Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 24 (2013) 103 – 113

Procedia Computer Science

17th Asia Pacific Symposium on Intelligent and Evolutionary Systems, IES2013

A Simulation Framework for Neuron-based Molecular Communication

Junichi Suzuki*, Harry Budiman, Timothy A. Carr and Jane H. DeBlois

Department of Computer Science, University of Massachusetts Boston, Boston, MA 02125, USA

Abstract

Molecular communication is a nanoscale communication paradigm that utilizes molecules as a communication medium between nanoscale devices. Development and availability of simulators is critical in this emerging research area since research on molecular communication heavily depends on simulations as the only means to evaluate analytical models and protocols. This paper proposes a simulation framework for neuron-based molecular communication, which utilizes electrochemical signals through neuronal networks. The proposed framework is designed to integrate various simulation components such as visualizers and editors for neuronal networks, media access controllers to neuronal networks and communication schedulers for neuronal signal transmissions. This paper describes key simulation components in the proposed framework and validates them with a case study that implements a TDMA-based signaling protocol in a simulated neuronal network and examines its performance characteristics with evolutionary multiobjective optimization algorithms.

Selection and peer-review under responsibility of the Program Committee of IES2013

Keywords: Molecular communication; intrabody nanonetworks; simulations; communication optimization; evolutionary algorithms

1. Introduction

Molecular communication is a communication paradigm that utilizes molecules as a communication medium between nanomachines. Nanomachines are nanoscale devices that perform simple computation, sensing and/or actuation tasks [1]. They may be man-made devices built in the *top-down* approach, downscaling the current

* Corresponding author. Tel.: +1-617-287-6462; fax: +1-617-287-6433.
E-mail address: jxs@cs.umb.edu.

1877-0509
Selection and peer-review under responsibility of the Program Committee of IES2013
doi:10.1016/j.procs.2013.10.032

microelectronic and micro-electromechanical technologies, or in the *bottom-up* approach, assembling synthesized nanomaterials such as graphene nanoribbons and carbon nanotubes [2]. Alternatively, nanomachines may be *bio-hybrid*, integrating man-made nanostructures with biological materials such as DNA strands, artificial cells and genetically engineered cells, or *bio-enabled*, synthesizing biological materials without man-made nanostructures [2, 3]. Due to its advantages such as inherent nanometer scale, biocompatibility and energy efficiency, a key application domain of molecular communication is *intrabody nanonetworks*, where nanomachines are networked through molecular communication to perform their tasks in the human body for biomedical and prosthetic purposes (e.g., in-situ physiological sensing, biomedical anomaly detection, targeted drug release, medical operations with cellular/molecular level precision and neural signal transduction) [3-5].

Molecular communication is often classified to *short-range* communication (nanometers to millimeters) and *long-range* communication (millimeters to meters) [6]. While several models and protocols have been proposed for short-range molecular communication (e.g., molecular motors [7], calcium signaling [8] and bacteria communication [9]), research efforts on long-range communication are relatively limited [6].

This paper focuses on long-range molecular communication that utilizes neurons as a primary communication component [10-12]. A neuron-based intrabody nanonetwork consists of a set of nanomachines and a network of neurons that are artificially formed into a particular topology. It allows nanomachines to interface (i.e., activate and deactivate) neurons and communicate to other nanomachines with electrochemical signals through a chain of neurons. This paper proposes and evaluates a simulation framework for neuron-based molecular communication. The proposed simulation framework is designed to integrate various simulation components such as three-dimensional visualizers and editors for neuronal networks, media access controllers to neuronal networks and communication schedulers for neuronal signal transmissions. This paper describes key simulation components in the proposed framework and validates them with a case study that implements a TDMA-based signaling protocol in a simulated neuronal network and examines its performance characteristics with evolutionary multiobjective optimization algorithms (EMOAs).

2. Neuron-based Molecular Communication

This section provides a background on neuronal signaling and intrabody nanonetworks.

2.1. Neuronal Signaling

Neurons are a fundamental component of the nervous system, which includes the brain and the spinal cord. They are electrically excitable cells that process and transmit information via electrical and chemical signaling.

The structure of a neuron consists of a cell body (or soma), dendrites and an axon (Fig. 1(a)). The soma is the central part of a neuron. It can vary from 4 to 100 micrometers in diameter. Dendrites are thin structures that arise from the soma. They form a complex “dendritic tree” that extends the farthest branch a few hundred micrometers from the soma. Dendrites are where the majority of inputs to a neuron occur. An axon is a cellular extension that arises from the soma. It branches before it terminates and travels through the body in bundles called nerves. Its length can be over one meter in the human nerve that arises from the spinal cord to a toe.

