

NeuroSim: A Neuronal Signaling Simulator for Nanoscale Intrabody Sensor Networks

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

Department of Computer Science
University of Massachusetts, Boston
Boston, MA, 02125-3393, USA
{jxs, hbudiman, timcarr, hdeblois, dwwang}@cs.umb.edu

Abstract. This paper proposes and evaluates a simulator for nanoscale intrabody sensor networks that utilize neurons as a primary communication medium and neuronal signals (i.e., electrochemical signals) as communication signals. The proposed simulator, called NeuroSim, is designed to integrate various simulation components such as neuronal network generators, visualizers and editors and communication schedulers for neuronal signal transmissions. This paper describes key components in NeuroSim and evaluates them with a case study that implements a TDMA signaling protocol in a simulated neuronal network.

1 Introduction

A nanoscale system consists of one or more nanomachines, which are the most basic functional unit ranging from one to a few hundred nanometers. Each of them performs very simple computation, sensing and/or actuation tasks. An emerging design strategy for nanoscale systems is to *network* nanomachines for operating in larger physical spaces in higher spatial and temporal resolutions. Although individual nanomachines are limited in computation, sensing and actuation capabilities, an assembly of nanomachines can potentially organize into a “large-scale” network that spreads on centimeter to meter scale and collaboratively performs tasks that no individual nanomachines could.

Molecular communication is one of a few options to network nanomachines. It leverages molecules as a communication medium between nanomachines [1]. Due to its advantages such as inherent nanometer scale, biocompatibility and energy efficiency [2], a key application domain of molecular communication is intrabody nanonetworks where nanomachines are networked 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, 4].

This paper focuses on molecular communication that utilizes neurons as a primary component to build nanoscale intrabody sensor networks and proposes a simulator for neuron-based molecular communication. A neuron-based intrabody sensor network consists of a set of nanomachines (e.g., bio-sensors) and a

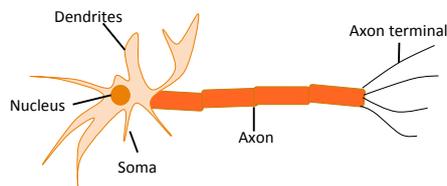


Fig. 1. The Structure of Neurons

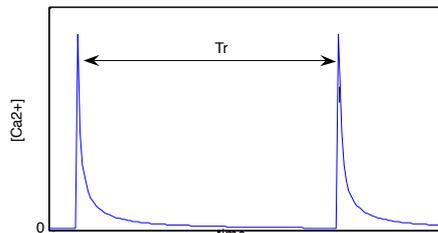


Fig. 2. Intracellular Ca^{2+} Concentration

network of neurons that are artificially formed into a particular topology [5–7]. It allows nanomachines to interface (i.e., activate and deactivate) neurons and communicate to other nanomachines with electrochemical signals through a chain of neurons. The proposed simulator, called NeuroSim, is designed to integrate various simulation components via XML data transport. Simulation components in NeuroSim include visualizers and editors for neuronal networks, neuronal network generators, media access controllers to neuronal networks and communication schedulers for neuronal signal transmissions. This paper describes key simulation components in NeuroSim and validates them with a case study that generates simulated neuronal networks and operates a Time Division Multiple Access (TDMA) signaling protocol in those networks.

2 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 (nanometers to millimeters) [8–13]. Most available simulators focus on simulation of molecular diffusion based on Brownian motion among stationary nanomachines [9–12]. In [13], a generic simulation platform is proposed to accommodate various types of nanomachines, channel models, molecular propagation models and nanomachine mobility models.

In contrast, for long-range molecular communication (millimeters to meters), simulator development has not been reported in literature except for NeuroSim. To the best knowledge of the authors of this paper, NeuroSim is the first simulator for molecular communication research with neuronal signaling.

This paper extends prior research efforts on neuron-based molecular communication, which study TDMA communication scheduling on neuronal networks [5–7]. Unlike this paper, those prior work never focus on development and engineering of a simulator for neuron-based molecular communication.

3 Neuron-based Molecular Communication

This section provides a background on neuron-based intrabody sensor networks.

