la manera como las hormigas miden las distancias recorridas ha permanecido elusiva por mucho tiempo. Experimentos recientes demuestran que la hipótesis más plausible es el uso de un mecanismo de conteo de pasos (Ronacher, 2008; Wittlinger, Whener & Wolf, 2006; Thiélin-Bescond & Beugnon, 2005). Ambos mecanismos son altamente complejos y los detalles neurofisiológicos continúan bajo estudio (Collete, 2019; Heinze et al., 2018).

Se han logrado avances importantes con respecto a la manera como la información posicional es integrada para producir el vector de regreso al nido. Algunas estrategias también se han descartado. Por ejemplo, se ha logrado demostrar que las hormigas del desierto no hacen uso de información cartográfica acerca de la ubicación del nido ni de la fuente de alimento (Andel & Wehner, 2004). La implicación de este hecho es que las hormigas no adquieren ni usan mapas cognitivos como mecanismo de navegación (Wehner, Boyer, Loertscher, Sommer & Menzi, 2006). También se sabe que no calculan la suma de los vectores recorridos. Por el contrario, los hallazgos experimentales sugieren que el mecanismo está soportado en una estrategia aproximada, sujeta a errores. Como se anotó anteriormente, las hormigas pueden emplear marcas en el entorno (cuando están disponibles) y en casos extremos recurrir a búsquedas sistemáticas.

Müller y Wehner (1988) propusieron un modelo que es consistente con la información experimental disponible. El modelo parte del supuesto de que las hormigas pueden medir de manera iterativa la media aritmética de todas las desviaciones efectuadas durante un pequeño trayecto, ponderando el resultado con la longitud total recorrida. Esta hipótesis es formulada como sigue:

$$\varphi_{n+1} = \frac{l_n \varphi_n + \varphi_n + \delta}{l_n + 1} = \frac{\varphi_n (l_n + 1) + \delta}{l_n + 1}$$

$$l_{n+1} = l_n + 1 - \frac{\delta}{90^\circ}$$

$\varphi_n$  denota la orientación media tras  $n$  pasos.  $l_n$  denota la distancia cubierta por la hormiga tras  $n$  pasos, y  $\delta$  la diferencia angular entre  $\varphi_n$  y la dirección de la hormiga en el paso  $n + 1$ .  $\varphi_{n+1}$  y  $l_{n+1}$  denotan la dirección y la distancia tras  $n + 1$  pasos, respectivamente. La inter-

pretación geométrica de las ecuaciones es presentada en la figura 7.

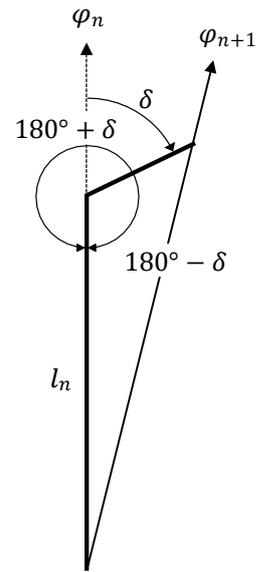


Figura 7. Interpretación gráfica del modelo de integración de rutas Fuente: Müller y Wehner, 1988

La integración sucede tras cada segmento recorrido y el resultado es actualizado en la memoria de la hormiga. El mecanismo es innato y no requiere, en principio, aprendizaje. Sin embargo, debido a factores de entorno, fisiológicos y cognitivos, la integración es propensa a errores. Es necesario, entonces, que la hormiga memorice (aprenda), por lo menos, algunos atributos de la entrada del nido y sus alrededores como complemento de la información que provee el integrador (Ronacher, 2008). En la siguiente sección, se propone un modelo del mecanismo de integración de rutas propuesto por Müller & Wehner (1988) con el fin de analizar su expresividad computacional. Las dinámicas de aprendizaje, por su parte, no son consideradas.

### Computación interactiva secuencial para volver a casa

En esta sección se presenta la simulación de una hormiga artificial que sigue el modelo de integración de rutas propuesto por Müller y Wehner (1988) (en adelante referido como *modelo Müller-Wehner*). Asimismo, se utiliza una Máquina de Turing Persistente (MTP) como modelo del sistema de integración de rutas (SIR) que emplea la hormiga *Cataglyphis fortis*. Aunque la MTP no captura todos los mecanismos presentados en la figura 4, el modelo logra capturar aspectos fundamentales sobre la naturaleza y la expresividad computacional del SIR.

