

Hierarchical winner-take-all particle swarm optimization social network for neural model fitting

Brandon S. Coventry¹ · Aravindakshan Parthasarathy¹ · Alexandra L. Sommer¹ · Edward L. Bartlett²

Received: 19 April 2016 / Accepted: 14 September 2016 / Published online: 10 October 2016
© Springer Science+Business Media New York 2016

Abstract Particle swarm optimization (PSO) has gained widespread use as a general mathematical programming paradigm and seen use in a wide variety of optimization and machine learning problems. In this work, we introduce a new variant on the PSO social network and apply this method to the inverse problem of input parameter selection from recorded auditory neuron tuning curves. The topology of a PSO social network is a major contributor to optimization success. Here we propose a new social network which draws influence from winner-take-all coding found in visual cortical neurons. We show that the winner-take-all network performs exceptionally well on optimization problems with greater than 5 dimensions and runs at a lower iteration count as compared to other PSO topologies. Finally we show that

this variant of PSO is able to recreate auditory frequency tuning curves and modulation transfer functions, making it a potentially useful tool for computational neuroscience models.

Keywords Biological neural networks · Evolutionary computation · Model optimization · Particle swarm optimization

1 Introduction

Computational modeling at all levels of abstraction can be a powerful tool to make experimentally falsifiable hypotheses and guide experimental design. Furthermore, network models can make inferences about neural circuits that are difficult or impossible to test in physiology. However, one of the major issues incurred in modeling is the tuning and fitting of model parameters to physiological data. Typically, models fit general classes of physiological observations rather than having the ability to fit individual or small sets of observations.

In order to capture biophysically relevant neural behaviors, models often have to encapsulate a wide variety of parameters or include computationally costly Hodgkin-Huxley modeling. Particle swarm optimization (PSO) is a swarm intelligence optimization paradigm that models the schooling of fish or flocking of birds to solve complex problems (Kennedy and Eberhart 1995). PSO was initially developed by Kennedy and Eberhart as a nonlinear optimization solver which models the emergent solving problem capabilities of swarms of relatively simple agents. As its operation is independent of the mathematical program cost function, it has found applications in far reaching

Action Editor: S.A. Shamma

This work was supported by the National Institutes of Health under grant DC-011580

✉ Edward L. Bartlett
ebartle@purdue.edu

Brandon S. Coventry
bcovent@purdue.edu

Aravindakshan Parthasarathy
apartha@purdue.edu

Alexandra L. Sommer
smit1681@purdue.edu

¹ Weldon School of Biomedical Engineering,
Purdue University, Purdue, USA

² Weldon School of Biomedical Engineering
and the Department of Biological Sciences,
Purdue University, Purdue, USA

fields such as the design of optimal power systems (Kim et al. 2011) or the training of neural networks (Salerno 1997; Zhang and Shao 2000). PSO has also been extended to solving multi-objective optimization problems (Hu and Eberhart 2002; Coello Coello and Lechuga 2002).

Particle swarm optimization is a derivative-free meta-heuristic which does not require explicit knowledge of underlying system dynamics to solve mathematical programming problems. Classical optimization methods, such as gradient descent and linear programming, utilize problem derivatives to quickly converge to minima or maxima. However, if this information is not easily available or is too computationally costly to calculate iteratively, such as those found in Hodgkin-Huxley based neural models, these classical models fail. PSO can be applied to problems where no explicit derivative exists. PSO is also advantageous to other meta-heuristics such as genetic algorithms (GA) when calculation of the mathematical problem is computationally expensive because PSOs generally converge in fewer iterations than GAs (Elbeltagi et al. 2005). GAs also require the problem to be recast into a binary optimization problem, while PSO in general does not require modification of the problem. Finally, each iteration of PSO is computationally cheap, only requiring vector addition and scalar multiplication to update its parameters, easing the computational load of optimization.

The power of PSO arises from the emergence of intelligent problem solving behavior in the interactions of simple agents in social networks. Adjustments to the PSO algorithm typically arise in the mathematical representation of agent interactions or social network topologies, such as inertial weight constraints (Shi and Eberhart 1998; Poli et al. 2007). Metaheuristics, PSO specifically, have an inherent tradeoff between swarming and convergence behaviors (Blum and Roli 2003; Kennedy and Mendes 2002). As these are population metrics, they are largely controlled by the architecture of the social network. Too little swarming may lead to the swarm prematurely converging to an incorrect local minimum, while too much may become computationally burdensome and may prevent convergence on the optimal solution (Kennedy 1999). Therefore, the study and design of social networks is of prime importance to PSO.

Swarm social networks can largely be thought of as a continuum across two fundamental swarm types: the fully connected and ring topologies. The ring topology (Fig. 1 right) only transmits information between its nearest neighbors. Information therefore takes a relatively long time to traverse across all agents of the network (Kennedy 1999). This may allow for a more thorough searching of the solution space. Conversely, the fully connected topology can be thought of as a fully connected graph in which every agent has instantaneous access to information from every other agent (Fig. 1 left). This offers quick convergence

but may cause the swarm to become trapped in local minima. Most other social networks draw from aspects from the two extremes. The four-corners network, for example, (Fig. 1 bottom) is fully connected locally but information is bottle-necked by sparse connectivity of the four networks.

Models of central nervous system neurons and networks often run into the “curse of dimensionality”, in which even relatively simple optimization problems become difficult to solve when the number of input variables is high. Furthermore, in most models, general classes of response profiles are modeled, rather than being able to model individual response profiles. Particle swarm optimization has been used in many computational biology models, including fitting somatosensory responses to chirp functions (Vayrynen et al. 2016) and training recurrent neural networks which model gene regulatory networks (Kentzoglanakis and Poole 2012). Van Geit et al. (2007) developed an optimization program which allows for the optimization of neuron model parameters to fit time dependent traces. While this is advantageous for tuning models to fit single neuron spike trains, it does not lend itself to the fitting of tuning where model response is dependent on frequency or another non-time based measure.

In this work we present a new particle swarm social network which draws influence from small world networks found in the visual cortex of the central nervous system.

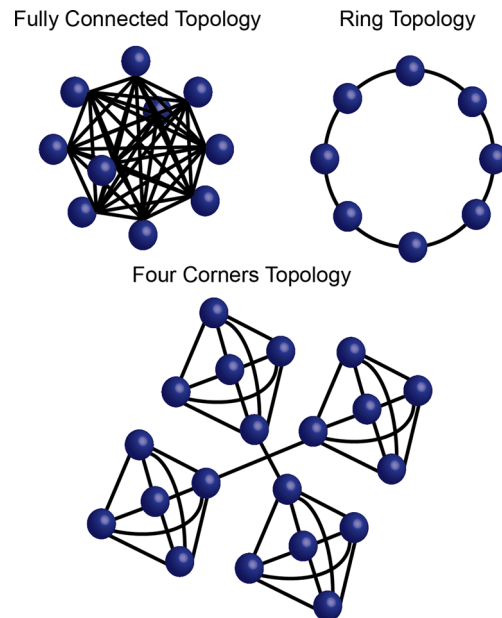


Fig. 1 The continuum of social networks. Top left: The fully-connected topology is a fully-connected graph where each agent is directly connected to every other agent. Top right: The ring topology slows down information transfer by only allowing connections between an agents nearest neighbors. Bottom: Many social networks draw from aspects of both fully-connected and ring topologies. The four corners social network is fully-connected locally but sparsely connected between neighborhoods (Mendes et al. 2004)

(Oleksiak et al. 2011, Lampl et al. 2004, Van de Heuvel and Sporns 2011). Specifically, winner-take-all particle swarm optimization (WTAPSO) is a dynamic, hierarchical topology that utilizes winner-take-all coding strategies found in the visual system which shows improved performance on high-dimensional problems as compared to traditional particle swarm networks (Fig. 2). We begin with a derivation of WTAPSO, and demonstrate that WTAPSO. We demonstrate that WTAPSO is able to fit a wide variety of neural models by showing fits of physiological data from a biophysically accurate computational model of an inferior colliculus (IC) neuron.