3.1 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). 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 receive the majority of inputs to a neuron. An axon is a cellular extension that arises from the soma. It branches 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 junction between two neurons. A synapse contains molecular machinery that allows a (presynaptic) neuron to transmit a chemical signal to another (postsynaptic) neuron. 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+}) [14]. Changes in the cross-membrane ion concentration (i.e., voltage) can alter the function of ion channels. If the concentration changes by a large enough amount (e.g., approx. 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 to the postsynaptic neuron. The neurotransmitter electrically excites the postsynaptic neuron, which in turn 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. 2 shows how Ca^{2+} concentration changes in a neuron. When the concentration peaks, the neuron releases a neurotransmitter. Upon a neurotransmitter release, the neuron goes into a *refractory period* (T_r in Fig. 2), which is the time required for the neuron to replenish its internal Ca^{2+} store. During T_r , it cannot process any incoming signals. The refractory period is approximately two milliseconds in a giant squid.

3.2 Neuron-based Intrabody Sensor Networks

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., [15]) and form topologically-specific neuronal networks (e.g., [16–18]).

Fig. 3 illustrates an example intrabody sensor network. 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 sensor networks 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 [5]) or light [19].

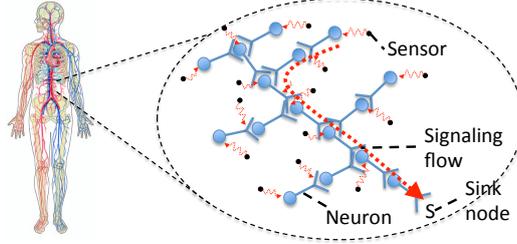


Fig. 3. An Example Neuron-based Intrabody Sensor Network

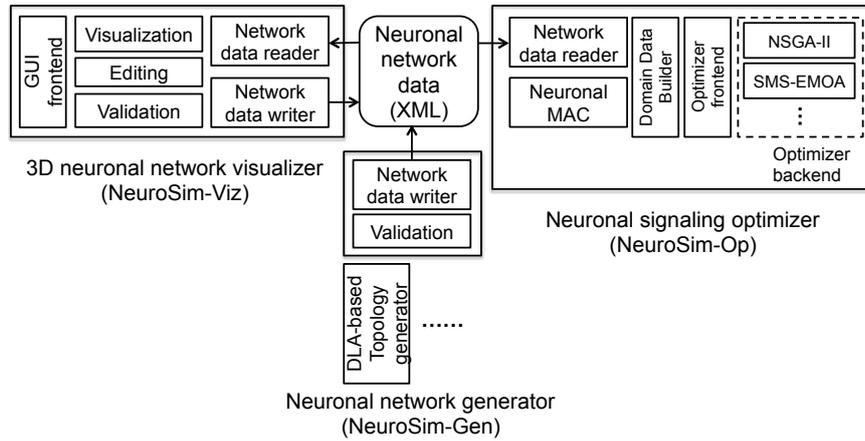


Fig. 4. An Architectural Overview of NeuroSim

4 NeuroSim

NeuroSim aids nanonetwork designers to specify intrabody neuron-based sensor networks, simulate neuronal signaling on the networks and validate communication protocols for neuronal signaling. It integrates various simulator components such as neuronal network generators, neuronal network visualizers and editors, media access controllers to neuronal networks and communication schedulers for neuronal signal transmissions. NeuroSim is implemented in Java.

This paper focuses on three components in NeuroSim: neuronal network generator (NeuroSim-Gen), neuronal network visualizer (NeuroSim-Viz) and neuronal signaling optimizer (NeuroSim-Op) (Fig. 4). NeuroSim-Gen forms and validates the topological structure of a neuronal network. NeuroSim-Viz visualizes the three-dimensional structure of a neuron-based sensor network. NeuroSim-Op seeks the optimal scheduling schedules for sensors to fire neuronal signaling in

a given neuronal network. The structure of a neuron-based sensor network is encoded and exchanged among simulator components with XML.

Fig. 4 shows the steps required to run a simulation. With NeuroSim-Gen, the user generates the topological structure of a neuronal network, encodes the generated network in XML and supplies it to NeuroSim-Viz. With NeuroSim-Viz, the user defines the locations of nanomachines to complete the specification of an intrabody sensor network. Alternatively, the user chooses to manually define a set of neurons and its topology without using NeuroSim-Gen. Once the structure of an intrabody sensor network is specified, it is formatted in XML and supplied to NeuroSim-Op. Using NeuroSim-Op, the user configures various parameters for neuronal signaling optimization and obtains the optimal signaling schedules for nanomachines. 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 evaluation metrics. Simulation results are processed to produce graphical figures so that the user can analyze signaling schedules and their performance characteristics.