El modelo Müller-Wehner fue simulado usando la plataforma de simulación basada en agentes NetLogo (Wilensky, 1999). Se dispuso un entorno caracteriza-

do por una cuadrícula de 160 x 160. En la parte inferior derecha se ubicó el nido (de color morado) y en la parte superior izquierda una presa (de color azul). Se programó una hormiga que parte del nido siguiendo una caminata semialeatoria similar a la que realizan las hormigas reales. Antes de cada paso, durante el proceso de forrajeo, se le permitió a la hormiga decidir su orientación de manera aleatoria dentro de un rango de (-35°, 35°). El desplazamiento que logra la hormiga dentro de cada lapso de la simulación (denominado *tick*) equivale a un metro de desplazamiento. Para cada *tick*, la hormiga calcula su desplazamiento y su ángulo de giro. Esos dos datos son suministrados a un procedimiento denominado *integrador\_de\_rutas*, que aplica el modelo Müller-Wehner. Al encontrar el alimento, el *integrador\_de\_rutas* informa a la hormiga el vector de regreso. La figura 8 muestra tres ejemplos del funcionamiento del modelo. En la imagen de la izquierda, la ruta de forrajeo (línea roja) equivale a 624 metros y la ruta de retorno al nido (línea verde) 189. En la imagen del centro, la ruta de forrajeo equivale a 337 metros, mientras que la ruta de retorno al nido solo 172. Finalmente, la imagen de la derecha muestra una ruta de forrajeo de 437 metros y de 164 de retorno. Los resultados son consistentes con las características del experimento de la figura 6.

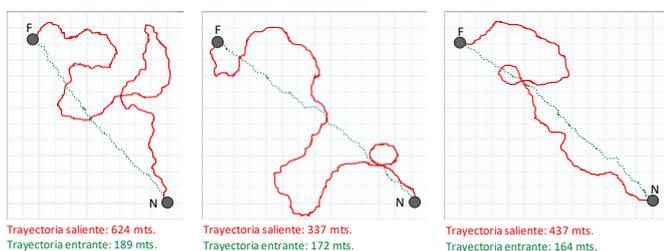


Figura 8. Simulación de la integración de rutas  
Fuente: simulaciones realizadas por el autor utilizando NetLogo

El hecho de que las hormigas no regresen de manera estrictamente recta (como se evidencia en la figura 6), supone la existencia de un *mecanismo de compensación* que, hasta donde sabemos, no ha sido estudiado. Con el único fin de hacer más realista la apariencia visual del modelo que se presenta en la figura 8, se implementó un mecanismo de "wigggle compensado". Dicho mecanismo opera toda vez que la hormiga se orienta hacia el nido para ejecutar su vector de regreso. La figura 9 muestra el diagrama de flujo de la hormiga artificial durante el proceso de búsqueda de alimento y retorno al nido.

Ahora bien, se sabe que el SIR opera sin referencia a guías externas y que nunca deja de funcionar (Wehner & Srinivasan, 2003). De acuerdo con esto y con la simulación realizada, el SIR puede ser entendido en sí mismo como un agente autónomo y dinámico, dotado

con una memoria rudimentaria y con la capacidad de interactuar con su entorno (la hormiga). Se sabe también que los subsistemas que miden la rotación y la traslación son independientes y trabajan en paralelo. Para efectos prácticos, se supone que tanto el ángulo de rotación, como la distancia recorrida, son presentados al SIR en la forma de un paquete único de información. Dicho paquete será denominado pA-D (paquete ángulo-distancia).

El SIR, entonces, percibe el pA-D y lo procesa con base en el modelo Müller-Wehner. Como resultado, se obtiene el vector de regreso al nido  $\vec{v}_i$ . Dicho vector es almacenado en la memoria del SIR. Si existe un vector previo almacenado  $\vec{v}_{i-1}$  es reemplazado por el nuevo vector. Este proceso es repetido de forma indefinida hasta que la hormiga encuentra el alimento. Cuando esto sucede, el SIR informa a la hormiga el contenido de su memoria y la hormiga emplea esa información para retornar al nido. Una vez que la hormiga arriba al nido, la memoria del integrador se reinicia en cero. Mientras la hormiga permanezca en el nido, la memoria no se actualizará y, sin embargo, el SIR permanecerá activo. El proceso global sucede cada vez que la hormiga sale a forrajear.

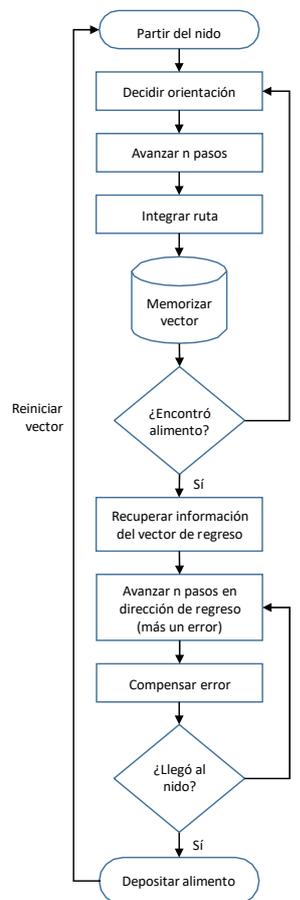


Figura 9. Diagrama de flujo de la hormiga artificial que implementa el modelo Müller-Wehner  
Fuente: elaboración propia

La información del vector de regreso almacenada en memoria ( $\vec{v}_{i-1}$ ) es mantenida (memorizada) de un macropaso de computación al siguiente y afectará el cómputo de los vectores subsecuentes. De esta manera, el SIR es un sistema persistente. Asimismo, los cálculos efectuados por el SIR dependen estrictamente de su interacción con el entorno. Puesto que siempre está activo, todo procesamiento de información dentro del SIR ocurre *durante la computación*. Por la misma razón, el SIR no es un sistema terminante, salvo que sufra un daño o que la hormiga perezca.

