Published in final edited form as:

J Neural Eng. 2017 December ; 14(6): 066013. doi:10.1088/1741-2552/aa8270.

An improved genetic algorithm for designing optimal temporal patterns of neural stimulation

Isaac R. Cassar¹, Nathan D. Titus¹, and Warren M. Grill

Department of Biomedical Engineering, Duke University, Box 90281, Durham, NC 27708-0281, USA

Abstract

Objective—Electrical neuromodulation therapies typically apply constant frequency stimulation, but non-regular temporal patterns of stimulation may be more effective and more efficient. However, the design space for temporal patterns is exceedingly large, and model-based optimization is required for pattern design. We designed and implemented a modified genetic algorithm (GA) intended for design optimal temporal patterns of electrical neuromodulation.

Approach—We tested and modified standard GA methods for application to designing temporal patterns of neural stimulation. We evaluated each modification individually and all modifications collectively by comparing performance to the standard GA across three test functions and two biophysically-based models of neural stimulation.

Main Results—The proposed modifications of the GA significantly improved performance across the test functions and performed best when all were used collectively. The standard GA found patterns that outperformed fixed-frequency, clinically-standard patterns in biophysically-based models of neural stimulation, but the modified GA, in many fewer iterations, consistently converged to higher-scoring, non-regular patterns of stimulation.

Significance—The proposed improvements to standard GA methodology reduced the number of iterations required for convergence and identified superior solutions.

1. Introduction

Neuromodulation via electrical stimulation is a rapidly expanding field, including cochlear stimulation to restore hearing (Wilson and Dorman 2008), retinal stimulation to restore sight (Keserü *et al.* 2012), sacral nerve stimulation to restore bladder function (Sherman and Amundsen 2007), deep brain stimulation for Parkinson's disease (Benabid *et al.* 2009), spinal cord stimulation for pain (Verrills *et al.* 2016), and many others. Neuromodulation devices typically deliver stimulation pulse trains with regular, repeating interpulse intervals and only the frequency, pulse duration, and amplitude are varied. However, non-regular temporal patterns of stimulation may be more effective (Brocker *et al.* 2013), more efficient (Brocker *et al.* 2017), more versatile (Tan *et al.* 2014), and produce fewer side effects (De Ridder *et al.* 2010). For parameters such as pulse duration and amplitude, there are well understood relationships between the parameter and the effect, e.g. increasing the amplitude

¹These authors contributed equally to this work.

increases neural activation (Ramasubbu *et al.* 2013). However, there is no clear basis for selection of temporal patterns of stimulation, which requires either ad hoc selection or rigorous model-based design and evaluation. Stimulation parameters have been optimized using genetic algorithms (Kent and Grill 2013, Brocker *et al.* 2017, Feng *et al.* 2007a, Feng *et al.* 2007b, Wongsarnpigoon and Grill 2010), particle swarm optimization (Peña *et al.* 2017), and artificial neural networks (Giuffrida and Crago 2005). Application-specific modifications to the optimization methodology can improve performance (Trevathan *et al.* 2017, Elhossini *et al.* 2010, Sankar *et al.* 2007), and significant effort has been made in other fields to identify the best algorithm and modifications (Yusup *et al.* 2012). Here we propose and evaluate several novel modifications to improve the convergence speed and accuracy of a genetic algorithm for model-based design of optimized temporal patterns of stimulation.

A genetic algorithm (GA) is a parameter optimization technique inspired by biological evolution that uses the concepts of mating, random mutation, genetic diversity, and natural selection (Man *et al.* 1996, Ashlock 2006, Davis 1991, Goldberg 1989). A GA is most easily described using metaphors from evolution, starting with a group of input vectors called “organisms” that contain independently manipulated components called “genes” which represent modifiable parameters to be optimized by the algorithm. In addition, the GA uses a cost function to determine the performance or “fitness” of each organism in the “population”, and then uses a method of “natural selection” to assign each organism a probability of “mating” proportional to its fitness. During mating, the genes of the two organisms are combined by “crossover” and “mutation” to form a “child” organism. These children are joined by “immigrants”, new organisms with random genetic information, to create the next generation of the population, and the process is repeated. In addition, “elite” organisms with the highest fitness are typically passed onto the next generation without modification. This combination of mating, mutation, immigration, and preservation of elites is collectively referred to as repopulation. The GA performs repopulation and scoring iteratively to search the parameter space and thereby arrive at an optimal solution.