2 Methods

Particle swarm optimization is a metaheuristic, derivative free optimization paradigm which models flocks of birds or the schooling of fish to solve mathematical problems (Kennedy and Eberhart 1995; Blum and Roli 2003). Individual agents are relatively simple with complex behavior emerging from carefully designed social networks. We begin with a treatment of agent interactions in canonical PSO, before moving to the formulation of WTAPSO.

2.1 Canonical Particle Swarm Optimization

While many updates schemes have been derived for PSO, for this work we will utilize the constriction coefficient method (Poli et al. 2007, VandenBergh and Engelbrecht 2004, Clerc and Kennedy 2002) as it is a widely used update method and insures swarm stability with appropriate parameters selected. In short, updates for the ring topology are calculated as follows:

$$v(i+1) = \chi(v(i) + U(0, \phi_1) * (p_i - x_i) + U(0, \phi_2) * (p_g - x_i)) \quad (1)$$

$$x(i+1) = v(i+1) + x(i) \quad (2)$$

where v corresponds to an agents velocity, x to its position, and U is a uniform random number drawn between 0 and $\phi_{1,2}$. For this application, $\phi_1 = \phi_2$. The constriction coefficient χ is defined as Poli et al. (2007)

$$\chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \quad (3)$$

where $\phi = \phi_1 + \phi_2 = 4.1$. In the case of the ring social network, p_i is the current agent's best position, p_g is the best position of the best agent in the neighborhood, and x_i is the agent's current position.

The fully-informed PSO can be formulated by allowing each agent to provide the agent under update with its best past performance (Mendes et al. 2004).

$$v(i+1) = \chi(v(i) + \frac{1}{K_i} \sum_{n=1}^{K_i} U(0, \phi) * (p_{bn} - x_i)) \quad (4)$$

$$x(i+1) = v(i+1) + x(i) \quad (5)$$

Here p_{bn} is the best position of the n^{th} neighbor. Sociometrically, this network draws influence from each of its neighbors irrespective of neighbor performance. This network has shown good performance in some cases, but may suffer from premature convergence to local minima.

Many other social networks have been developed to compromise between the two extremes of social network design, namely the ring and fully connected social networks. Here we develop a biophysically inspired network which maintains the relative simplicity of static social networks while performing well on benchmark optimization problems.

2.2 Winner-take-all particle swarm optimization

Though PSO is inspired by swarming biological systems, social networks presented for particle swarm optimization are not often biophysically inspired. Here we present a new social network which takes inspiration from winner-take-all (WTA) neural coding in visual cortex. Seminal work in graph theory has shown that small world, high information throughput networks emerge from normal ring topologies after a relatively small number of non-neighbor connections are made (Watts and Strogatz 1998; Collins and Chow 1998). These small world networks are of particular interest in the central nervous system. Specifically, previous studies have shown the emergence of hierarchical structures and points of influence in connection heavy areas of the human connectome (Van den Heuvel and Sporns 2011). In the realm of PSO, Kennedy has shown that PSO problem solving behavior is topology dependent and exhibits small world dependence (Kennedy 1999). Interestingly, full small world inclusion does not necessarily facilitate more robust problem solving behavior, and may even be detrimental in comparison to a more sparsely connected network (Mendes et al. 2004). Indeed, our preliminary studies have indicated that simply creating a fully informed network typically exhibited weaker problem solving ability than a ring social network (data not shown). Therefore, inclusion of small world shortcuts into swarm topologies must be carefully designed and evaluated.

Drawing from biological neural networks, we propose a new social network for particle swarm optimization. Figure 2 displays the overall social network architecture. The social network is built around a dynamic hierarchical

structure that allows for performance based promotion among agents. A number of networks in visual and parietal cortex exploit winner-take-all (WTA) coding schemes (Oleksiak et al. 2011; Lampl et al. 2004). In WTA coding, input neurons compete for output neuron activation, with the response of the output cell most closely resembling the firing rate and tuning characteristics of the strongest input. Physiologically, WTA rules are thought to be involved in visual attention and saliency detection throughout the visual system, including object orientation and direction (Itti and Koch 2001; Salzman and Newsome 1994). Visual cortical regions also exhibits small world connectivity (Shi et al. 2015). The social network is built around a dynamic hierarchical structure that allows for performance based promotion among agents. First, a global leader is selected by polling the swarm for the agent who is performing best on the fitness function. Agents are then grouped into a lower tier of local neighborhoods. From the neighborhoods, local

leaders are selected based upon which agent is performing best on the fitness function from the neighborhood. The hierarchical structure shares characteristics of information transmission seen in ring networks by limiting the speed of update influence between any two agents. The graph analysis software Gephi (Bastian et al. 2009) was used to derive relevant static graph measures for each social network structure (Table 1). Interestingly, the average path length, a measure of the distance information travels along a graph, is shorter in WTAPSO than the ring or dynamic ring topologies, but longer than that in the four corners social network. The cluster coefficient, a measure of the amount of grouping, displays the same characteristic, with WTAPSO being more clustered than the ring networks, but less so than the four-corners network. These however, while true in the static sense, will be fundamentally altered in dynamic networks when parameters change during updates as connections are dynamically broken and reassembled.

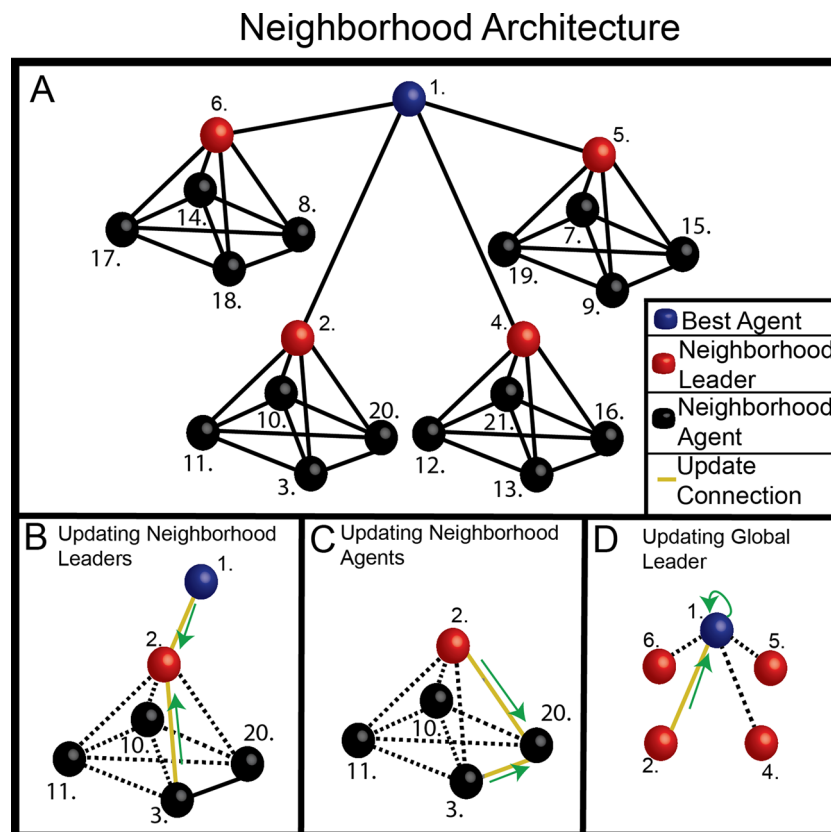


Fig. 2 Hierarchical architecture for the winner-take-all particle swarm social network. **A.)** The overall topology of the WTAPSO social network. The global leader is shown as a blue sphere, while local neighborhood leaders and agents are shown as red and black spheres respectively. Numbers near agents represent a sample of agent placement in the swarm. By definition, the global leader is always 1, while local leaders represent the best performing agents of a given neighborhood. Once the best performing local agent is determined, update connections are broken, represented by dotted black lines, so that only good influences are updating the target agent. **B.)** A sample update of

a neighborhood leader. The neighborhoods are disconnected at update and only the best two performing agents are allowed to update the neighborhood leader. By its rank, the global leader always updates the neighborhood leader. **C.)** Updating a neighborhood agent is very similar to updating a neighborhood leader. Only the best two best performing agents are allowed to update the target agent. By virtue of its position, the local leader will always update the target agent, while the next best performing local agent acts as the other update agent. **D.)** The global leader receives update information from its own previous best performance and the best performing local leader