El esquema conceptual del SIR que se acaba de introducir es ahora reformulado en el lenguaje de las MTP. La cinta de trabajo es usada para almacenar el vector de regreso vigente y calcular las nuevas versiones de este. Los pasos computacionales son definidos por una función de mapeo Turing-computable de la forma:

(token de entrada, memoria actual)  $\rightarrow$  (token de salida, memoria nueva)

De acuerdo con esto, el SIR *está* definido por:

$$\begin{aligned} f_1(\text{calcular } \vec{v}_i, \vec{v}_{i-1}) &= (ok, \vec{v}_{i-1}); \\ f_1(\text{almacenar } \vec{v}_i, \vec{v}_{i-1}) &= (ok, \vec{v}_i); \\ f_1(\text{reportar}, v_i) &= (v_i, v_i); \\ f_1(\text{reiniciar}, v_i) &= (\text{hecho}, v_0). \end{aligned}$$

Una MTP recibe un flujo de entrada que es generado dinámicamente por el entorno. En este caso, es la hormiga quien genera el flujo de entrada para el SIR. El siguiente flujo de entrada resalta la forma como la hormiga interactúa con su entorno durante el forrajeo. De esta manera, permite ilustrar que el procesamiento de información que realiza el SIR es dependiente de la historia:

(calcular A, almacenar A, calcular B, almacenar B, ..., reportar, reiniciar, calcular A', almacenar A', calcular B', almacenar B', ..., reportar, reiniciar, ...)

La hormiga parte del nido (el cero relativo) y se desplaza una longitud  $l_1$  con ángulo  $\delta$  respecto al origen. Esa información (el pA-D) es obtenida por la hormiga y usada por el SIR para calcular el vector de regreso (*calcular A*). Una vez calculada, la información del vector de regreso es almacenada por el integrador de rutas (*almacenar A*). Luego, la hormiga se desplaza otro segmento de longitud  $l_2$  con un ángulo  $\delta'$ . Los nuevos datos percibidos, junto con la información almacenada en el integrador de rutas (*A*), son usados para calcular un nuevo vector de regreso. Los puntos suspensivos indican que el proceso se repite indefinidamente hasta que el alimento es encontrado por la hormiga. Cuando esto sucede, el SIR informa a la hormiga el vector de regreso al nido

(*reportar*). El SIR reporta, en todos los casos, el último vector almacenado hasta ese momento, el cual contiene la distancia y el ángulo que la hormiga debe recorrer para llegar al nido. Ya en el nido, la hormiga reinicia en cero el SIR. El *stream* de salida correspondiente, de acuerdo con la función de mapeo, es de la forma:

(ok, ok, ok, ok, ..., T, hecho, ok, ok, ok, ok, ..., T', hecho, ...)

Nótese que ante el mismo *token* de entrada (*reportar*) presentado al SIR en momentos distintos, se generan dos vectores de salida distintos ( $T$  y  $T'$ ). Esto obedece a dos razones fundamentales que se desprenden de la semántica de las MTP. Por un lado, la ruta que sigue la hormiga depende estrictamente de las microdecisiones que son tomadas, en tiempo real, a medida que interactúa con su entorno. Por otra parte, como cada trayecto recorrido en una dirección específica contribuye a la integración del vector de regreso, el sistema se hace, en efecto, dependiente de la historia. Aunque la MTP cuenta con la misma sintaxis que la MT estándar, la modificación de la semántica (persistencia e interacción) le permite a la hormiga resolver el problema de la integración de rutas de manera no funcional (dicho en términos matemáticos<sup>2</sup>) y no algorítmica (puesto en términos computacionales). Este modelo deja claro que la hormiga no puede precomputar su vector de regreso, pues esto implicaría que, además, precomputara hasta el último detalle de su ruta de forrajeo.

## Computación interactiva distribuida para optimizar la recolección de alimento

La interacción con el entorno le permite a una hormiga recolectar información suficientemente precisa para regresar al nido. En realidad, los mecanismos involucrados en la integración de rutas operan en el nivel neurofisiológico de forma autoorganizada y emergente. Los detalles de su operación se desconocen aún. Por ejemplo, para capturar la polarización de la luz del cielo, las hormigas usan una pequeña parte de su sistema visual, la cual recibe información de una fracción muy pequeña de los fotorreceptores del ojo que son sensibles a la luz polarizada. La manera cómo esa información fluye en los distintos niveles sinápticos y es representada e interpretada por la hormiga sigue siendo objeto de estudio. Lo mismo sucede con las redes neuronales que "instancian" el integrador de rutas. El modelo presentado en la sección anterior no captura esos aspectos. No

2 Una función matemática establece una correspondencia entre una colección de valores de entrada posibles y una colección de valores de salida, tal que cada entrada es asignada a una única salida. El proceso de determinar el valor de salida que una función asigna a una entrada específica se llama computar la función.

obstante, logra capturar de forma sensata el fenómeno en una escala global. Gracias a ello, es posible mostrar que el SIR no opera de forma algorítmica ni se ajusta a la estructura de una función matemática. Pero ahora queda la pregunta ¿es más expresiva la interacción distribuida que la secuencial?