Neurons are connected with each other to form a network(s). Neurons communicate with others via synapses, each of which is a membrane-to-membrane junction between two neurons. A synapse contains molecular machinery that allows a (presynaptic) neuron to transmit a chemical signal to another (postsynaptic) neuron. In general, signals are transmitted from the axon of a presynaptic neuron to a dendrite of a presynaptic neuron. An axon transmits an output signal to a postsynaptic neuron, and a dendrite receives an input signal from a presynaptic neuron.

Presynaptic and postsynaptic neurons maintain voltage gradients across their membranes by means of voltage-gated ion channels, which are embedded in the presynaptic membrane to generate the differences between intracellular and extracellular concentration of ions (e.g., Ca^{2+}) [13]. Changes in the cross-membrane ion concentration (i.e., voltage) can alter the function of ion channels. If the concentration (i.e., voltage) changes by a large enough amount (e.g., approximately 80 mV in a giant squid), ion channels initiate a voltage-dependent process; they pump extracellular ions inward. Upon the increase in intracellular ion concentration, the presynaptic neuron releases a chemical called a neurotransmitter (e.g., acetylcholine (ACh)), which travels through the synapse

from the presynaptic neuron's axon terminal to the postsynaptic neuron's dendrite. The neurotransmitter electrically excites the postsynaptic neuron, and the neuron generates an electrical pulse called an action potential. This signal travels rapidly along the neuron's axon and activates synaptic connections (i.e., opens ion channels) when it arrives at the axon's terminals. This way, an action potential triggers cascading neuron-to-neuron communication. Fig. 1(b) shows how Ca^{2+} concentration changes in a neuron. When the concentration peaks, the neuron releases a neurotransmitter to trigger an action potential. Upon a neurotransmitter release, the neuron goes into a refractory period (*Tr* in Fig. 1(b)), which is the time required for the neuron to replenish its internal Ca^{2+} store. During *Tr*, it cannot process any incoming signals. The refractory period is approximately two milliseconds in a giant squid.

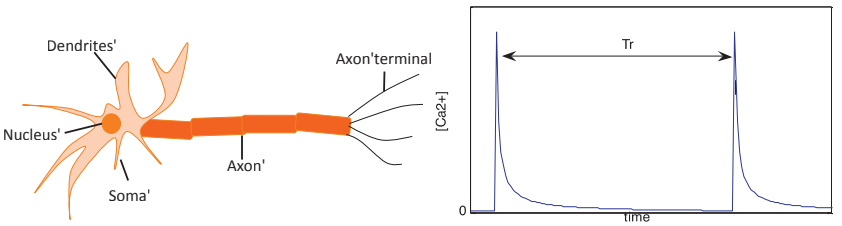

Fig. 1. (a) The structure of a neuron; (b) Intracellular Ca^{2+} concentration

2.2. Neuron-based Intrabody Nanonetworks

This paper assumes neuronal signaling in a network of natural neurons that are artificially grown and formed into particular topology patterns. This assumption is made upon numerous research efforts to grow neurons on substrates (e.g., [14]) and design topologically-specific neuronal networks (e.g., [15-17]).

Fig. 2 illustrates an example neuron-based intrabody nanonetwork. It contains an artificially-grown neuronal network and several nanomachines such as sensors and a sink. Sensors use neuronal signaling to transmit sensor data to the sink, which might work as an actuator or transducer. As potential applications, prosthetic devices and medical rehabilitation devices could leverage neuron-based intrabody nanonetworks to better perform sensing, transducing and actuation tasks in the body.

This paper assumes that nanomachines (e.g., sensors) interact with neuronal networks in a non-invasive manner. This means that it is not required to insert carbon nanotubes into neurons so that nanomachines can trigger signaling. Nanomachines may use a neurointerface based on chemical agents (e.g., acetylcholine and mecamylamine [10]) or light [18].