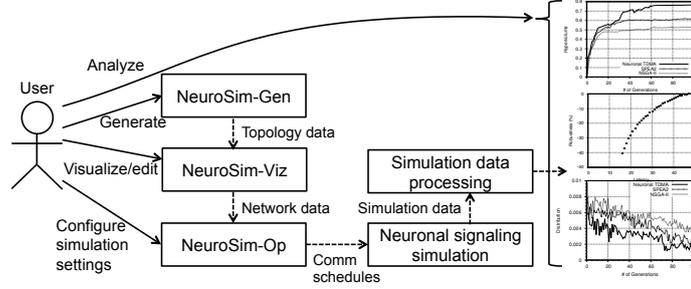


Fig. 5. Steps to Required to Run a Simulation

4.1 NeuroSim-Gen: Neuronal Network Generator

NeuroSim-Gen generates a three-dimensional neuronal network by building the topology of neuron-to-neuron connections and forming a dendritic tree for each neuron (Figs. 4 and 5).

Using the algorithm shown in Fig. 6, NeuroSim-Gen generates a tree-structured neuronal network that contains N neurons including L leaf neurons. Branching in a tree structure can be adjusted with two parameters (σ and D in Fig. 6). The root neuron in a neuronal network is assumed to be associated with the sink nanomachine. Non-

Input: N , Total number of neurons. $N \geq 2$

Input: L , Total number of leaf neurons.

$$1 \leq L < N$$

Input: σ, D , Control parameters

- 1: $l = 1, \mathcal{I} = \emptyset$
- 2: Create the root neuron and add it to \mathcal{I} .
- 3: **while** $l < L$ **do**
- 4: **if** $N - L - 1 = 0$ **then**
- 5: Create a leaf neuron and connect it with the root neuron.
- 6: $l = l + 1$
- 7: **end if**
- 8: $n = \lfloor \frac{N(N-l-|\mathcal{I}|, \sigma)}{D} \rfloor$
s.t. $0 \leq n < N - |\mathcal{I}| - l - 1$
- 9: Create a branch with a sequence of n neurons and insert the branch between a newly-created leaf neuron and a neuron that is randomly selected from \mathcal{I} .
- 10: $l = l + 1$
- 11: **end while**

Fig. 6. Topology Generation Algorithm

root neurons can be associated with sensors (Fig. 3).

Once the topology of a neuronal network is determined, NeuroSim-Gen performs an algorithm based on Diffusion Limited Aggregation (DLA) [20] to generate a dendritic tree for each neuron. It determines the structure aspects (e.g., dendritic length, distribution of segments and direction of branches) of each dendritic tree. It is known that the DLA-based algorithm can reproduce the spatial embedding of multiple neuronal types: granule cells, Purkinje cells, pyramidal cells, and dendritic and axonal trees of interneurons [20]. Fig. 7 shows an example dendritic tree for a particular neuron (*n19*). A yellow ball indicates a soma. Red lines indicate dendrites. Red balls indicate dendrite ends. A white ball indicates a nanomachine (sensor).

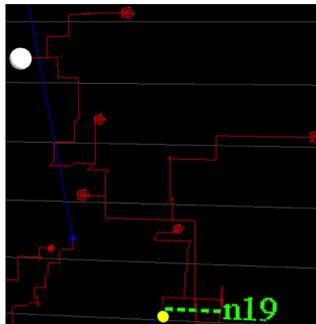


Fig. 7. An Example Dendritic Tree

4.2 NeuroSim-Viz: Neuronal Network Visualizer

NeuroSim-Viz 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 graphically view and define these structural elements with three-dimensional editing features such as zoom-in/out, pan-up/down/left/right, toggle and viewpoint shift. NeuroSim-Viz manages and renders all structural elements as OpenGL objects. (It currently uses a Java implementation of OpenGL, JOGL.) Figs. 8 and 9 show two example screenshots of intrabody sensor networks.

NeuroSim-Viz 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. When no editing errors are found, NeuroSim-Viz can generate the structural information on a neuronal network into an XML file.