Although GAs can use real numbers as genes, temporal patterns of stimulation are best implemented in a binary representation where each 1 or 0 corresponds to the presence or absence of a stimulation pulse at each time point. Although binary repopulation techniques of parent selection, crossover, and mutation are highly standardized (Man *et al.* 1996, Ashlock 2006, Davis 1991, Goldberg 1989), the performance of this standard GA was surprisingly poor for this application. This motivated our effort to improve GA performance by incorporating principles specific to the design of temporal patterns of stimulation. We introduce five modifications to the standard repopulation method (figure 1) and evaluate their utility on three test functions and two biophysically-based computational models of neural stimulation. The results were compared to the performance of a standard GA designed with established binary repopulation techniques (Ashlock 2006).

2. Methods

2.1. Standard GA

The standard GA was a generational evolutionary algorithm with a modified form of roulette selection (Ashlock 2006). GA parameters were adopted that were successful in our previous

work, but other values could be used. These parameters were kept constant to compare different repopulation strategies. Each trial had 50 organisms and was run for 200 generations. For selection, the fitness values in a population were sorted and then fit to a half-normal distribution, after which standard roulette selection was performed to choose mating parents. The half-normal distribution prevents domination by a few patterns and retains information about scoring. Each pair of parents contributed to a two-point crossover to produce a child in the next generation. Point mutations were then made in 0.5% of each child's genes. Immigrants were also introduced by randomly generating new organisms. Finally, the top 10% of the population in each generation was passed onto the following generation as elites. In total, 60% of each generation were children, 30% immigrants, and 10% elites.

2.2. Modifications to the GA

2.2.1. Pulse Mutation—Point mutations and probabilistic mutations are two common methods for performing mutations on a binary GA (Ashlock 2006). Point mutations are performed by specifying a set number of mutations and then randomly determining the location within the gene for each mutation. Probabilistic mutations are performed by using a probability to determine whether a mutation will occur at each point in the gene; this allows for a variable number of mutations. Both approaches bias solutions towards an equal proportion of ones and zeros (figure 2a). Although they may be useful when this proportion holds no significance, in our application these approaches alter the average pulse repetition frequency of stimulation pulse trains. Relevant temporal patterns of stimulation tend to be characterized by infrequent pulses and relatively long periods without pulses (Miller *et al.* 2016). For example, a standard 100 Hz stimulation pattern with a temporal resolution of 1 ms has a 9:1 ratio of zeros to ones, and using either of the standard mutation methods would result in a bias towards higher frequencies. We modified the standard mutation methods to address this problem.

The Pulse Mutation Method (PMM, figure 2b) uses a preset number of mutations, in this case 0.5% of the length of the pattern, to prevent frequency bias. The PMM randomly chooses whether to add a pulse, remove a pulse, or move a pulse and then randomly chooses the location in the gene to mutate. If the destination in the gene for the moved pulse already contains a pulse, then a different pulse is chosen to move. This method introduces no pulse repetition frequency bias as there is an equal probability of increasing or decreasing the number of pulses (figure 2a).

2.2.2. Competitive Immigrants—Randomly generated organisms – termed immigrants or invaders – are a common feature of GAs to maintain genetic diversity in the population. However, these random immigrants are rarely selected to reproduce in later generations as the average fitness of the population greatly exceeds that of randomly-generated immigrants or their progeny. Therefore, we modified the immigrant method to produce immigrants whose fitness remained competitive across the generations of the GA.

Competitive Immigrants (CI, figure 3) were created by mating a random immigrant with a parent chosen from the population via the standard parent selection method. This produced

an organism with a portion of its gene from the high-fitness parent and the remainder from the randomly generated immigrant. Additionally, to maintain competitive fitness between immigrants and the population, we increased the proportion of the competitive immigrant genome that was from the parent over generations: the competitive immigrants were completely random until generation 5 and then the proportion of random genetic contribution declined progressively to 2.5% at generation 200. This is in contrast to normal two-point crossover which chooses ~50% of the genome from each parent.