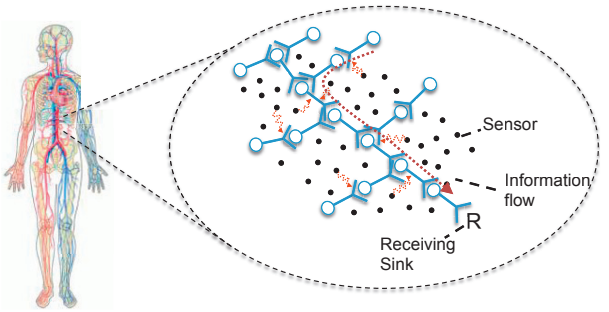

Fig. 2. An Example Neuron-based Intrabody Nanonetwork

3. Related Work

Development and availability of simulators is critical for molecular communication research since it heavily depends on simulations as the only means to evaluate analytical models and protocols. Several simulators exist for short-range molecular communication [19-24]. For example, one of the first research efforts is reported in [19], where Moore et al. describe simulation requirements for nanonetworks with molecular motors. NanoNS [20] and N2Sim [21] are simulators for molecular diffusion based on Brownian motion among stationary nanomachines. In [22], Toth et al. also investigate a simulator for Brownian motion. It takes a dual time-step approach to manage the runtime complexity due to a large number of simulated particles. While a particle is far from the destination, its movement is simulated in a coarse-grained time steps. When it comes closer to the destination, a fine-grained time steps are used. In [23], a simulator development is reported for molecular diffusion by taking advantage of High Level Architecture (HLA). In [24], a generic simulation platform is proposed to accommodate various types of nanomachines, channel models, molecular propagation models and nanomachine mobility models.

In contrast to short-range communication, simulator development has not been reported in literature for long-range molecular communication, particularly neuron-based communication. This paper describes the first research attempt to develop and evaluate a simulator for neuron-based molecular communication.

4. The Proposed Simulation Framework

The proposed framework is designed to aid communication protocol designers to simulate neuron-based intrabody nanonetworks and validate protocols in the simulated nanonetworks. It integrates various simulation components such as visualizers and editors for neuronal networks, neuronal topology generators, media access controllers to neuronal networks and communication schedulers for neuronal signal transmissions. The proposed framework is implemented in Java.

This paper focuses on two components in the proposed framework, neuronal topology visualizer and Neuronal TDMA optimizer, and discusses an XML-based data transport between those components (Fig. 3). The topology visualizer allows the user to edit and view the structure of a neuronal network. Neuronal TDMA optimizer seeks the optimal schedules for nanomachines to fire neuronal signaling in a given neuronal network. The data transport encodes the topology of a neuronal network and the location of nanomachines with XML.

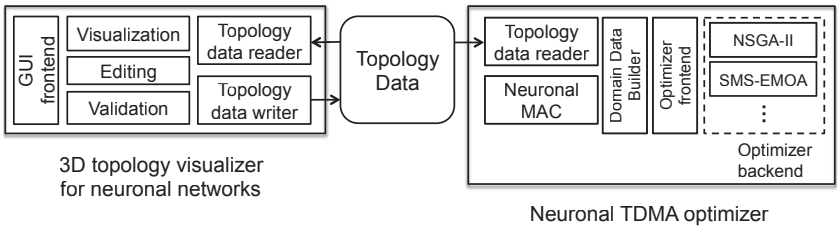

Fig. 3. An Architectural Overview of the Proposed Simulation Framework

Fig. 4 shows a block diagram of the steps required to run a simulation. The user graphically defines a set of neurons and its topology as well as the location of nanomachines. Those information is formatted in XML and supplied to Neuronal TDMA optimizer. The user configures various parameters for neuronal signaling optimization and obtains the optimal signaling schedules. The schedules are then examined on a simulated neuronal network in order to obtain performance characteristics for each signaling schedule based on a given set of metrics. Simulation results are processed to produce graphical figures so that the user can analyze signaling schedules and their performance characteristics.

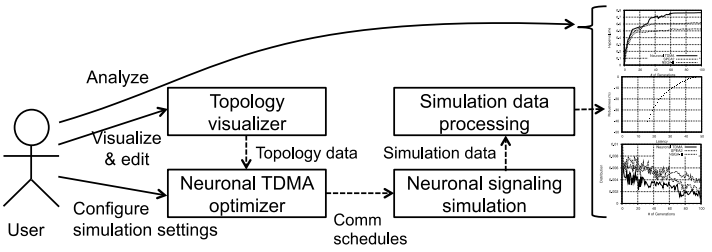

Fig. 4. Steps Required to Run a Simulation

4.1. Neuronal Topology Visualizer

Neuronal topology visualizer is a three-dimensional GUI tool that allows the user to define a series of structural elements in a simulated neuronal network; e.g., the shape and location of each neuronal parts (e.g., dendritic tree, soma and axon) and the connectivity of neurons. In addition, nanomachine locations and nanomachine-to-neuron interface are defined with this tool. The user can intuitively view and define these structural elements with three-dimensional editing features such as zoom-in/out, pan-up/down/left/right, toggle and viewpoint shit. The proposed framework manages and renders all structural elements as OpenGL objects. (It currently uses a Java implementation of OpenGL, JOGL.) Fig. 5 shows several example screenshots of neuronal topology visualizer.