Table 1 Social network parameters

| Network | Ave. Path length | Diameter | Modularity | Ave. Clustering coefficient |
|--------------|------------------|----------|------------|-----------------------------|
| WTAPSO | 2.971 | 4 | 0.664 | 0.584 |
| Ring | 5.5 | 10 | 0.56 | 0 |
| Dynamic ring | 5.5 | 10 | 0.56 | 0 |
| Four | 2.263 | 3 | 0.62 | 0.76 |

In WTAPSO, updates between any two agents follow a modified winner-take-all rule where agents compete for influence on the agent under update. To update a neighborhood agent (Fig. 2C), the neighborhood is polled for the two best performing agents with the agent under update excluded from the poll. Since the neighborhood leader is by definition the best performer of the neighborhood, it is included as one of the updating agents. The other updater is then the local agent who is currently performing the best on the fitness function. At this point, the neighborhood becomes temporarily disconnected and the agent under update is only updated by the two best performing agents. This works to remove any bad influence from updating a target agent. Quantitatively, updates for neighborhood agents are as follows (Poli et al. 2007):

$$v(i+1) = \chi(v(i) + U(0, \phi_1) * (p_n - x_i) + U(0, \phi_2) * (p_l - x_i)) \quad (6)$$

$$x(i+1) = v(i+1) + x(i) \quad (7)$$

where p_n , p_l correspond to the best neighbor position and best neighborhood leader positions respectively. Updates of neighborhood leaders draw influence from their local neighborhood as well as the global leader (Fig. 2B). Like the agent update, the neighborhood leader's neighborhood is polled for the best performing agent, excluding the neighborhood leader. Neighborhood leaders will always draw influence from the global leader as the global leader by definition is the best performing agent of the swarm. Similar to neighborhood agents, updates are made as follows:

$$v(i+1) = \chi(v(i) + U(0, \phi_1) * (p_n - x_i) + U(0, \phi_2) * (p_g - x_i)) \quad (8)$$

$$x(i+1) = v(i+1) + x(i) \quad (9)$$

where p_n corresponds to the best performing neighborhood agent and p_g corresponds to the global leader. Finally, the global leader draws influence from the best performing of the local leaders as well as its past best performance

(Fig. 2D). This is to ensure continued progress towards viable solutions. Updates are given by:

$$v(i+1) = \chi(v(i) + U(0, \phi_1) * (p_l - x_i) + U(0, \phi_2) * (p_g - x_i)) \quad (10)$$

$$x(i+1) = v(i+1) + x(i) \quad (11)$$

where p_g corresponds to the global leader's best past performance and p_l corresponds to the best performing local leader. After updates are complete, agents are re-evaluated against the fitness function and positions within the swarm are updated. If at any point an agent, irrespective of its place in the social network, performs better than the global leader on the fitness function, it is promoted to the global leader position. Neighborhoods are then reformed and local leaders elected.

2.3 Methods of testing optimization algorithms

In practical optimization, metrics such as algorithm performance and time costs are very important. For example, algorithms that have better problem solving ability may not be useful if run times are exceptionally long. For this study, we tested WTAPSO's ability to solve high dimensional problems, its problem solving ability across a variety of problems, and its runtime and compared these against commonly used social networks designed in-house and in software toolboxes.

For control PSO groups, we utilize the common ring network, a dynamic variant of the ring network which reshuffles agent positions in the ring after each update, and the four corners social network (Fig. 1). Updates for these networks followed a static constriction coefficient method discussed above. We also utilized the Common and Trelea type 2 networks found in a pre-designed Matlab Particle Swarm Optimization toolbox (Birge 2006).

Next, we aim to test WTAPSO performance across dimensions compared to other social networks. For this experiment, the classic test optimization Rosenbrocks function (Rosenbrock 1960) was utilized and spanned across 2 to 10 dimensions. Rosenbrocks function in arbitrary

dimension is given by (Rosenbrock 1960, Shang and Qiu 2006):

$$f(x) = \sum_{i=1}^n [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2] \quad (12)$$

and has a minimal value of 0 when all inputs are identically 1. Significance was assessed via a multi-way ANOVA followed by post hoc Tukey Honest Significant Difference (HSD) tests between groups. Points were deemed significant for $p \leq 0.05$. Groups were across social network type and problem dimension. Each optimization problem was repeated 500 times.

To see how WTAPSO performs against a diversity of problems, we ran each social network against a conglomerate of test optimization functions. Their known global minimums are shown in Table 2. In this test, we used 10 dimensional Ackley, Griewank, Rastrigin, Sphere, and Styblinsky-Tang problems, 30 and 100 dimensional Rosenbrock function, 2 dimensional Schaffer $N.2$ and $N.4$ functions, and 10 and 20 dimensional Styblinsky-Tang functions. These functions represent a wide variety of classical test optimization functions at many different dimensions and degrees of difficulty. The less common Styblinsky-Tang function is used to demonstrate that the solutions were not preprogrammed into the algorithm, as its global minimum is a function of problem dimension. A subset of these functions was tested with the Common and Trelea networks found in the Matlab Particle Swarm Optimization Toolbox. These were the functions pre-built into the toolbox. Again, each problem was run 500 times. Initial conditions were set such that initial position was set as a uniform random variable between the function domain and velocities set to 0 as previous studies have shown better swarm solutions in this condition (VandenBergh and Engelbrecht 2004).

Table 2 Frequency tuning optimization parameters

| Frequency | AM Tuning |
|-----------------------------------|-----------------------------------|
| AMPA Conductance (%) | AMPA Conductance (%) |
| NMDA Conductance (%) | NMDA Conductance (%) |
| GABA _A Conductance (%) | GABA _A Conductance (%) |
| Membrane Potential (mV) | Membrane Potential (mV) |
| Excitatory input Q10 factor | EX Rate Beta params |
| Excitatory input firing rate | IN Rate Beta params |
| Inhibitory input Q10 factor | EX VS Beta params |
| Inhibitory input firing rate | IN VS Beta params |
| Inhibitory input CF (Hz) | EX,IN total rate |
| Inhibitory Lag | EX,IN total VS |
| Excitatory input PSTH factors | |
| Inhibitory input PSTH factors | |

To test the speed of algorithm completion, we utilize the Rosenbrock function across 2,5,8,10, and 30 dimensions. The swarm was allowed to run until a function value of 100 or less was reached, which is considered a good fit for the Rosenbrock function (Mendes et al. 2004). However, if the swarm reaches 10000 iterations, the solution is coded as not found and the program stopped. After the 10000 iteration mark, it is unlikely the swarm will find the target solution in a reasonable time frame. For this study, only the ring, dynamic ring, and four corners social network were used. As the PSO toolbox was self-contained, we could not test run time in this manner. Tests were repeated 500 times for each problem. All graphs are plotted as means with standard deviations.

2.4 Neuron model tests

To test the ability of WTAPSO to fit neural models, a previously published biophysical model of an inferior colliculus neuron with NEURON software (Carnevale and Hines 2006) was modified and utilized (Rabang et al. 2012). The inferior colliculus is a major point of convergence of the central auditory system, integrating excitatory inputs from dorsal and ventral cochlear nucleus and lateral and medial superior olive and inhibitory inputs from dorsal and ventral nucleus of the lateral lemniscus and superior parabrachial complex (Kelly and Caspary 2005; Saldaña et al. 2009). It is not well understood, however, how the confluence of these diverse inputs form neural responses to simple and complex stimuli. Computational models are a useful tool for exploring integration patterns of the IC. We demonstrate in this work how physiological recordings and computational models can be used to solve the inherent inverse problem of input parameters to response generation. Briefly, this biophysical conductance based neuron model was constructed using NEURON software (Carnevale and Hines 2006), modified from a previous IC modeling study (Rabang et al. 2012). Real physiological spiking data were evoked via presentation of tonal and amplitude-modulated stimuli. To demonstrate the diverse range of neural problems WTAPSO can solve, frequency tuning curves (FTC) and temporal modulation transfer functions (tMTF) from pure sinusoidal and sinusoidal amplitude modulated tones respectively were modeled. In each case, five repetitions of each stimuli were modeled.