2.2.3 Variable Pattern Length—Since therapeutic electrical stimulation is typically applied from minutes to days, the GA is designed to solve for a pattern of limited length that is then repeated for the duration of stimulation. Choosing a shorter pattern length tends to increase the speed of convergence of the algorithm, but may preclude potentially important low frequency variation (figure 4a). The variable pattern length (VPL) modification (figure 4b) progressively increases the length of the stimulation pattern across generations. Early patterns are created from subsegments of the pattern length, which are replicated and concatenated to produce full length patterns. The proportion of each organism that is used in this process increases over the course of several generations until the GA manipulates the full-length pattern and is able to identify any previously absent low frequency information (figure 4c). In this way, the VPL modification increases the convergence speed and robustness of the algorithm without limiting the range of achievable scores.

2.2.4. Predictive Immigrants—GAs evaluate every pattern of every generation, but typically only use information from the most recent generation to populate the next generation. This results in substantial information being discarded. To take advantage of the performance sampling that occurs for all patterns in each generation, we created Predictive Immigrants (PI) that incorporate the attributes of the highest scoring patterns across generations. The PI function tracks the features of patterns that score highest in each generation, creates new patterns that incorporate those features, and replaces half of the immigrants with the resulting patterns. The PI function has two major components: performance tracking and pattern creation. In the tracking stage (figure 5a) it records for each pattern the number of pulses, the interpulse intervals (IPIs), and the number of repeating IPI subpatterns (N_r). Subpatterns are determined using the following procedure:

1. Determine how many repeats of each individual IPI are in a pattern: n_r
2. For IPIs with n_r less than 10% of the total number of IPIs in a pattern:
  - a. If a neighboring IPI has an n_r greater than 10%, then add the n_r to it
  - b. If there is not, then ignore this IPI
3. For the remaining IPIs, take the lowest n_r , this is the number of repeating IPI subpatterns: N_r

These features are then associated with the pattern's fitness and added to a collective table that contains the data from every pattern tested. This table is then used to create probability distributions for choosing each pattern feature, with the probability of choosing a given value for a feature proportional to the fitness of that feature averaged across generations.

Then features are chosen by uniform random sampling of the cumulative probability distribution dependent variable.

The PI function uses three different methods to create patterns from these data, each producing patterns with different desired attributes. The first is the original method, which only uses the number of pulses and the IPIs:

1. Use the probability distribution of the number of pulses to choose a number of pulses: N_p
2. Use the probability distribution of the IPIs to choose $N_p - 1$ IPIs
3. Create a pattern with the corresponding IPIs in random order
4. If the pattern is longer than the desired overall pattern length, truncate it to length

The second is the Repeats method (figure 5b), which chooses fewer IPIs that are repeated N_r times, and was inspired by observing that many optimal patterns exhibited a repeating subpattern of IPIs:

1. Use the probability distribution of the number of pulses to choose a number of pulses: N_p
2. Use the probability distribution of the number of repeats to choose N_r
3. Use the probability distribution of the IPIs to choose $(N_p - 1)/N_r$ IPIs
4. Create a subpattern with the chosen IPIs and repeat it N_r times
5. If the pattern is longer than the desired overall pattern length, truncate it

The third technique is the Banding method and was based on observations that the optimal pattern often appeared to be semi-regular (e.g., a bursting pattern), and that persistent sub-optimal solutions (local maxima) were patterns where the longest IPIs (e.g., interburst intervals) were incorrect:

1. Skew the probability distribution of IPIs towards higher or lower values
2. Use the skewed IPI distribution to determine the number of times each IPI will be used
3. Create a pattern by placing pulses with given IPI's from smallest to largest and then largest to smallest
4. Repeat this process until the pattern exceeds the desired length and truncate

Four versions of the PI were tested: Original (PI-1), Repeats, Banding, and a combination of Repeats and Banding (PI-2). Repeats and Banding alone were comparable to PI-1 when averaged across the test functions, but the combination of the two substantially increased performance. PI-2 was the most effective of the PI variations and was used for all subsequent evaluations reported herein. The results comparing the different PI versions are in Appendix A.