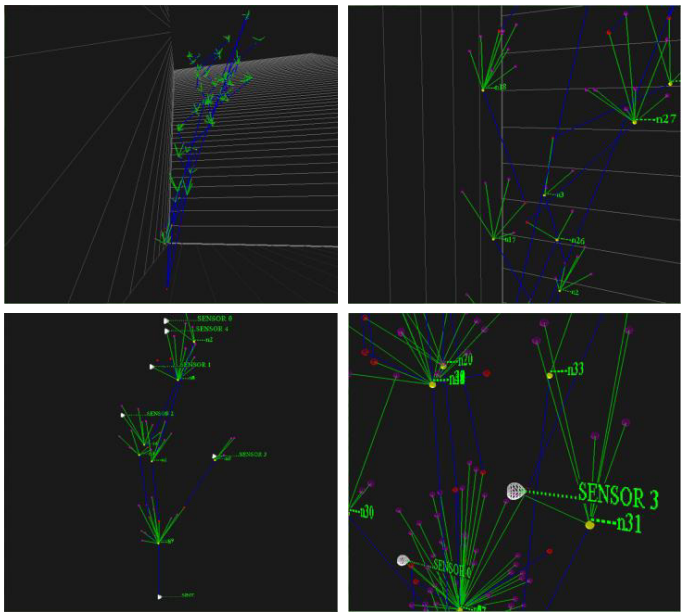

Fig. 5. Screenshots of Neuronal Topology Visualizer

Neuronal topology visualizer also validates the structure of a neuronal network and reports the user editing errors such as unconnected neurons, missing nanomachines and nanomachines unconnected with neurons. If no editing errors are found, the visualizer can generate the structural information on a neuronal network into an XML file.

4.2. XML Data Transport

XML data transport in the proposed simulation encodes the structure of a neuronal network in XML and transports the structural information among simulation components. Fig. 6 shows the XML DTD (Document Type Definition) that XML data should follow in the proposed framework. Each simulation component in the proposed framework implements input and/or output modules to read and write XML data formatted in the DTD in Fig. 6.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT simulation (description?, nanomachines?, neuronalNetwork)>
<!ELEMENT description (#PCDATA)>

<!ELEMENT nanomachines (description?, sensor+, sink)>
<!ELEMENT sensor (sensorName?, neuronAssociatedWith)>
      <!ATTLIST sensor id ID #REQUIRED>
<!ELEMENT sensorName (#PCDATA)>
<!ELEMENT neuronAssociatedWith (description?)>
      <!ATTLIST neuronAssociatedWith idn IDREF #REQUIRED>
<!ELEMENT sink (sinkName?, neuronAssociatedWith)>
      <!ATTLIST sink id ID #REQUIRED>
<!ELEMENT sinkName (#PCDATA)>