No existe, de momento, un modelo de CI u otro enfoque formal que permita responder a esta pregunta. Carecemos de modelos rigurosos que caractericen los problemas computacionales que puede resolver un cerebro líquido, como las hormigas, y los que no (Solé et al., 2019). En esta sección, sin embargo, se hace una aproximación intuitiva a la respuesta. Para ello se considera el problema de la recolección de alimentos en colonias de hormigas. Se utiliza una variación del modelo de simulación basada en agentes propuesto por Wilensky (1997). Puesto que el modelo no está calibrado empíricamente, su valor es heurístico. Se trata de una población de hormigas dispuesta en un entorno de 70 x 70 unidades. En el centro se dispone un nido (de color morado) y en la periferia tres fuentes de alimento (con diferentes tonos de azul), ubicadas a distintas distancias. La simulación opera de acuerdo con la descripción del proceso de formación de senderos realizada en la sección "Computación individual y colectiva en las hormigas".

Las hormigas son modeladas como agentes autónomos con la capacidad de forrajear en búsqueda de alimento, recolectar pequeñas porciones de este, retornar al nido mediante integración de rutas cuando el alimento es recolectado, depositar feromonas en el camino de regreso al nido y percibir feromonas del entorno. Se definen como parámetros de la simulación el número de hormigas (población) y las tasas de difusión y evaporación de las feromonas. La figura 10 muestra una captura de pantalla de la estructura y el funcionamiento del modelo. Las feromonas son representadas dentro del rango de colores que va de blanco (más intensa) a verde (menos intensa).

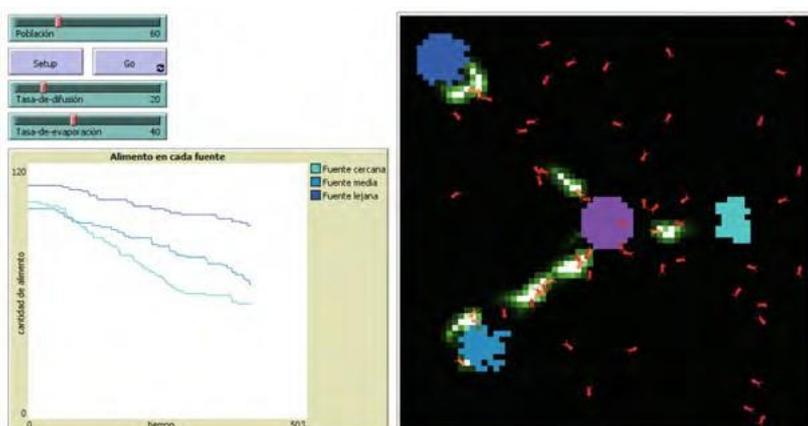


Figura 10. Captura de pantalla del modelo *Ants* tras 325 *ticks* de simulación  
Fuente: Elaboración propia empleando NetLogo

Se definen como criterios de expresividad computacional la eficiencia en la recolección del alimento (velocidad de respuesta), la robustez del proceso de recolección (tolerancia al ruido y a la aleatoriedad) y la generalidad de la respuesta. Estos criterios están conformes a la propuesta de MacLennan (2004). Del mismo modo, estos criterios se enmarcan en el argumento de Eberbach et al. (2004), según el cual la capacidad de un sistema interactivo para resolver problemas es directamente proporcional a su expresividad computacional.

Con estos elementos en mente se analizaron tres casos. En el caso **a**, se simula una población de 125 hormigas que no disponen de un sistema de integración de rutas (SIR) ni de mecanismos de interacción a través de feromonas. Cada hormiga busca alimento de manera aleatoria y del mismo modo busca el nido para depositar el alimento obtenido. En el caso **b**, una población igual de hormigas es liberada, pero esta vez cada una cuenta con un SIR. Los mecanismos de segregación y seguimiento de feromonas permanecen deshabilitados. Es decir, en este caso cada agente representa una MTP que interactúa con las fuentes de alimento, pero no con otros agentes. En el caso **c**, las hormigas que conforman la población cuentan con el SIR y con los mecanismos de segregación y seguimiento de feromonas activos. Los parámetros de tasa de difusión y de tasa de evaporación fueron calibrados de acuerdo con el modelo original de Wilensky para posibilitar la formación de senderos.

Las condiciones de entorno son exactamente las mismas para los tres casos. Esto significa que la disposición del nido y de las fuentes de alimento siempre es igual. Adicionalmente, se impusieron condiciones de frontera que impiden que las hormigas vayan más allá de la cuadrícula de 70 x 70 que delimita el entorno. En la figura 11 se analizan los resultados de las tres situaciones.

La dinámica de explotación en los casos **a** y **b** fue muy similar. En ambas situaciones las fuentes fueron explotadas de forma simultánea con diferencias que obedecen a la distancia de las distintas fuentes al nido. Sin embargo, los tiempos de explotación que emplearon las colonias en cada caso variaron considerablemente.