2.2.5. Jumpstart—Like many optimization algorithms GAs often identify local fitness maxima, and we developed the Jumpstart (JS) function to escape local maxima by replacing the standard repopulation method when the GA records the same maximum fitness for 20 consecutive generations. Jumpstart broadens the tested parameter space by reducing the proportion of new patterns that are produced via traditional mating between parents and replacing them with an increased number of PI and CI patterns to promote genetic diversity (figure 6). Jumpstart continues for a preset number of generations or until the population records a higher maximum fitness.

2.3. Evaluation of algorithm performance

The effects of each modification individually, as well as their combinations, on the performance of the GA were evaluated first using test functions and subsequently using biophysically-based models of neural stimulation (see 2.4, below). To evaluate performance required hundreds of iterations of the GA, and each iteration (typically hundreds of generations) could take several days using biophysically-based neural models. Therefore, simpler test functions (or fitness tests) were created that required minimal computation but were sufficiently challenging to solve and contained multiple local maxima in addition to the global maximum. The functions were designed such that the worst and best solutions were known *a priori*, and all fitness scores were shifted by the minimum score and scaled by the maximum score to normalize scores on a zero (worst) to one (best) interval.

2.3.1. Interpulse Interval (IPI) Fit—IPI Fit was designed to mimic a scenario where the optimal stimulation pattern is composed of a few select IPI ranges (figure 7a). Each pattern was 450 bits long, and the function defined three ranges of IPIs with different number of points assigned to each of them: 15–17 (1 point), 30–32 (2.5 points), and 40–45 (3 points). Each stimulation pattern was then scored based upon the distribution of its respective IPIs across these ranges. To add complexity, we incentivized a lower average frequency of stimulation by dividing the final score by the weighted sum of the total number of pulses and the number of IPIs that did not fall into one of the three scoring ranges. This function was challenging to solve because there are multiple local maxima but only one global maximum.

2.3.2. Refractory Fit—Refractory Fit aimed to mimic the neuronal refractory period (figure 7b), with the goal of stimulating a neuron as often as possible, but not within its refractory period. Each pattern was 450 bits long, with each bit of 1 representing a pulse at that time. After the first pulse, a refractory period was randomly chosen between 0 and 7 bits long. If the next pulse in the pattern occurred after the refractory period of the previous pulse, it elicited an “action potential” and was assigned a new, random refractory period. If the pulse occurred during the refractory period, it was not assigned a new refractory period. The fitness was calculated by adding a point each time an applied pulse caused an “action potential” and subtracting a point each time a pulse was delivered during a refractory period. Finally, the fitness was normalized to a 0–1 scale, with the maximum score assumed to be a pattern that had a pulse every 7 bits, thus never occurring during a refractory period. The order of assigned refractory periods varied randomly across generations, so optimized patterns had to respond robustly for the population to converge. This resulted in multiple local maxima, and no single pattern could be optimal across all generations of the GA.

2.3.3. N-Queens Fit—The N-Queens Fit (figure 7c) is a common optimization problem where the objective is to place the most queens on a chess board without any two of them able to attack each other. The chessboard was expanded to 30×30 squares to increase the difficulty, so each pattern was 900 bits long. Each pattern was first converted to locations on the chess board. One point was awarded for each queen on the board that could not be attacked, and one point was deducted for each queen that was threatened. Finally, each score was normalized to a 0–1 scale based on the known maximum number of queens, 30. This test is not explicitly based on neuronal concepts, but shares certain characteristics (low pulse to location ratio, pulse to pulse dependence, etc.) that make it a useful test case for the proposed modifications.

2.4. Validation in complex biophysical models of neural stimulation

In addition to the three test functions, the modifications to the GA were tested on two biophysically-based computational models of neural stimulation: deep brain stimulation (DBS) in a network model of the cortical – basal ganglia – thalamic system (Kumaravelu *et al.* 2016) and spinal cord stimulation (SCS) in a network model of the dorsal horn (Zhang *et al.* 2014). In the model of the basal ganglia, the objective was to identify the temporal pattern of DBS that minimized both the signal power of neural activity in the beta frequency band (13 – 35 Hz) in the globus pallidus (GP), as this is associated with symptoms of bradykinesia (Kühn *et al.* 2006, Hammond *et al.* 2007), and the average stimulation frequency, as this reduces energy demands of DBS (Brocker *et al.* 2017). The fitness function was:

$$f = \frac{1000}{\text{Beta Power} + 3 \times \text{number of pulses}} \quad (1)$$