Inputs in the frequency tuning models consisted of peristimulus time histograms whose input shapes were allowed to dynamically vary according to swarm updates. Once input PSTHs were formed, a curve-fit was made to approximate an input probability density function. Cumulative density functions were then formed. Input spike times were drawn from excitatory and inhibitory input CDFs

via an inverse transform sampling method (Devroye 1986). Briefly, a value U was generated from a uniform distribution. Then, a random variable X was found satisfying:

$$X = F^{-1}(U) \quad (13)$$

with $F(*)$ denoting the CDF which a sample is to be drawn from. The random variable X is then a sample from CDF $F(*)$. It should be noted that while this is a traditional method of sampling from an arbitrary probability distribution, slight aliasing is inherent in the method. The model feature space can be defined as necessary to capture relevant aspects of the model. Table 3 displays features for model reconstruction in AM and frequency tuning models. Finally, as PSO performance modulates across fitness functions (Table 3), careful selection of an appropriate function is important to algorithm performance. In order to preserve shape and absolute rates endemic to frequency tuning curves, a fitness function which accounts for both the correlation between experimental and model curves as well as their relative distance was used (Tiilikainen et al. 2007) as shown:

$$F = 1 - (1 + [RMSE(x_c, x_m)[1 + r(x_c, x_m)]]^{-3})^{-1} \quad (14)$$

where RMSE corresponds to the root mean square error, r is the correlation coefficient, and x_c and x_m are the experimental and model curves respectively. This function is bounded between 0 and 1, with values closer to zero corresponding to a better fit. For AM rate tuning, inputs consisted of two-parameter Beta-functions which approximate all-pass, band-pass, and low-pass inputs dependent on parameter value. A mean-square error metric was utilized for AM tuning responses. Spontaneous rates were modeled by an

Ornstein-Uhlenbeck process (Destexhe et al. 2001) with mean rates modeled from spontaneous activity recordings in young and aged rats.

Responses were recorded from Fischer-344 rats, similar to Rabang et al. (2012) and Herrmann et al. (2015). Briefly, all surgical procedures used were approved by the Purdue University animal care and use committee (PACUC 06-106). Auditory brainstem responses were recorded from these animals a few days preceding surgery to ensure all animals had hearing thresholds typical for their age and to ensure there were no abnormal auditory pathologies. Surgeries and recordings were performed in a 9'x 9' double-walled acoustic chamber (Industrial Acoustics Corporation). Anesthesia was induced in the animals using a mixture of ketamine (VetaKet, 80mg/kg for young animals and 60 mg/kg for aged) and medetomidine (Dexdomitor, 0.2 mg/kg for young and 0.1 mg/kg for aged animals) administered intra-muscularly via injection. The animals were maintained on oxygen through a manifold. The pulse rate and oxygen saturation were maintained using a pulse-oximeter to ensure normal range. Supplementary doses of anesthesia (20 mg/kg of ketamine, 0.05 mg/kg of medetomidine) were administered intra-muscularly as required to maintain areflexia and a surgical plane of anesthesia. An initial dose of dexamethasone and atropine were administered to reduce swelling. Constant body temperature was maintained via a heating pad (Gaymar) set at 37 degrees C. A central incision was made along midline and calvaria was exposed. A stainless steel headpost was cemented anterior to bregma and secured via bonescrews and a headcap constructed with dental cement. Craniectomy was performed posterior to lambda and 1 mm lateral from midline. The dura matter was kept intact and target structure location was estimated via stereotaxic coordinates and physiological measurements in response to auditory stimuli. Recordings were typically made from the right IC.

Table 3 Social network performance on test optimization functions

| Problem | WTAPSO | Dynamic Ring | Ring | Four |
|------------------|-----------------------------|--------------------------|----------------------|--------------------|
| Ackley10 | 0.15(±0.33) | 1.68(±0.64) | 1.92(±0.66) | 2.81(±0.53) |
| Griewank10 | 0.03(±0.09) | 1.99(±1.53) | 2.61(±2.02) | 6.66(±3.41) |
| Rastrigin10 | 0.90(±4.40) | 28.67(±8.19) | 38.09(±10.48) | 32.98(±6.85) |
| Rosenbrock30 | 0.03(±0.08) | $5.85E^5(±3.89E^5)$ | $8.23E^5(±5.52E^5)$ | $1.3E^6(±7.98E^5)$ |
| Rosenbrock100 | 0.10(±0.25) | $4.77E^6(±1.56E^6)$ | $6.21E^6(±2.06E^6)$ | $8.2E^6(±2.65E^8)$ |
| Schaffer's F_2 | $8.98E^{-13}(±1.78E^{-12})$ | $6.25E^{-6}(±1.4E^{-4})$ | 0(±0) | 0.01(±0.02) |
| Schaffer's F_4 | $0.29(±2.20E^{-6})$ | $0.29(±4.98E^{-16})$ | $0.29(±4.98E^{-16})$ | 0.30(±0.01) |
| Sphere10 | 0.004(±0.005) | 167.12(±197.12) | 189.97(193.97) | 683.56(±416.84) |
| Stybl10 | -365.55(±34.07) | -316.65(±19.77) | -322.35(±20.61) | -260.91(±20.31) |
| Stybl20 | -757.48(±70.44) | -530.73(±31.60) | -542.26(±33.72) | -446.43(±32.02) |

2.5 Neural stimuli

Once the inferior colliculus was located, stimuli were presented free-field, azimuth 0° elevation 0° . Stimuli were generated via Tucker-Davis Technology (TDT) Sig Gen RP. Stimuli were sampled at 97.64kHz and presented through custom written interface in Open-Ex software. For frequency tuning curves, sinusoidal stimuli were 200 ms in length and presented at five repetitions in logarithmic steps between 500 and 40000 Hz presented in random order for each trial. Sound level was set as the minimum sound level that produced a maximum response, typically corresponding to 60-70dB SPL in young animals. Sinusoidal amplitude modulated stimuli were 750 long with modulation frequencies in logarithmic steps between 8 and 1024 Hz.

3 Results

Of paramount importance to the WTAPSO is its collective behavior, performance against a wide variety of mathematical problems, and its viability as a tool for making testable hypotheses for electrophysiology studies.

3.1 WTAPSO behavior

To begin, we demonstrate local and global neighborhood behavior of the WTAPSO algorithm. Figure 3 displays 4 sample paths of WTAPSO solving the 30 dimensional

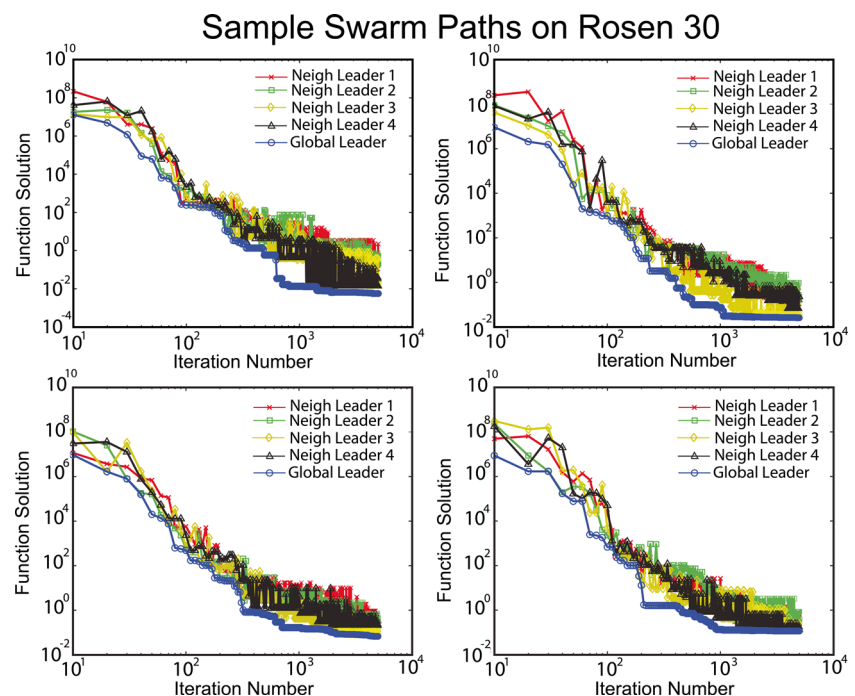
Rosenbrock function. This figure only represents values for the individual neighborhood leaders and does not indicate when leadership changes. Neighborhood leader dynamics show highly oscillatory behavior as the iteration count increases, indicative of the fact that minor changes in input variables from update changes can cause relatively significant changes in the output of the Rosenbrock function. Oscillation behavior might also be due to agents changing rank in the social network. However, we cannot say this for certain as leadership changes were not tracked for this test. Irrespective, the overall trend for each neighborhood leader is towards the minimum. The global leader is always monotonically decreasing towards the minimum value. Notably, there are areas in which the global leader seems to get stuck in local minima, but network dynamics push it out of these minima (Fig. 3 Upper and Lower Right).