La explotación de la fuente más cercana tomó 1735 *ticks* de simulación para el caso **a** y 1102 para el caso **b**. La explotación de la fuente intermedia 1972 y 1071 *ticks*, respectivamente. La fuente más lejana fue explotada en 2769 y 1861 *ticks* para cada escenario. El caso **b**, en el que se implementa el SIR es, de lejos, más eficiente que el caso **a**. En el caso **c** los tiempos de explotación se redujeron de una manera importante. Al hormiguero del tercer escenario le tomó 221, 576 y 1076 *ticks*, respectivamente, explotar las respectivas fuentes. La tabla 1 resume los datos obtenidos. Los gráficos de la figura 11 son presentados de manera ilustrativa. Sin embargo, se eligieron porque no distan considerablemente del comportamiento promedio tras varias ejecuciones.

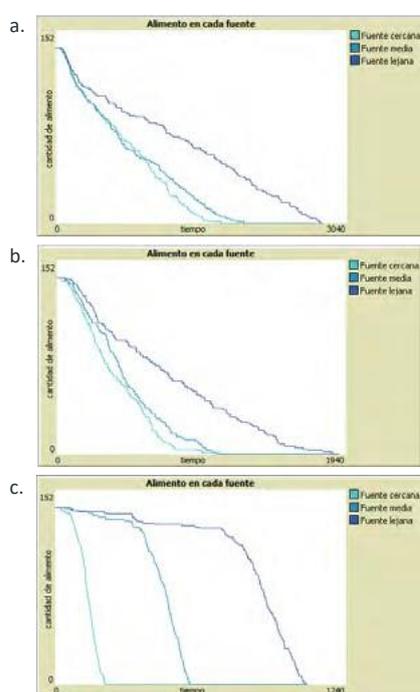


Figura 11. Dinámica de recolección de alimentos en una colonia de hormigas bajo distintos criterios de interacción. En **a** no se admite integración de rutas ni comunicación mediante feromonas. En **b** se admite integración de rutas, pero no comunicación mediante feromonas. En **c** tanto la integración de rutas, como la comunicación mediante feromonas son admitidas.

Fuente: Elaboración propia empleando NetLogo

Tabla 1. Tiempos de explotación (en *ticks*) de tres fuentes de alimento bajo distintos parámetros de interacción

	Tiempo de explotación de las fuentes de alimento		
	Más cercana	Distancia media	Más lejana
Caso a. (Interacción limitada)	1753	1972	2769
Caso b. (Interacción secuencial)	1102	1071	1861
Caso c. (Interacción distribuida)	221	576	1076

Fuente: Elaboración propia

Esta sencilla prueba muestra que la velocidad con la que se explotan las tres fuentes es, en general, más eficiente en el caso de la interacción secuencial que en el caso de interacción limitada. Análogamente, que la interacción distribuida es más eficiente que la secuencial. Hay dos hechos adicionales, altamente relevantes, que vale la pena mencionar. En el caso **c**, cuando la interacción distribuida fue incluida, el hormiguero concentró sus esfuerzos de recolección: una fuente a la vez. Adicionalmente, la priorizó: primero la fuente cercana, luego la que se encuentra a una distancia media y finalmente la más lejana (figura 12). Este comportamiento es emergente e imposible de lograr, incluso, con un sistema de computación interactiva secuencial paralelo, como el explorado en el caso **b**.

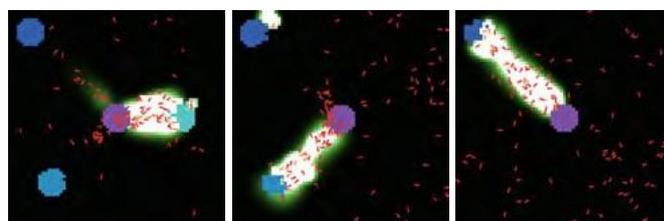


Figura 12. Formación emergente de senderos y recolección óptima de recursos en un hormiguero simulado

Fuente: Elaboración propia empleando NetLogo

Los resultados obtenidos muestran cómo un grupo de hormigas que se comunica indirectamente mediante feromonas puede construir senderos de forma autoorganizada, decidir colectivamente qué fuente de alimento recolectar y explotar de manera óptima los recursos en su entorno. En el mundo real, estas estrategias tienen, sin lugar a duda, ventajas evolutivas para el hormiguero frente a los escenarios evaluados en los casos **a** y **b**. La dinámica global exhibida por el hormiguero en el escenario **c** es lograda a través de un balance entre la exploración aleatoria que realiza cada hormiga y la explotación sistemática que sucede en la macroescala, cuando la colonia logra un refuerzo positivo de los rastros de feromonas.

En consecuencia, el procesamiento interactivo y distribuido de información es más eficiente en cuanto a la velocidad con la que se explotan los recursos globalmente, pero también con respecto a la estrategia de priorización de la explotación. Asimismo, la aleatoriedad y el ruido que esta puede generar en el proceso son aprovechadas como recursos computacionales para balancear la exploración del entorno y la explotación de las fuentes de alimento halladas. Esto hace al sistema altamente flexible y robusto. Si la configuración del entorno varía, el hormiguero se adapta a las nuevas condiciones y sigue respondiendo de manera eficiente. En conclusión, se puede argumentar que la interacción distribuida es más expresiva que la secuencial. No es

extraño, entonces, que sea esta la forma de procesamiento de información que predomina en los sistemas vivos.