4.3 NeuroSim-Op: Neuronal Signaling Optimizer

NeuroSim-Op is a communication optimization suite in NeuroSim (Fig. 4). It performs scheduling optimization for Neuronal TDMA [6], 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. NeuroSim-Op can plug optimization algorithms as its back-end modules (Fig. 4). Given a particular optimization algorithm, NeuroSim-Op seeks the optimal signaling schedules (i.e., which neurons to activate and when

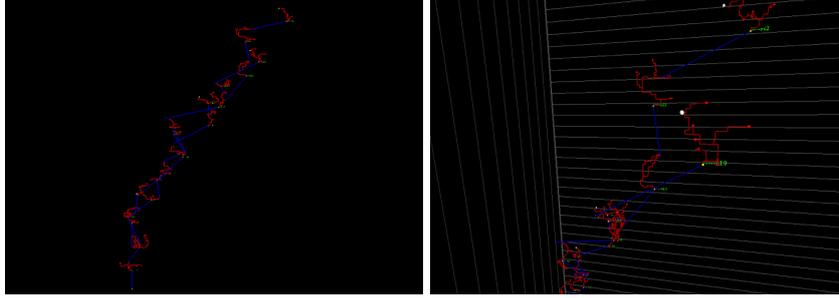


Fig. 8. An Intrabody Sensor Network of 20 Neurons and 5 Sensors

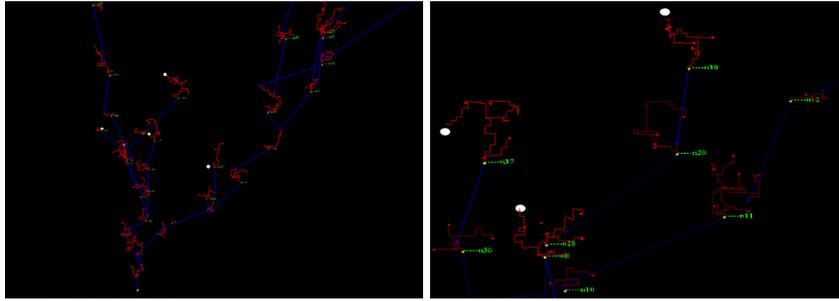


Fig. 9. An Intrabody Sensor Network of 40 Neurons and 10 Sensors

to activate them to trigger signal transmissions) for sensor nodes with respect to given optimization objectives.

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. An optimizer in Neuronal TDMA seeks the optimal TDMA schedules for a set of sensors in a given neuronal network.

Fig. 10 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). Fig. 11 illustrates an example TDMA schedule for those sensors to fire neurons. The scheduling cycle period lasts 5 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

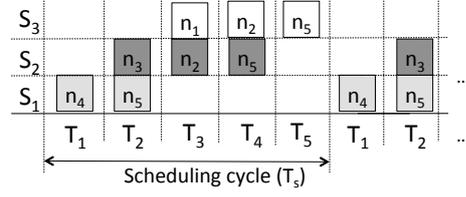
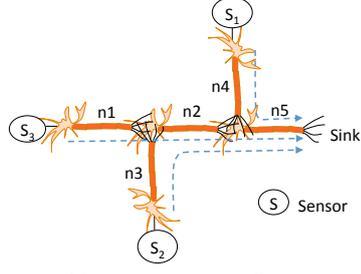


Fig. 10. An Example Sensor Network **Fig. 11.** An Example TDMA Schedule

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. 2).

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

- Signaling yield (f_Y) is computed as follows. It is to be maximized.

$$f_Y = \sum_{i=1}^M |E^{s_i}| \quad (1)$$

This objective indicates the total number of signals that the sink receives from all M sensors during the scheduling cycle T_s .

- Signaling fairness (f_F) is computed as follows. It is to be maximized.

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

$t_d^{k(s_l)}$ 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 situation where a limited number of sensors dominate the network. Higher fairness means that sensors access the neuronal network more equally.

- Signaling delay (f_D) is computed as follows. It is to be minimized.

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

$t_a^{|E^{s_i}|((s_i))}$ denotes the arrival time at which the sink receives the last (the $|E^{s_i}|$ -th) signal that s_i transmits. f_D indicates how soon the sink receives all signals from all M sensors. f_D determines the scheduling cycle period T_s ($T_s = f_D$).