<!ELEMENT neuronalNetwork (description?, neuron+)>
<!ELEMENT neuron (neuronName?, dendriteEndSynapticContact*,axonBranchEndSynapticContact*,
                  soma, dendriteEndSynapseLocation*, axonBranchEndSynapseLocation+,
                  dendriteSegment*)>
      <!ATTLIST neuron idn ID #REQUIRED>
<!ELEMENT neuronName (#PCDATA)>
<!ELEMENT dendriteEndSynapticContact (neuronName?)>
      <!ATTLIST dendriteEndSynapticContact idn IDREF #REQUIRED>
<!ELEMENT axonBranchEndSynapticContact (neuronName?)>
      <!ATTLIST axonBranchEndSynapticContact idn IDREF #REQUIRED>
<!ELEMENT soma (pointName?, x, y, z)>
      <!ATTLIST soma idn IDREF #REQUIRED>
<!ELEMENT dendriteEndSynapseLocation (pointName?, x, y, z)>
      <!ATTLIST dendriteEndSynapseLocation idd ID #REQUIRED>
<!ELEMENT axonBranchEndSynapseLocation (pointName?, x, y, z)>
      <!ATTLIST axonBranchEndSynapseLocation ida ID #REQUIRED>
<!ELEMENT pointName (#PCDATA)>
<!ELEMENT x (#PCDATA)>
      <!ATTLIST x type (double|int) #REQUIRED>
<!ELEMENT y (#PCDATA)>
      <!ATTLIST y type (double|int) #REQUIRED>
<!ELEMENT z (#PCDATA)>
      <!ATTLIST z type (double|int) #REQUIRED>
<!ELEMENT dendriteSegment (rootEnd, synapseEnd)>
<!ELEMENT rootEnd (x, y, z)>
<!ELEMENT synapseEnd (x, y, z)>
```

Fig. 6. XML DTD used in the Proposed Framework

4.3. Neuronal TDMA Optimizer

Neuronal TDMA optimizer is a communication optimization suite in the proposed simulation framework (Fig. 3). It performs scheduling optimization for Neuronal TDMA [12], which is a single-bit Time Division Multiple Access (TDMA) protocol for neuron-based intrabody nanonetworks. Neuronal TDMA allows nanomachines to multiplex and parallelize neuronal signaling while avoiding signal interference to ensure that signals reach the sink node. Neuronal TDMA optimizers can plug optimization algorithms as its backend modules (Fig. 3). Given a particular optimization algorithm, the optimizer seeks the optimal signaling schedules (i.e., which neurons to activate and when to activate them to trigger signal transmissions) for sensor nodes with respect to given optimization objectives.

4.3.1 Neuronal TDMA

Neuronal TDMA periodically assigns a *time slot* to each sensor node. Sensors fire neurons, one after the other, each using its own time slot. This allows multiple sensors to transmit signals to the sink through the shared neuronal network. Each sensor transmits a single signal (a single bit) within a single time slot. This single-bit-per-slot design is based on two assumptions: (1) a signal (i.e., action potential) is interpreted with two levels of amplitudes, which represent 0 and 1, and (2) after a signal transmission, a neuron goes into a refractory period (waiting/sleeping period).

An important goal of Neuronal TDMA is to avoid signal interference, which occurs when multiple signals fire the same neuron at the same time and leads to corruption of transmitted sensor data at the sink. Signals can easily interfere with each other if sensors fire their neighboring neurons randomly. Neuronal TDMA is intended to eliminate signal interference by scheduling which sensors fire which neurons with respect to time. Neuronal TDMA optimizer seeks the optimal TDMA schedules for a set of sensors in a given neuronal network.

Fig. 7 shows an example intrabody nanonetwork that contains four nanomachines (three sensors and a sink) and a network of five neurons (n_1 to n_5). Figure 5 illustrates an example TDMA schedule for those sensors to fire neurons. The scheduling cycle period lasts 6 time slots (T_s = 5). The sensor s_1 fires the neuron n_4 to initiate signaling in the first time slot T_1. The signal travels through n_5 in the next time slot T_2 to reach the sink. The sensor s_2 transmits a signal on n_3 in T_2. During T_2, two signals travel in the neuronal network in parallel. The duration of each time slot must be equal to, or longer than, the refractory period T_r (Fig. 1(b)).

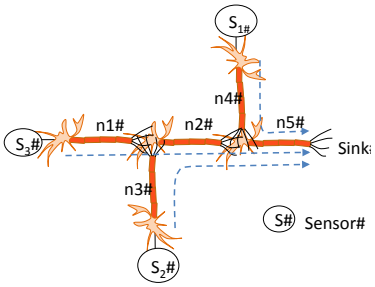

Fig. 7 An Example Neuron-based Nanonetwork

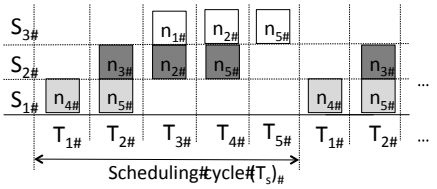

Fig. 8 An Example TDMA Schedule