La computación interactiva le abre paso a un paradigma de computación sin precedentes que promete acelerar la comprensión y la explicación de la vida no desde lo que es, sino desde lo que hace para vivir (que es computar). Por ahora, más esfuerzos de investigación son necesarios. La apreciación de Amos, Hodgson y Gibbons (2007) deja abierta la discusión:

El desarrollo de patrones funcionales extendidos espacialmente como resultado de la interacción local entre individuos y entre estos y su entorno inmediato es una forma de computación extremadamente común y útil en la naturaleza, pero aún es poco explorada, pobremente entendida y subexplotada en las ciencias de la computación (pág. 200)

## Conclusiones

En los sistemas vivos, la computación emerge de la interacción orquestada entre múltiples componentes descentralizados. Esta computación se caracteriza por ser robusta, autoorganizada, asíncrona, adaptativa, tolerante a fallos y capaz de evolucionar (Gómez-Cruz & Niño, 2020). Pensar la vida en términos computacionales implica, por lo tanto, pensar más allá del marco de relevancia de la computación algorítmica que captura la MT. La CI propone un nuevo marco de relevancia que provee un escenario apropiado para pensar y explorar la riqueza y la expresividad computacional de la vida. Este escenario forma parte de lo que se ha denominado hipercomputación biológica (Maldonado & Gómez-Cruz, 2015)

En este artículo se demostró que la solución al problema de la integración de rutas en las hormigas del desierto no se puede llevar a cabo en términos algorítmicos. Esto significa que una hormiga no puede precomputar su vector de regreso, sino que tiene que generarlo cada vez a medida que interactúa con el entorno. La lógica general de la integración de rutas fue capturada con una máquina de Turing persistente, un modelo que es más expresivo que la máquina de Turing estándar. El mecanismo de integración de rutas estudiado, que comparte algunas características computacionales esenciales con otros fenómenos biológicos, es un ejemplo de computación interactiva secuencial.

Adicionalmente, desde un punto de vista conceptual, se mostró que el procesamiento de información colectivo que realiza una colonia para obtener el alimento de una manera eficiente excede la expresividad computacional

de las hormigas consideradas individualmente. De esta manera, se aportó evidencia a la conjetura de que la computación interactiva distribuida es más expresiva que la computación interactiva secuencial. No es casual que toda la vida sobre la Tierra, desde los organismos unicelulares hasta los insectos sociales, se caracterice por ser interactiva y colectiva.

En rigor, la MT y todos los modelos de computación equivalentes representan solamente un caso particular dentro de los modelos de computación posibles. Si bien la vida respeta los límites físicos de la computación, también es cierto que ella establece un escenario abierto, mediado por la evolución, para la síntesis y la creatividad de los procesos computacionales naturales. La creatividad, la diversidad y la expresividad computacional de la vida no se pueden reducir, en absoluto, a la aplicación mecánica de un procedimiento algorítmico. La computación interactiva en los sistemas vivos, como ha sido introducida en este artículo, busca capturar la riqueza del procesamiento de información que lleva a cabo la vida, sin reducirla a un mecanismo recursivo formal.

## Agradecimientos

Le agradezco al Dr. Luis Fernando Niño Vásquez, director del Laboratorio de Sistemas Inteligentes de la Facultad de Ingeniería de la Universidad Nacional de Colombia, por las discusiones que sostuvimos acerca de los aspectos fundamentales de esta investigación y por sus observaciones sobre el manuscrito.

## Referencias

- Almér, A., Dodig-Crnkovic, G. & von Haugwitz, R. (2015). Collective cognition and distributed information processing from bacteria to humans. In: *AIBS Convention 2015* (code 112552). Canterbury: University of Kent.
- Amos, M., Hodgson, D. A. & Gibbons, A. (2007). Bacterial self-organisation and computation. *International Journal of Unconventional Computing*, 3(3), 199-210.
- Andel, D. & Wehner, R. (2004). Path integration in desert ants, *Cataglyphis*: how to make a homing ant run away from home. *Proceedings of the Royal Society B*, 271(1547), 1485-1489.
- Barrett, L. (2011). *Beyond the Brain: How Body and Environment Shape Animal and Human Minds*. Princeton: Princeton University Press.
- Bonabeau, E. (1998). Social insect colonies as complex adaptive systems. *Ecosystems*, 1(5), 437-443.
- Bonabeau, E., Dorigo, M. & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford University Press.
- Bonabeau, E., Theraulaz, G., Deneubourg, J. L., Aron, S. & Camazine, S. (1997). Self-organization in social insects. *Trends in*