Evolutionary Algorithms NeuroSim-Op currently uses evolutionary multiobjective optimization algorithms (EMOAs) to solve the scheduling optimization problem described above. EMOAs are used as NeuroSim-Op’s backend modules (Fig. 4). 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 [21] 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 in NeuroSim-Op (e.g., NSGA-II [22]). It is also used for processing and evaluating simulation data.

In order to run EMOAs in NeuroSim-Op, each individual represents a particular TDMA schedule for M sensors. Fig. 12 shows the structure of an example individual, which represents the schedule shown in Fig. 11. In this example, the first sensor, s_1 , fires its neighboring neuron, n_4 , in the first time slot T_1 . Similarly, n_2 and n_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. 12. Individual Representation

5 Case Study

This section evaluates NeuroSim with a case study that generates simulated neuronal networks, operates Neuronal TDMA in the neuronal networks and examines its performance characteristics with an EMOA, NSGA-II [22].

This case study uses two simulated neuronal networks shown in Figs. 8 and 9. The user defines the two neuronal networks with NeuroSim-Gen and NeuroSim-Viz. NeuroSim-Viz verifies the structures of the two networks and passes XML-encoded data to NeuroSim-Op, which seeks the optimal TDMA schedules with NSGA-II. NSGA-II is 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. 12).

Fig. 13 shows how individuals increase the union of the hypervolumes that they dominate in the objective space as the number of generations grows. The hypervolume metric quantifies the optimality and diversity of individuals [23]. A higher hypervolume means that individuals are closer to the Pareto-optimal front and more diverse in the objective space. As Fig. 13 shows, NeuroSim-Op increases its hypervolume measure in the first 10 generations and converges around the 25th and 40th generation in 20-neuron and 40-neuron networks, respectively.

Table 1. NSGA-II Configurations

| Parameter | Value |
|--|-------|
| Max. # of generations in each simulation | 100 |
| Population size | 100 |
| Crossover rate | 0.9 |
| Mutation rate | $1/Q$ |

At the last generation, all individuals are non-dominated in the population. This demonstrates that NeuroSim-Op allows individuals to efficiently evolve and improve their quality and diversity.

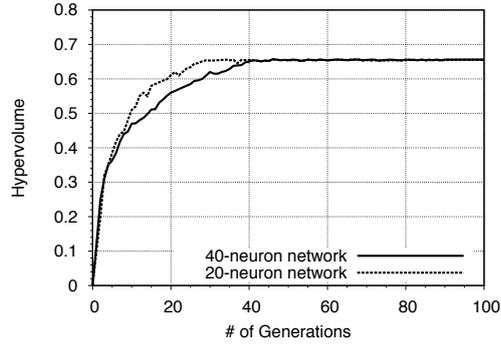
**Fig. 13.** Hypervolume

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

Table 2. Objective Function Values

| | f_Y | f_F | f_D |
|-------------------|--------------|-------------|--------------|
| 20-neuron network | 8.97 (1.1) | 0.08 (0.10) | 15.17 (3.44) |
| 40-neuron network | 16.11 (3.22) | 0.08 (0.10) | 29.78 (6.91) |

Fig. 13 and Table 2 demonstrate that NeuroSim-Op performs scheduling optimization efficiently and effectively for simulated neuronal networks. In summary, this case study confirms that NeuroSim successfully aids communication protocol designers to define/verify neuron-based nanonetworks and optimize TDMA scheduling on simulated neuron-based nanonetworks.

6 Conclusions

This paper proposes and evaluates a simulator called NeuroSim for neuron-based intrabody sensor networks. NeuroSim integrates a neuronal network generator,

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 simulated neuronal networks and examines its performance characteristics with evolutionary multiobjective optimization algorithms.