The scheduling problem in Neuronal TDMA is defined as an optimization problem where an intrabody nanonetwork contains M sensors, $S = \{s_1, s_2, ..., s_i, ..., s_M\}$, and N neurons, $N = \{n_1,n_2,...,n_j,...,n_N\}$. Each sensor transmits at least one signals to the sink during the scheduling cycle T_s. $E^{si} = \{E_1^{si}, E_2^{si}, ..., E_k^{si}, ..., E_{|Esi|}^{si}\}$ denotes the signals that a sensor s_i transmits to the sink. $|E_{si}|$ is the total number of signals that s_i transmits during the scheduling cycle T_s. This paper considers the following three optimization objectives:

- *Signaling yield* (Eq.1): It is to be maximized. This objective indicates the total number of signals that the sink receives from all M sensors during the scheduling cycle T_s.

$$f_Y = \sum_{i=1}^{M} |E^{s_i} \tag{1}$$

- *Signaling fairness among sensors* (Eq. 2): It is to be maximized. $t_d^{k(sl)}$ denotes the departure time of the k-th signal that s_l transmits to the sink. This objective encourages sensors to equally access the shared neuronal network for signaling in order to avoid a simulation where a limited number of sensors dominate the network. Higher fairness means that sensors access the neuronal network more equally.

$$f_F = \sum_{l=1}^{M} \sum_{m=1}^{M} \sum_{k=1}^{|E^{s_l}|} \frac{1}{|t_d^{k(s_l)} - t_d^{k(s_m)}|}, \quad l \neq m \tag{2}$$

- *Signaling delay* (Eq. 3): It is to be minimized. $T_a^{|Esi|(si)}$ denotes the arrival time at which the sink receives the last (the $|E^{si}|$-th) signal that s_i transmits. This objective indicates how soon the sink receives all signals from all M sensors. It determines the scheduling cycle period T_s (T_s=f_D).

$$f_D = max_{s_i \in S}\ t_a^{|E^{s_i}|((s_i)} \tag{3}$$

4.3.2 Evolutionary Algorithms in Neuronal TDMA Optimizer

Neuronal TDMA optimizer currently uses evolutionary multiobjective optimization algorithms (EMOAs) to solve its scheduling optimization problem. EMOAs are used as the optimizer's backend modules (Fig. 3). An EMOA iteratively evolves the population of solution candidates, called *individuals*, through several operators (e.g., crossover, mutation and selection operators) toward the Pareto-optimal solutions in the objective space.

In order to seek Pareto optimality, the notion of dominance [25] plays an important role. An individual *i* is said to dominate an individual *j* if both of the following conditions are hold.

- *i*'s objective values are superior than, or equal to, *j*'s in all objectives.
- *i*'s objective values are superior than *j*'s in at least one objectives.

The notion of dominance is used in some of EMOAs (e.g., NSGA-II [26]) in the proposed simulation framework. It is also used for processing and evaluating simulation data (c.f. Fig. 4).

In order to run EMOAs in Neuronal TDMA optimizer, each individual represents a particular TDMA schedule for *M* sensors. Fig. 9 shows the structure of an example individual, which represents the schedule shown in Fig. 8. In this example, the first sensor, s_1, fires its neighboring neuron, n_4, in the first time slot T_1. Similarly, s_2 and s_3 fire their neighboring neurons (n_2 and n_1) in T_2 and T_3, respectively.

S_3	0	0	1	0	0
S_2	0	1	0	0	0
S_1	1	0	0	0	0
	T_1	T_2	T_3	T_4	T_5

Fig. 9 Individual Representation

5. Case Study: Neuronal TDMA Optimization with EMOAs

This section evaluates the proposed simulation framework with a case study that implements Neuronal TDMA in a simulated neuronal network and examines its performance characteristics with two evolutionary multiobjective optimization algorithms (EMOAs): NSGA-II [26] and SMS-EMOA [27].

In this case study, a simulated neuronal network contains 43 neurons. 11 sensors are evenly distributed in the neuronal network. The topology of neurons follows a tree structure (Fig. 5). The user defines a simulated neuronal network with Neuronal topology visualizer (Section 4.1). The visualizer verifies the network's structure and encodes it in XML. XML data transport (Section 4.2) passes the XML-encoded data to Neuronal TDMA optimizer (Section 4.3), which seeks the optimal TDMA schedules with NSGA-II and SMS-EMOA.