3.2 WTAPSO is able to solve high dimensional problems

Our first test compared the performance of WTAPSO on high dimensional problems compared to other social networks. Figure 4 displays the performance of all social networks on the Rosenbrock optimization problem between 2 and 10 dimensions. Significance was assessed by a two-way ANOVA with Tukey Honestly-Significant-Difference (HSD) post tests between groups. Data was log transformed before analysis. Minimal value was truncated at 10^{-30} in order to represent data on a log plot.

In lower dimensions (less than 5), traditional social networks can perform better than WTAPSO. This is not to

Fig. 3 Sample paths for WTAPSO on the Rosenbrock 30 problem. The global leader shows monotonically decreasing function values while the local leaders demonstrate decreasing trends with oscillations about error values, most likely due to the rearranging of the swarm networks. This also demonstrates WTAPSO's ability to move out of local minima until convergence at higher iteration values



Performance of Social Networks Vs Problem Dimension

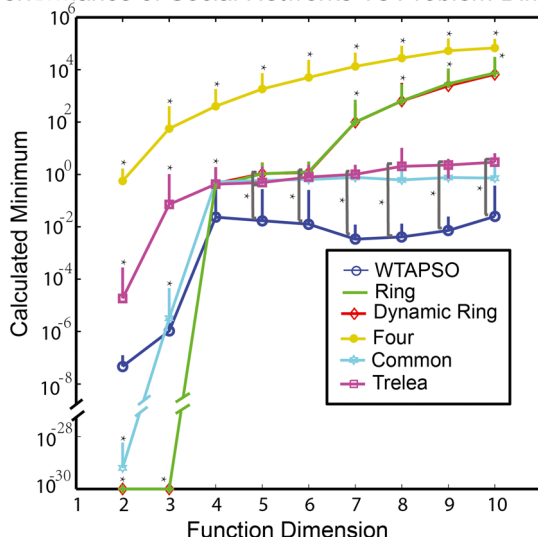


Fig. 4 Social network performance versus dimensionality on the Rosenbrock function. While Canonical ring, dynamic ring, Common and Trelea social networks perform best in low dimensions, the winner take all social network performs on par with the toolbox Matlab toolbox Common network and better than all others. Significance was assessed by a two-way ANOVA with Tukey honestly-significant-difference post tests. * indicates a significant difference between WTAPSO and social network under test

say, however, that the WTAPSO performs poorly. According to stopping conditions presented in Mendes et al. (2004), WTAPSO is providing solutions well within the range of the global minimum. As higher order dimensions are introduced, performance of every social network decreases as expected. After 4 dimensions, WTA PSO starts to perform as well as Matlab PSO toolbox Common network and outperforms every other network. It is interesting to note that many of the social networks grow worse in problem solving ability as dimensionality increases, but WTAPSO (and the toolbox networks) seem to plateau and maintain a stable ability to solve problems. Subsequent experiments show that even in extremely high dimensional problems (30 – 100 dimensions), WTAPSOs problem solving ability does decrease somewhat, but remains much more robust than the other social networks (Tables 3 and 4).

3.3 WTAPSO successfully solves a wide variety of problems

The no-free-lunch theorem (NFL) (Wolpert and Macready 1997) states that given the set of all possible optimization problems, on average no optimization paradigm will perform better than any other paradigm on all possible problems. In other words, there are some problems that will be more difficult for an optimization paradigm to solve. With that in mind, we will show that WTAPSO can solve a wide variety of nonlinear problems that will be of interest to real systems analysis. Table 3 shows the results of WTAPSO, ring, dynamic ring, and four social networks against the 10 dimensional Ackley function, 10 dimensional Griewank problem, 10 dimensional Rastrigin problem, 30 and 100 dimensional Rosenbrock programs, 2 dimensional Schaffer *F2* and *F4* functions, 10 dimensional sphere function, and 10 and 20 dimensional Styblinsky -Tang function. Inputs to each function leading to global minimums can be found in Table 2 and problem formulations are found in the appendix. On the Ackley and Griewank 10 dimensional problems, all social networks got relatively good results, but WTAPSO in both cases averaged closer to the global minimum of 0 on both problems. The 10 dimensional Rastrigin problem proved difficult to solve for the three test social networks to solve, but WTAPSO was able to find values closer to the global minimum of 0. To test the ability of WTAPSO to solve very high dimensional problems, the 30 and 100 dimensional Rosenbrock problems were used. Due to the nonlinear nature of the Rosenbrock problem, even slight variations in the input can cause explosion of the output. The ring and four corners social networks fail to find satisfactory answers in all both dimensional cases. WTAPSO was able to find satisfactory answers even in the very difficult 100 dimensional case. In the two dimensional Schaffer problems, all social networks performed roughly the same, with the dynamics ring social network finding the exact minimum of 0 on the Schaffer *N.2* problem and smaller standard deviations in the Schaffer *N.4* problem.

To show that WTAPSO is able to track changing global minimums, the Styblinsky-Tang function was used, which

Table 4 WTAPSO vs. Toolbox networks

| Problem | WTAPSO | Common | Trelea |
|---------------|---------------|---|---|
| Ackley10 | 0.14(±0.33) | 17.20(±6.79) | 16.16(±7.77) |
| Griewank10 | 0.031(±0.092) | 2.77(±2.34) | 3.03(±2.41) |
| Rastrigin10 | 0.90(±4.40) | 9.67(±5.54) | 10.52(±6.10) |
| Rosenbrock30 | 0.03(±0.08) | 37.02(±39.67) | 12.03(±15.84) |
| Rosenbrock100 | 0.10(±0.25) | 203.79(±56.29) | 714.27(±4.5E ³) |
| Sphere10 | 0.004(±0.01) | 9.49E ⁻⁴⁰ (±1.02E ⁻³⁸) | 1.12E ⁻³⁶ (±3.96E ⁻³⁶) |

has a global minimum that changes with problem dimension, defined by:

$$f(x) = -39.16599 * n \quad (15)$$

In both the 10 and 20 dimensional variants, WTAPSO performed better than the other social networks, demonstrating the ability to track changing global minimums on a problem definition. It should be noted that these topologies used in Mendes et al (2004) do provide better results. However, the update functions and use of information were fundamentally different, owing to the difference in observations and further proving the importance of all aspects of the design of a swarm.

Next we compared the performance of WTAPSO against a previously established Matlab PSO toolbox. Only the subset of test functions available within the toolbox were used for this analysis. Results in Table 4 demonstrate that WTAPSO performs better on most test functions, with the exception of the 10 dimensional sphere problem. This helps to illustrate the no-free-lunch theorem and that there are problems that are easier for some social networks to solve.