References

1. Suda, T., Moore, M., Nakano, T., Egashira, R., Enomoto, A.: Exploratory research on molecular communication between nanomachines. In: Proc. ACM Genetic and Evol. Computat. Conference. (2005)
2. Freitas, R.A.: Current status of nanomedicine and medical nanorobotics. *J. Computational and Theoretical Nanoscience* **2**(1) (2005)
3. Atakan, B., Balasubramaniam, S., Akan, O.B.: Body area nanonetworks with molecular communications in nanomedicine. *IEEE Comm. Magazine* (January 2012)
4. Gine, L.P., Akyildiz, I.F.: Molecular communication options for long range nanonetworks. *Computer Networks* **53** (2009)
5. Balasubramaniam, S., Boyle, N.T., Della-Chiesa, A., Walsh, F., Mardinoglu, A., Botvich, D., Prina-Mello, A.: Development of artificial neuronal networks for molecular communication. *Nano Communication Networks* **2**(2-3) (2011)
6. Suzuki, J., Balasubramaniam, S., Prina-Mello, A.: Multiobjective TDMA optimization for neuron-based molecular communication. In: Proc. of the 7th Int'l Conference on Body Area Networks. (September 2012)
7. Suzuki, J., Balasubramaniam, S.: Networking and scheduling in neuron-based molecular communication. In: Proc. of NSF Workshop on Biological Computations and Communications. (November 2012)
8. Moore, M., Enomoto, A., Nakano, T., Suda, T., Kayasuda, A., Kojima, H., Sakakibara, H., Oiwa, K.: Simulation of a molecular motor based communication network. In: Proc. Intl Conference on Bio-Inspired Models of Network, Information and Computing Systems. (2006)
9. Gul, E., Atakan, B., Akan, O.B.: NanoNS: A nanoscale network simulator framework for molecular communications. *Nano Communication Networks* **1**(2) (2010)
10. Llatser, I., Pascual, I., Garralda, N., Cabellos-Aparicio, A., Alarcon, E.: N3Sim: A simulation framework for diffusion-based molecular communication. In: *IEEE TC on Simulation*, 8. (2011)
11. Toth, A., Banky, D., Grolmusz, V.: 3-D brownian motion simulator for high-sensitivity nanobiotechnological applications. *IEEE Trans. NanoBioscience* **10**(4) (2011)
12. Akkaya, A., Tugcu, T.: dMCS: Distributed molecular communication simulator. In: Proc. Intl Conference on Body Area Networks. (2013)
13. L. Felicetti, M.F., Reali, G.: A simulation tool for nanoscale biological networks. *Nano Communication Networks* **3**(1) (2012)
14. Stuart, G., Schiller, J., Sakmann, B.: Action potential initiation and propagation in rat neocortical pyramidal neurons. *Journal of Physiology* **505.3** (1997)
15. Nguyen-Vu, T.D., Chen, H., Cassell, A.M., Andrews, R.J., Meyyappan, M., Li, J.: Vertically aligned carbon nanofiber architecture as a multifunctional 3-D neural electrical interface. *IEEE Trans. Biomed. Eng.* **54**(6) (2007)

16. Morin, F., Nishimura, N., Griscomb, L., LePioufle, B., Fujita, H., Takamura, Y., Tamiya, E.: Constraining the connectivity of neuronal networks cultured on microelectrode arrays with microfluidic techniques: A step towards neuron-based functional chips. *Biosensors and Bioelectronics* **21** (2006)
17. Wyart, C., Ybert, C., Bourdieu, L., Herr, C., Prinz, C., Chatenay, D.: Constrained synaptic connectivity in functional mammalian neuronal networks grown on patterned surfaces. *J. Neurosci. Methods* **117**(2) (2002)
18. Jun, S.B., Hynd, M.R., Dowell-Mesfin, N., Smith, K.L., Turner, J.N., Shain, W., Kima, S.J.: Low-density neuronal networks cultured using patterned poly-L-lysine on microelectrode arrays. *Journal of Neurosci. Methods* **160**(2) (2007)
19. Grossman, N., Nikolic, K., Degenaar, P.: The neurophotonic interface: stimulating neurons with light. *The Neuromorphic Engineer* (2008)
20. Luczak, A.: Measuring neuronal branching patterns using model-based approach. *Front. Comput. Neurosci.* **4**(135) (2010)
21. Srinivas, N., Deb, K.: Multiobjective function optimization using nondominated sorting genetic algorithms. *Evol. Computat.* **2**(3) (1995)
22. Deb, K., Agrawal, S., Pratab, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: *Proc. Conf. Parallel Problem Solving from Nature.* (2000)
23. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms: A comparative study. In: *Proc. Int'l Conf. on Parallel Problem Solving from Nature.* (1998)