EMOAs in the proposed framework are configured with a set of parameters shown in Table 1. *Q* denotes the total number of time slots in an individual (Q = 15 in Fig. 10).

Table 1. EMOA Configurations

Parameter	Value
Population size	100
Maximum number of generations	100
Crossover rate	0.9
Mutation rate	1/Q

Fig. 10 shows how individuals increase the union of the hypervolumes that they dominate in the objective space as the number of generations grows in NSGA-II and SME-EMOA. The hypervolume metric quantifies the optimality and diversity of individuals [28]. A higher hypervolume means that individuals are closer to the Pareto-optimal front and more diverse in the objective space. As Fig. 10 shows, both NSGA-II and SMS-EMOA rapidly increase their hypervolume measures in the first 10 generations and converge around the 60th generation. At the last generation, all individuals are non-dominated in the population. This demonstrates that NSGA-II and SMS-EMOA allow individuals to efficiently evolve and improve their quality and diversity within 100 generation.

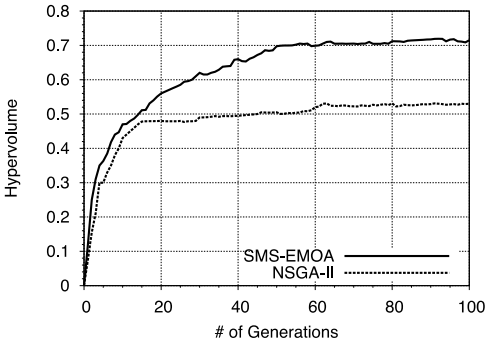

Fig. 10 Hypervolume

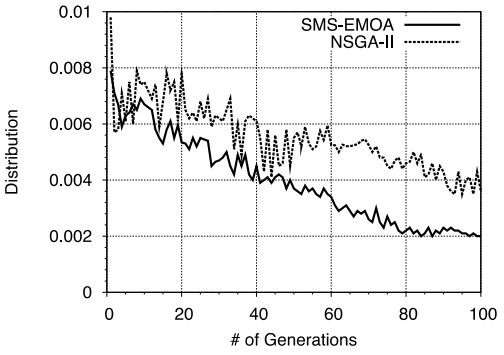

Fig. 11 Distribution

Fig. 11 illustrates the diversity of individuals with the distribution metric. This metric measures the degree of uniform distribution of individuals in the objective space. It is computed as the standard deviation of Euclidean distances among individuals:

$$\sqrt{\frac{\sum_{i=1}^{N}(d_i - d_\mu)^2}{N-1}}$$

d_i denotes the Euclidean distance between a given individual (the i-th individual in the population) and its closest neighbor in the objective space. d_u denotes the mean of d_i. N denotes the number of individuals in the population. The objective space is normalized to compute the distribution metric. Lower distribution means that individuals are more uniformly (or evenly) distributed. As shown in Fig. 11, both NSGA-II and SMS-EMOA improve the diversity of individuals as the number of generations grows. This result is consistent with the result in Fig. 10.

Table 2. Average Objective Values

	Yield (f_Y)	Fairness (f_F)	Delay (f_D)
SMS-EMOA	20.75 (4.67)	0.12 (0.43)	24.01 (5.44)
NSGA-II	16.56 (3.09)	0.07 (0.99)	31.87 (7.33)

Table 2 shows the average of each objective value at the last generation. A value in parentheses indicates a standard deviation of objective values that an EMOA yields in 20 independent simulations.

Figs. 11 and 12 and Table 2 demonstrate that Neuronal TDMA optimizer performs scheduling optimization efficiently and effectively. In summary, this case study confirms that the proposed simulation framework successfully aid communication protocol designers to define/verify neuron-based nanonetworks and optimize TDMA scheduling on simulated neuron-based nanonetworks.

6. Conclusion

This paper proposes and evaluates a simulation framework for neuron-based molecular communication. The proposed framework integrates a neuronal network visualizer/editor and a neuronal signaling scheduler through an XML data transport. This paper describes those simulation components and validates them with a case study that implements a TDMA-based signaling protocol in a simulated neuronal network and examines its performance characteristics with evolutionary multiobjective optimization algorithms.

References

[1] I. F. Akyildiz, F. Brunetti, and C. Blazquez. Nanonetworks: a new communication paradigm. Computer Networks Journal, 52(12), 2008.
[2] I. F. Akyildiz and J. M. Jornet. Electromagnetic wireless nanosensor networks. Elsevier Nano Communication Networks, 1(1), 2010.
[3] R. A. Freitas, Current status of nanomedicine and medical nanorobotics. J. Computational and Theoretical Nanoscience, 2(1), 2005.
[4] B. Atakan, O. B. Akan, and S. Balasubramaniam. Body area nanonetworks with molecular communications in nanomedicine. IEEE Communications Magazine, 50(1), 2012.
[5] I. F. Akyildiz and J. M. Jornet. The internet of nano-things. IEEE Wireless Communications, 17(6), 2010.
[6] L. P. Gine and I. F. Akyildiz. Molecular communication options for long range nanonetworks. Computer Networks, 53, 2009.