3.4 WTAPSO takes fewer iterations to solve problems

Our final test of basic algorithm behavior was to determine whether WTAPSO was able to solve problems faster than other social networks. For this test, to account for differences in CPU time and processing, we utilized iteration count as our time metric. To standardize against problem type, we used only the Rosenbrock problem at 2,5,8,10 and 30 dimensions. Following the convention of Mendes et al (2004), swarms were allowed to run until a function value of 100, corresponding to an acceptable solution, was reached or the number of iterations exceeded 10,000, at which point it is assumed that the swarm would not converge onto an adequate solution. Figure 6 demonstrates that on the 2 dimensional Rosenbrock problem, every swarm was able to reach an adequate solution at a reasonable number of iterations. In 5 dimensions, the ring social networks are able to solve the problem in an adequate time, but the four corners social network can no longer find a solution. As dimensionality increases, only the WTAPSO is able to solve the problem and does so in a remarkably low number of iterations (100 iterations). This test shows that, independent of CPU processing power, WTAPSO is able to find solutions in a lower iteration count than the test social networks which in most cases will translate to lower run times.

3.5 Inferior colliculus measurements

To demonstrate the viability of WTAPSO to solve biophysical neuron model inverse problems, we show response generation from three different auditory feature representations in the IC.

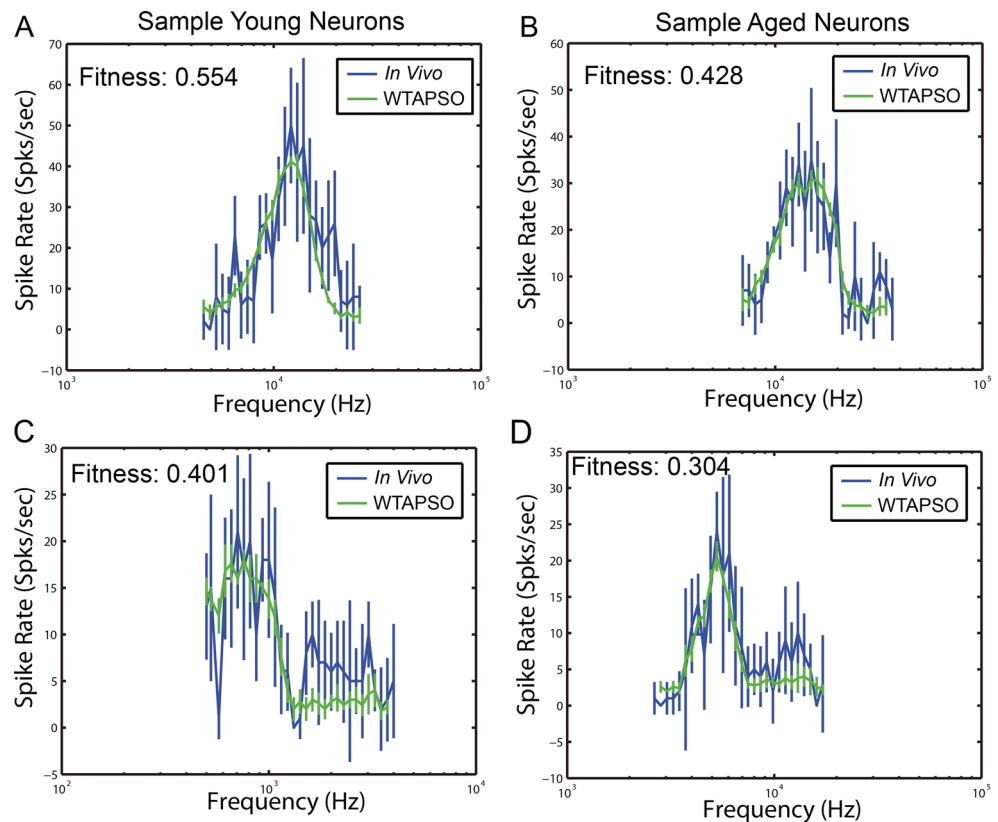
3.5.1 Frequency tuning curve fitting

Neurons throughout the auditory pathway exhibit tonotopic frequency mappings with different spatial regions of auditory regions characterized by neurons exhibiting maximal firing at a specific frequency, known as the characteristic frequency. We demonstrate WTAPSO systematically fitting input parameters and distributions to recreate FTCs recorded *in vivo* (Fig. 5). Figure 5A,D demonstrates reconstruction of typical FTC curves with maximum mean rate at its center frequency. WTAPSO is also able to recreate more challenging tuning curve, such as those with two disjoint peaks in the mean (Fig. 5B). Finally, WTAPSO is able to recreate tuning curves with less well defined shapes, such as in Fig. 5C. While the driven region of the curve is well fit, there is some error at the non-driven region, as spontaneous rate is a fixed parameter not subject to optimization in this version of the model. Table 4 shows the reconstructed parameters for the fits in Fig. 5. Several constraints were put in place to ensure operation in a biophysically relevant region, such as resting membrane potential set between -80 and -50 mV and fully positive conductance values. For the purpose of this work, parameters governing input PSTH values were not constrained. However, careful attention to these should be paid when drawing physiological relevance from fits.

3.5.2 AM-tuning fits

Utilizing the same biophysical IC neuron model, responses to sinusoid-amplitude modulated tones were reconstructed. Figure 6 demonstrates reconstructed modulation transfer functions (MTFs). For these examples, inputs were derived from a beta prior and demonstrate that incomplete knowledge of input parameters can still lead to good fits. Importantly we are able to fit a wide variety of IC neural response classes seen in the IC (Rabang et al. 2012). For these fits, we utilized a mean-square error metric which measures relative distance between the mean of the experimental and model rMTFs, with lower values representing better fits. Sources of error most likely arise due to the choice of the beta function prior which may not fully encompass input distributions. With naïve assumptions about the underlying input distribution, WTAPSO was able to recreate the

Fig. 5 Recreation of *in vivo* FTCs using a biophysical neuron IC model using WTAPSO. Responses were elicited from sinusoid tones of varying frequencies. The fitness function varies between 0 and 1 with 1 being a better fit. Spontaneous responses were not fit, but set to rates typically seen in recordings (data not shown). WTAPSO is able to recreate a wide variety of responses types and shapes seen *in vivo*



rMTFs of individual IC neurons that were representative of the most common rMTF classes, demonstrating the ability to draw inference even in situations with incomplete information.

4 Discussion

4.1 WTAPSO performs best in high dimensional problems

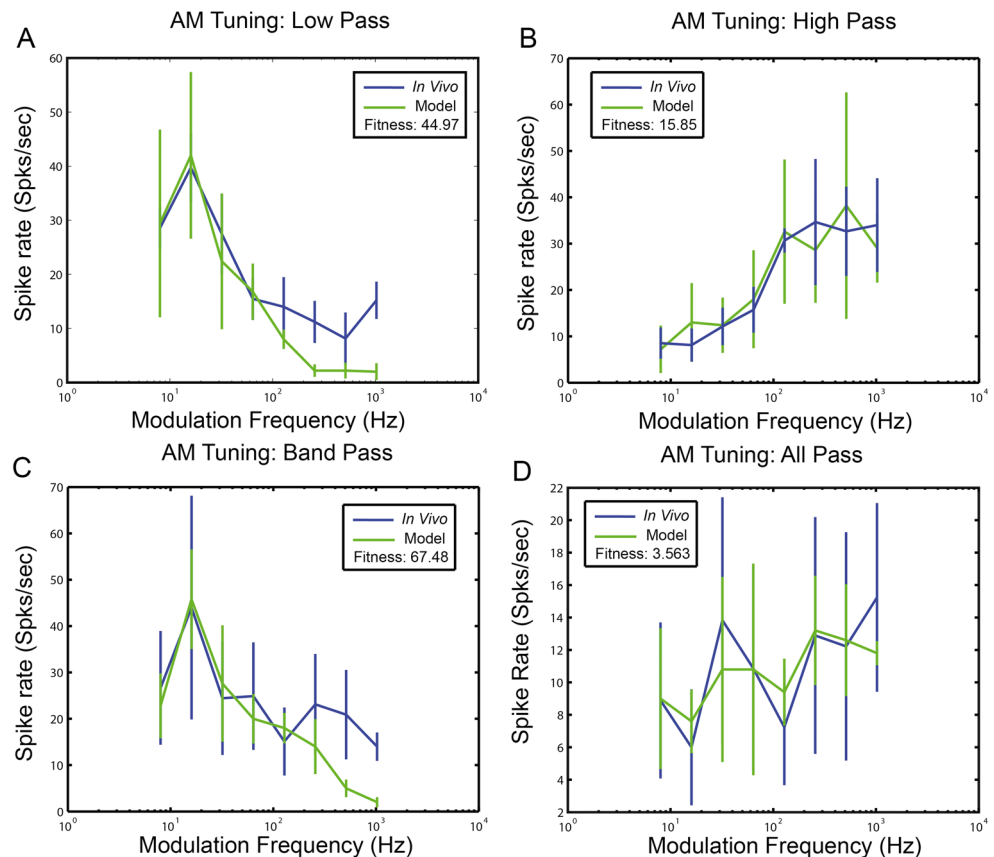
In optimization, many solvers suffer from the curse of dimensionality, where higher dimensional problems become overly difficult to solve. This necessitates the need for specialized high dimensional solvers. Further complicating these issues is the fact that many experimental paradigms may not provide access to explicit derivatives, meaning that classical optimization methods cannot be utilized. Furthermore, computational neuroscience models are often high dimensional in nature. In this work we present a new social network for particle swarm optimization which mimics winner-take-all connectivity found in visual cortex neural networks. We have demonstrated that WTAPSO is able to solve very high dimensional problems, outperforming traditional social networks and publicly available particle swarm