- Ecology & Evolution*, 12(5), 188-193.
- Brenner, S. (2012). Life's code script. *Nature*, 482(7386), 461.
- Camazine, S., Deneubourg, J. L., Franks, N. R., Sneyd, J., Theraulaz, G. & Bonabeau, E. (2001). *Self-Organization in Biological Systems*. Princeton: Princeton University Press.
- Cleland, C. E. (2006). The Church-Turing thesis. A last vestige of a failed mathematical program. In A. Olszewski, J. Woleński & R. Janusz (Eds.), *Church's Thesis After 70 Years* (pp. 119-146). Frankfurt: Ontos.
- Collete, T. (2019). Path integration: how details of the honeybee waggle dance and the foraging strategies of desert ants might help in understanding its mechanisms. *Journal of Experimental Biology*, 222(11), jeb205187.
- Da Costa, N. & Doria, F. A. (2013). Metamathematical limits to computation. In K. Nakamatsu & L. C. Jain (Eds.), *The Handbook on Reasoning-Based Intelligent Systems* (pp. 119-141). New Jersey: World Scientific.
- Dodig-Crnkovic, G. (2011). Dynamics of information as natural computation. *Information*, 2(3), 460-477.
- Dodig-Crnkovic, G. (2011a). Significance of models of computation, from Turing model to natural computation. *Minds and Machines*, 21(2), 301-322.
- Dodig-Crnkovic, G. (2010). Biological information and natural computation. In J. Vallverdú (Ed.), *Thinking Machines and the Philosophy of Computer Science: Concepts and Principles* (pp. 36-52). Hershey, PA: IGI Global.
- Dornhaus, A. & Franks, N. R. (2008). Individual and collective cognition in ants and other insects (Hymenoptera: Formicidae). *Myrmecological News*, 11, 215-226.
- Dorigo, M. & Stützle, T. (2004). *Ant Colony Optimization*. Cambridge, MA: MIT Press.
- Eberbach, E., Goldin, D. & Wegner, P. (2004). Turing's ideas and models of computation. In: C. Teuscher (Ed.), *Alan Turing: Life and Legacy of a Great Thinker* (pp. 159-194). Berlin: Springer.
- Feinerman, O. & Korman, A. (2017). Individual versus collective cognition in social insects. *Journal of Experimental Biology*, 220(1), 73-82.
- Forbes, N. (2004). *Imitation of Life: How Biology Is Inspiring Computing*. Cambridge, MA: MIT Press.
- Franks, N. R. (1989). Army ants: A collective intelligence. *American Scientist*, 77(2), 138-145.
- Goldin, D., Smolka, S. A., Attie, P. & Sonderegger, E. (2004). Turing machines, transition systems, and interaction. *Information & Computation Journal*, 194(2), 101-128.
- Goldin, D. & Wegner, P. (2008). The interactive nature of computing: Refuting the strong Church-Turing thesis. *Minds & Machines*, 18(1), 17-38.
- Gómez-Cruz, N. A. (2020). Computación biológica: el estudio de la naturaleza computacional de los sistemas vivos. En C. E. Maldonado (Ed.), *Biología Teórica, Explicaciones y Complejidad* (capítulo 7). Bogotá, Universidad del Bosque.
- Gómez-Cruz, N. A. (2018). Simulación basada en agentes: una metodología para el estudio de sistemas complejos. In M. L. Eschenhagen, G. Velez, Guerrero, G. & C. E. Maldonado (Eds.), *Construcción de problemas de investigación: diálogos entre el interior y el exterior* (páginas 230-268). Medellín: Universidad de Antioquia.
- Gómez-Cruz, N. A. (2013). *Vida Artificial: Ciencia e Ingeniería de Sistemas Complejos*. Bogotá: Universidad del Rosario.
- Gómez-Cruz, N. A. & Niño, L. F. (2020). Computación biológica: el estudio de la naturaleza computacional de los sistemas vivos. En C. E. Maldonado (Ed.), *Biología Teórica, Explicaciones y Complejidad* (capítulo 7). Bogotá, Universidad del Bosque.
- Gordon, D. (2016a). Collective wisdom of ants. *Scientific American*, 314(2), 44-47.
- Gordon, D. (2016b). The evolution of the algorithms for collective behavior. *Cell Systems*, 3(6), 514-520.
- Gordon, D. (2010). *Ant Encounters: Interaction Networks and Colony Behavior*. Princeton: Princeton University Press.
- Grimm, V., Revilla, E., Berger, U., Jeltsch, F., Mooij, W. M., Railsback, S. F. et al. (2005). Pattern-oriented modeling of agent-based complex systems: Lessons from ecology. *Science*, 310(5750), 987-991.
- Hartmann, G. & Wehner, R. (1995). The ant's path integration system: a neural architecture. *Biological Cybernetics*, 73(6), 483-497.
- Heinze, S., Narendra, A. & Cheung, A. (2018). Principles of insect path integration. *Current Biology*, 28(17), R1043-R1058.
- Hewitt, C. (2013). What is computation? Actor model vs. Turing's model. In H. Zenil (Ed.), *A Computable Universe. Understanding Computation and Exploring Nature as Computation* (pp. 159-186). Singapore: World Scientific.
- Hölldobler, B. & Wilson, E. O. (2014). *El Superorganismo. Belleza y Elegancia de las Asombrosas Sociedades de Insectos*. Buenos Aires: Katz Editores.
- Hölldobler, B. & Wilson, E. O. (1990). *The Ants*. Berlin: Springer.
- Horváth, G. & Varjú, D. (2004). *Polarized Light in Animal Vision: Polarization Patterns in Nature*. Berlin: Springer.
- Kari, L. & Rozenberg, G. (2008). The many facets of natural computing. *Communications of the ACM*, 51(10), 72-83.
- MacLennan, B. J. (2004). Natural computation and non-Turing models of computation. *Theoretical Computer Science*, 317(1-3), 115-145.
- Maldonado, C. E. & Gómez-Cruz, N. A. (2015). Biological hypercomputation: A new research problem in complexity theory. *Complexity*, 20(4), 8-18.
- Mitchell, M. (2009). *Complexity: A Guided Tour*. Oxford: Oxford University Press.
- Mitchell, M. (2012). Biological computation. *The Computer Journal*, 55(7), 852-855.
- Moussaid, M., Garnier, S., Theraulaz, G. & Helbing, D. (2009). Collective information processing and pattern formation in swarms, flocks y crowds. *Topics in Cognitive Science*, 1(3), 469-497.
- Müller, M. & Wehner, R. (1988). Path integration in desert ants, *Cataglyphis fortis*. *Proceedings of the National Academy of Sciences*, 85(14), 5287-5290.
- National Geographic (2011). Hermandad de tejedoras. Retrieved