[7] M. Moore, A. Enomoto, T. Nakano, R. Egashira, T. Suda, A. Kayasuga, H. Kojima, H. Sakakibara, and K. Oiwa. A design of a molecular communication system for nanomachines using molecular motors. In Proc. IEEE Int'l Conference on Pervasive Computing and Communications Workshops, 2006.
[8] T. Nakano, T. Suda, M. Moore, R. Egashira, A. Enomoto, and K. Arima. Molecular communication for nanomachines using intercellular calcium signaling. In Proc. IEEE Int'l Conf. on Nanotechnology, 2005.
[9] C. Wyart, C. Ybert, L. Bourdieu, C. Herr, C. Prinz, and D. Chatenay. Constrained synaptic connectivity in functional mammalian neuronal networks grown on patterned surfaces. J. Neurosci. Methods, 117(2), 2002.
[10] S. Balasubramaniam, N. T. Boyle, A. Della-Chiesa, F. Walsh, A. Mardinoglu, D. Botvich, and A. Prina-Mello. Development of artificial neuronal networks for molecular communication. Nano Communication Networks, 2(2-3), 2011.
[11] J. Suzuki and S. Balasubramaniam. Networking and scheduling in neuron-based molecular communication. In Proc. of NSF Workshop on Biological Computations and Communications. November 2012.
[12] J. Suzuki, S. Balasubramaniam, and A. Prina-Mello. Multiobjective TDMA optimization for neuron-based molecular communication. In Proc. of the 7th Int'l Conference on Body Area Networks. September 2012.
[13] G. Stuart, J. Schiller, and B. Sakmann. Action potential initiation and propagation in rat neocortical pyramidal neurons. Journal of Physiology, 505.3, 1997.
[14] T. D. Nguyen-Vu, H. Chen, A. M. Cassell, R. J. Andrews, M. Meyyappan, and J. Li. Vertically aligned carbon nanofiber architecture as a multifunctional 3-D neural electrical interface. IEEE Trans. Biomed. Eng., 54(6), 2007.
[15] S. B. Jun, M. R. Hynd, N. Dowell-Mesfin, K. L. Smith, J. N. Turner, W. Shain, and S. J. Kima. Low-density neuronal networks cultured using patterned polyl-L-lysine on microelectrode arrays. Journal of Neurosci. Methods, 160(2), 2007.
[16] F. Morin, N. Nishimura, L. Griscomb, B. LePioufle, H. Fujita, Y. Takamura, and E. Tamiya. Constraining the connectivity of neuronal networks cultured on microelectrode arrays with microSSuidic techniques: A step towards neuron-based functional chips. Biosensors and Bioelectronics, 21, 2006.
[17] C. Wyart, C. Ybert, L. Bourdieu, C. Herr, C. Prinz, and D. Chatenay. Constrained synaptic connectivity in functional mammalian neuronal networks grown on patterned surfaces. J. Neurosci. Methods, 117(2), 2002.
[18] N. Grossman, K. Nikolic, and P. Degenaar. The neurophotonic interface: stimulating neurons with light. The Neuromorphic Engineer, 2008.
[19] M. Moore, A. Enomoto, T. Nakano, T. Suda, A. Kayasuda, H. Kojima, H. Sakakibara and K. Oiwa. Simulation of a molecular motor based communication network. In Proc. Int'l Conference on Bio-Inspired Models of Network, Information and Computing Systems, 2006.
[20] E. Gul, B. Atakan and O. B. Akan. NanoNS: A nanoscale network simulator framework for molecular communications. Nano Communication Networks, 1(2), 2010.
[21] I. Llatser, I. Pascual, N. Garralda, A. Cabellos-Aparicio and E. Alarcon. N3Sim: A Simulation framework for diffusion-based molecular communication. IEEE TC on Simulation, 8, 2011.
[22] A. Toth, D. Banky and V. Grolmusz. 3-D Brownian motion simulator for high-sensitivity nanobiotechnological applications. IEEE Trans.

NanoBioscience, 10(4), 2011.
[23] A. Akkaya and T. Tugcu. dMCS: distributed molecular communication simulator. In Proc. Int'l Conference on Body Area Networks, 2013.
[24] L. Felicetti, M. Femminella and G. Reali. A simulation tool for nanoscale biological networks. Nano Communication Networks, 3(1), 2012.
[25] N. Srinivas and K. Deb. Multiobjective function optimization using nondominated sorting genetic algorithms. Evol. Computat., 2(3), 1995.
[26] K. Deb, S. Agrawal, A. Pratab and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In Proc. Int'l Conference on Parallel Problem Solving from Nature, 2000.
[27] N.Beume, B.Naujoks and M.Emmerich. SMS-EMOA:Multiobjective selection based on dominated hypervolume. *Eur. J. Oper. Res.*, 181(3), 2007.
[28] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms: A comparative study. In Proc. Int'l Conference on Parallel Problem Solving from Nature, 1998.