toolboxes. In this study, we used a relatively simple static constrained coefficient method. It is predicted that allowing for a dynamic constriction coefficient will yield better results as step size can be modified based on distance from a solution. It was also seen that individual agents in WTAPSO can get stuck in local minima and cannot be drawn out even with well behaving influences. Depending on problem complexity, there is some wasted processing spent on agents which are no longer able to find solutions. A way to mitigate this may be the inclusion of a grim-reaper routine, which searches for agents that are both far away from the best performing agent and no longer making significant advances in fitness function. This routine will then remove the agents from the swarm, and thus reduce computational load, or reinitialize the agent based on best performing network and global leaders. The latter may also provide the algorithm better influences as one more agent will now be searching a space that is more likely to contain the local minimum than its previous location, but this remains to be tested.

4.2 WTAPSO is relatively simple to implement

Some swarm intelligence methodologies seek to address the problem of dimensionality by subdividing the swarms

Fig. 6 Example rMTFs. Functions were optimized around synaptic conductance values and input beta distribution priors. WTAPSO is able to recreate a wide variety of response classes including low-pass, high-pass, band-pass, and all-pass shapes



into many sub-swarms to search problem sub-spaces and then pool what was learned to find an optimum solution (VandenBergh and Engelbrecht 2004). However, many problems, such as biophysical neural models which require the simultaneous tuning of many model parameters, are computationally demanding, and subdividing swarms into miniswarms would unacceptably blow up computation time. WTAPSO may provide a viable framework for modeling problems such as these because its construction is as simple as defining a new social network and does not require any more computation time than a swarm with comparable number of agents. Furthermore, it is able to find solutions in a fewer number of iterations in comparison with other PSO methods, meaning that complex models may run in faster time. Toolboxes adopting this method may then use ring topologies to solve low dimensional problems with an option for WTAPSO for high dimensional problems. While other particle swarm paradigms, such as comprehensive learning PSO (Liang et al. 2006), attempt to mitigate the amount of bad influences updating an individual agent, WTAPSO does so in a more deterministic fashion and allows for systematic testing of various particle hypotheses. For example, each neighborhood can be initialized to different regions of the solution space and allow the swarm to then run to completion.

4.3 Fitting of physiological response curves

We have shown the ability of WTAPSO to solve the inverse problem of retrieving physiological inputs from neural response tuning curves. While general input distributions can be utilized, as in the Beta distribution in the AM tuning model, physiologic inputs will most likely give rise to more accurate fits. Thus, when making inferences regarding neural function, one must pay special attention to the region of feasible physiologic inputs, both in feature selection and constraints. While this method does fit individual tuning curves, synaptic inputs are almost certainly not unique mappings, i.e., each response curve could be generated by different sets of inputs. However, this method generates inputs which serve as a basis for parameters which can empirically tested in electrophysiology experiments (Table 5). This method is also problem independent: as long as there is a mathematical model of the neuron under test, WTAPSO can be used to fit physiological responses.

4.4 Further considerations

One consequence of the No Free Lunch theorem is that there can be no guarantees on optimal performance on every

Table 5 Frequency tuning synaptic recreations

| Parameter | 13 Y2 | 14 Y10 | 13 A3 | 13 A6 |
|------------------------------------|--------|--------|--------|--------|
| AMPA conductance(%) | 257.79 | 95.76 | 183.33 | 149.90 |
| NMDA conductance(%) | 192.26 | 280.95 | 159.54 | 150 |
| GABA _A conductance(%) | 182.35 | 217.78 | 170.45 | 150 |
| Membrane potential (mV) | −57.37 | −57.88 | −50.86 | −53 |
| EX <i>Q</i> 10 factor | 3.13 | 2.12 | 4.76 | 2.91 |
| EX firing rate (<i>spks/sec</i>) | 129.31 | 60.60 | 80.46 | 115.89 |
| IN <i>Q</i> 10 factor | 3.70 | 3.36 | 7.1 | 5.2 |
| IN firing Rate (<i>spks/sec</i>) | 71.98 | 71.57 | 84.40 | 119.8 |
| Inhibitory CF (<i>kHz</i>) | 12.13 | .71 | 5.28 | 14.93 |

problem an optimization paradigm might encounter. However, in line with Mendes et al. (2004) assertion that the NFL theorem does not negate the need to find better PSO social networks, there is a large subset of interesting problems that are high dimensional. As our data have shown, traditional particle swarm social networks and toolboxes often fail at finding adequate solutions to high dimensional problems. The WTAPSO social network shows improved performance on high dimensional problems. Furthermore, our preliminary studies in using this social network on real-world system models has shown excellent performance in parameter tuning in high dimensional problems with run times that are reasonable on a personal PC. The WTAPSO social network also offers unique abilities to explore solutions spaces of real world problems. For example, the four social networks can be started at vastly different initial conditions that test potential biophysical mechanism hypotheses. The networks swarms can be allowed to run independently to find local optimum solutions then connected to see which physical parameter dominates in a certain operating zone. Finally, the swarm can be recursively defined, incorporating more neighborhoods extending below the lower agents which might contribute to better problem solving behavior.

5 Conclusion

Here we present a new hierarchical particle swarm optimization social network which draws influence from winner-take-all coding found in visual cortical neurons. WTAPSO shows improved performance in high dimensional test optimization problems and may be valuable in parameter tuning problems of physical models. While many PSO social networks are not biophysically inspired, we hope to show that power of drawing influence from real world networks as a way to inspire better problem solving behavior in artificial social networks. Furthermore, WTAPSO serves

as a potentially powerful tool for fitting complex neural responses across sensory modalities.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest

Appendix

The following test optimization functions were used in accessing WTAPSO performance.