- from [http://www.nationalgeographic.com/es/mundo-ng/grandes-reportajes/hermandad-de-tejedoras\\_4190/1](http://www.nationalgeographic.com/es/mundo-ng/grandes-reportajes/hermandad-de-tejedoras_4190/1)
- Newell, A. & Simon, H. (1976). Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, 19(3), 113-126.
- Ronacher, B. (2008). Path integration as the basic navigation mechanism of the desert ant *Cataglyphis fortis*. *Myrmecological News*, 11, 53-62.
- Shettleworth, S. J. (2001). Animal cognition and animal behavior. *Animal Behavior*, 61(2), 277-286.
- Sieglmann, H. T. (2013). Turing on super-Turing and adaptivity. *Progress in Biophysics & Molecular Biology*, 113(1), 117-126.
- Solé, R., Moses, M. & Forrest, S. (2019). Liquid brains, solid brains. *Philosophical Transactions B*, 374, 20190040.
- Solé, R. & Macia, J. (2011). Synthetic biocomputation: the possible and the actual. In T. Lenaerts, M. Giacobini, H. Bersini, P. Bourguine, M. Dorigo & R. Doursat (Eds), *Advances in Artificial Life, ECAL, 2011* (without numeration). Cambridge, MA: MIT Press.
- Stieb, S. (2010). Frontal overview of the brain of the desert ant *Cataglyphis fortis*. Retrieved from [http://www.graduateschools.uni-wuerzburg.de/life\\_sciences/gsls\\_newsletter/](http://www.graduateschools.uni-wuerzburg.de/life_sciences/gsls_newsletter/)
- Syropoulos, A. (2008). *Hypercomputation: Computing Beyond the Church-Turing Barrier*. New York: Springer.
- Thiélin-Bescond, M. & Beugnon, G. (2005). Vision-independent odometry in the ant *Cataglyphis cursor*. *Naturwissenschaften*, 92(4), 193-197.
- Wegner, P. (1997). Why interaction is more powerful than algorithms. *Communication of the ACM*, 40 (5), 80-91.
- Wegner, P. (1998). Interactive foundations of computing. *Theoretical Computer Science*, 192 (2), 315-351.
- Wehner, R. (2003). Desert ant navigation: how miniature brain solve complex systems. *Journal of Comparative Physiology A*, 189(8), 579-588.
- Wehner, R., Boyer, M., Loertscher, F., Sommer, S. & Menzi, U. (2006). Ant navigation: one-way routes rather than maps. *Current Biology*, 16(1), 75-79.
- Wehner, R. & Srinivasan, M. V. (2003). Path integration in insects. In K. J. Jeffery (Ed.), *The Neurobiology of Spatial Behaviour* (pp. 9-30). Oxford: Oxford University Press.
- Whener, R. & Whener, R. (1990). Insect navigation: use of maps or Ariadne's thread. *Ecology & Evolution*, 2(1), 27-48.
- Wilensky, U. (1997). *NetLogo Ants model*. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University. Retrieved from <http://ccl.northwestern.edu/netlogo/models/Ants>.
- Wilensky, U. (1999). NetLogo. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University. Downloaded from <http://ccl.northwestern.edu/netlogo/>.
- Wilensky, U. & Rand, W. (2015). *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. Cambridge, MA: MIT Press.
- Wittlinger, M., Whener, R. & Wolf, H. (2006). The ant odometer: stepping on stilts and stumps. *Science*, 312(5782), 1965-1967.
- Wittlinger, M., Wolf, H. & Whener, R. (2007). Hair plate mechanoreceptors associated with body segments are not necessary for three-dimensional path integration in desert ants, *Cataglyphis fortis*. *Journal of Experimental Biology*, 210(3), 375-382.
- Wolf, H. (2011). Odometry and insect navigation. *Journal of Experimental Biology*, 214(10): 1629-1641.