Rosenbrock's Function (Rosenbrock 1960; Shang and Qiu 2006)

$$f(x) = \sum_{i=1}^n [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2] \quad (16)$$

Minimum: 0 at $X = (1, 1, \dots, 1)$

Sphere Function (Mikki and Kishk 2008)

$$f(x) = \sum_{i=1}^n x_i^2 \quad (17)$$

Griewank's Function (Griewank 1981)

$$f(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (18)$$

Rastrigin's Function (Mühlenbein et al. 1991)

$$f(x) = A_n + \sum_{i=1}^n [x_i^2 - A * \cos(2\pi x_i)] \quad (19)$$

Ackley's Function (Ackley 1987)

$$f(x) = -20e^{-0.2\sqrt{\frac{1}{d}\sum_{i=1}^d x_i^2}} - e^{\frac{1}{d}\sum_{i=1}^d \cos(2\pi x_i)} + 20 + e^1 \quad (20)$$

Schaffer's F2 (Jamil and Yang 2013)

$$f(x, y) = 0.5 + \frac{\sin^2(x^2 - y^2) - 0.5}{1 + 0.001 * (x^2 + y^2)^2} \quad (21)$$

Schaffer's F4 (Jamil and Yang 2013)

$$f(x, y) = 0.5 + \frac{\cos^2(\sin(|x^2 - y^2|)) - 0.5}{(1 + 0.001(x^2 + y^2))^2} \quad (22)$$

Styblinski and Tang (1990)

$$f(x) = \frac{\sum_{i=1}^n x_i^4 - 16x_i^2 + 5x_i}{2} \quad (23)$$

References

- Ackley, D.H. (1987). *A connectionist machine for genetic hillclimbing*, 1st edn. Boston: Kluwer Academic Publishers.
- Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi an open source software for exploring and manipulating networks. In *International AAAI Conference on Weblogs and Social Media* (pp. 1–2).
- Birge, B. (2006). Particle swarm optimization toolbox. <http://www.mathworks.com/matlabcentral/fileexchange/7506-particle-swarm-optimization-toolbox>.
- Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization Overview and conceptual comparison. *ACM Computing Surveys*, 35(3), 268–308.
- Carnevale, N.T., & Hines, M.L. (2006). *The NEURON Book* Vol. 30: Cambridge University Press.
- Clerc, M., & Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1), 58–73.
- Coello Coello, C., & Lechuga, M. (2002). MOPSO: a proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, (Vol. 2 pp. 1051–1056): IEEE.
- Collins, J.J., & Chow, C.C. (1998). It's a small world. *Nature*, 393(6684), 409–10.
- Destexhe, A., Rudolph, M., Fellous, J.M., & Sejnowski, T.J. (2001). Fluctuating synaptic conductances recreate in vivo-like activity in neocortical neurons. *Neuroscience*, 107(1), 13–24.
- Devroye, L. (1986). *Non-uniform random variate generation*, 1st edn. New York: Springer.
- Elbeltagi, E., Hegazy, T., & Grierson, D. (2005). Comparison among five evolutionary-based optimization algorithms. *Advanced Engineering Informatics*, 19(1), 43–53.
- Griewank, A.O. (1981). Generalized descent for global optimization. *Journal of Optimization Theory and Applications*, 34(1), 11–39.
- Herrmann, B., Parthasarathy, A., Han, E.X., Obleser, J., & Bartlett, E.L. (2015). Sensitivity of rat inferior colliculus neurons to frequency distributions. *Journal of Neurophysiology*, 114(5), 2941–54.
- Hu, X., & Eberhart, R. (2002). Multiobjective optimization using dynamic neighborhood particle swarm optimization. In *Proceedings of the Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)* (2002), (Vol. 2 pp. 1677–1681): IEEE.
- Itti, L., & Koch, C. (2001). Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2(3), 194–203.
- Jamil, M., & Yang, X.S. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2), 150.
- Kelly, J.B., & Caspary, D.M. (2005). Pharmacology of the inferior colliculus, In Winer, J.A., & Schreiner, C.E. (Eds.) *The Inferior Colliculus* (pp. 248–281). New York: Springer.
- Kennedy, J. (1999). Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)* (pp. 1931–1938): IEEE.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, (Vol. 4 pp. 1942–1948).
- Kennedy, J., & Mendes, R. (2002). particle swarm performance. In *Proceedings of the Population structure Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)* (2002), (Vol. 2 pp. 1671–1676): IEEE.
- Kentzoglanakis, K., & Poole, M. (2012). A swarm intelligence framework for reconstructing gene networks Searching for biologically plausible architectures. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(2), 358–371.
- Kim, J.-Y., Mun, K.-J., Kim, H.-S., & Park, J.H. (2011). Optimal power system operation using parallel processing system and PSO algorithm. *International Journal of Electrical Power & Energy Systems*, 33(8), 1457–1461.
- Lampl, I., Ferster, D., Poggio, T., & Riesenhuber, M. (2004). Intracellular measurements of spatial integration and the MAX operation in complex cells of the cat primary visual cortex. *Journal of neurophysiology*, 92(5), 2704–2713.
- Liang, J.J., Qin, A.K., Suganthan, P.N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10(3), 281–295.
- Mendes, R., Kennedy, J., & Neves, J. (2004). The fully informed particle swarm: simpler, Maybe better. *IEEE Transactions on Evolutionary Computation*, 8(3), 204–210.
- Mikki, S.M., & Kishk, A.A. (2008). Particle swarm optimization a physics-based approach. *Synthesis Lectures on Computational Electromagnetics*, 3(1), 1–103.
- Mühlenbein, H., Schomisch, M., & Born, J. (1991). The parallel genetic algorithm as function optimizer. *Parallel Computing*, 17(6–7), 619–632.
- Oleksiak, A., Klink, P.C., Postma, A., Van der Ham, I.J.M., Lankheet, M.J., & Van Wezel, R.J.A. (2011). Spatial summation in macaque parietal area 7a follows a winner-take-all rule. *Journal of neurophysiology*, 105(3), 1150–1158.
- Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence*, 1(1), 33–57.
- Rabang, C.F., Parthasarathy, A., Venkataraman, Y., Fisher, Z.L., Gardner, S.M., & Bartlett, E.L. (2012). A computational model of inferior colliculus responses to amplitude modulated sounds in young and aged rats. *Frontiers in Neural Circuits* 6.
- Rosenbrock, H.H. (1960). An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3), 175–184.
- Saldaña, E., Aparicio, M.A., Fuentes-Santamaría, V., & Berrebi, A.S. (2009). Connections of the superior paraolivary nucleus of the rat: projections to the inferior colliculus. *Neuroscience*, 163(1), 372–387.
- Salerno, J. (1997). Using the particle swarm optimization technique to train a recurrent neural model. In *Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence* (pp. 45–49): IEEE Computer Society.
- Salzman, C., & Newsome, W. (1994). Neural mechanisms for forming a perceptual decision. *Science*, 264(5156), 231–237.
- Shang, Y.-W., & Qiu, Y.-H. (2006). A note on the extended rosenbrock function. *Evolutionary Computation*, 14(1), 119–126.
- Shi, L., Niu, X., & Wan, H. (2015). Effect of the small-world structure on encoding performance in the primary visual cortex: an electrophysiological and modeling analysis. *Journal of Comparative Physiology A*, 201(5), 471–483.
- Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360) (1998), pp. 69–73.
- Styblinski, M., & Tang, T.-S. (1990). Experiments in nonconvex optimization Stochastic approximation with function smoothing and simulated annealing. *Neural Networks*, 3(4), 467–483.
- Tiilikainen, J., Bosund, V., Mattila, M., Hakkarainen, T., Sormunen, J., & Lipsanen, H. (2007). Fitness function and nonunique solutions in x-ray reflectivity curve fitting: crosserror between surface roughness and mass density. *Journal of Physics D: Applied Physics*, 40(14), 4259–4263.

- Van den Heuvel, M.P., & Sporns, O. (2011). Rich-club organization of the human connectome. *Journal of Neuroscience*, 31(44), 15775–15786.
- Van Geit, W., Achard, P., & De Schutter, E. (2007). Neurofitter: a parameter tuning package for a wide range of electrophysiological neuron models. *Frontiers in neuroinformatics* 1.
- VandenBergh, F., & Engelbrecht, A. (2004). A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3), 225–239.
- Vayrynen, E., Nojonen, K., Vipin, A., Yuan, T.X., Al-Nashash, H., Kortelainen, J., & All, A. (2016). Automatic parametrization of somatosensory evoked potentials with chirp modeling. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 4320, c, 1–1.
- Watts, D.J., & Strogatz, S.H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684), 440–2.
- Wolpert, D., & Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82.
- Zhang, C., & Shao, H. (2000). Particle swarm optimisation for evolving artificial neural network. SMC 2000 Conference Proceedings. 2000 IEEE International Conference on Systems, Man and Cybernetics. *Cybernetics Evolving to Systems, Humans, Organizations, and their Complex Interactions (Cat. No.00CH37166)* 4, 2487–2490.