In the model of the dorsal horn, the objective was to identify a temporal pattern of SCS that reduced the average firing rate (AFR) of the model wide dynamic range neuron (WDR), as this is a proxy for the intensity of perceived pain (Simone *et al.* 1991) and reduced the average stimulation frequency (ASF), which, as in DBS, will reduce stimulator energy requirements. The fitness function was:

$$f = \begin{cases} 20 * (10 - \text{AFR}) + (6.2 - \text{AFR}) + (40 - \text{ASF}) & \text{if } \text{AFR} > 10 \\ (6.2 - \text{AFR}) + (40 - \text{ASF}) & \text{otherwise} \end{cases} \quad (2)$$

The fitness was determined by comparison to the performance of 40 Hz SCS, which reduced AFR to 6.2 spikes/s. Therefore, negative scores indicated patterns that did not perform as well as the constant frequency standard treatment. The piecewise function was implemented because it did not matter how efficient the pattern of stimulation was if it did not sufficiently reduce WDR activity.

Two versions of the GA were run on each model: all modifications on and all modifications off. In addition, for the modifications off condition, after 200 generations the modifications were all switched on and the algorithm was run for an additional 50-100 generations. Due to

the computational demands of these models, the algorithms were run on the Duke Shared Computing Cluster.

2.5. Data analysis

The GA was run on the three test functions in two primary conditions: each modification individually added (Plus) and all modifications on with each modification individually removed (Minus). We also ran a no modifications on case (None) and an all modifications on case (All) with which to compare the results. This strategy enabled determination of the effects of each modification individually and would reveal any redundancy in modification contribution. For each condition, the GA was repeated with each of the three test functions for 200 replicate trials and each trial ran for 200 generations with a population size of 50 in each generation. The best scores for generations 50 and 200 were saved, which provided insight into the short-term convergence (generation 50) as well as the long-term optimal solution (generation 200) of the GA. Since the fitness scores were multimodal and highly non-normal, we used a Kruskal-Wallis test for each dataset sampled at different iteration numbers to determine whether GA strategy had a significant effect on the normalized fitness ($p < 0.05$). Then we used a Steel-Dwallow's all pairs comparison at each sample endpoint to determine significance ($p < 0.05$). The results reveal whether two distributions are significantly different from one another, and if so, which distribution has a higher value.

3. Results

We evaluated the performance of the GA across three test functions and ran the algorithm for each of the 12 methodological variations (None, All, Plus, and Minus). The distributions of fitness scores across the GA variations were then compared across all three tests to draw conclusions about utility, and the performance was examined across individual tests for further insight into how each modification influenced performance. The figures show the actual distributions of fitness scores, and the tables display comparisons of the distributions and results of the statistical analysis of the distributions.

3.1. Overall effects of GA modifications across test functions

The minimum and median fitness values (figure 8) contain information important for considering GA performance that is not directly defined within our other analysis. The median represents the most likely result of a single GA run, while the minimum shows the expected lower bound of fitness score from a single iteration of the GA. The standard GA performed poorly, and the proposed modifications substantially increased both the minimum and median fitness at both generation 50 and generation 200. When used individually (Plus), each of the proposed modifications increased performance except for Jumpstart. In contrast, individual removal of each modification caused a substantial reduction in most median and minimum fitness scores at generation 50 from the case with all modifications on. However, individual removal (Minus) of several modifications did not cause reductions in many median and minimum fitness values at generation 200, suggesting that performance may be saturated by generation 200. Finally, the All case exhibited the consistently highest median and minimum fitness, indicating the robustness of the proposed modifications.

3.2. Effects of GA modifications on Interpulse Interval Fit task

When collectively used, the proposed modifications increased the median fitness scores on the IPI fit task from 0.32 and 0.49 to 0.97 and 0.98 at generations 50 and 200, respectively, and led to approximately five-fold increases in the minimum fitness scores (figure 9, table 1). This shows a dramatic increase in the short-term convergence speed of the GA and the consistency of the algorithm across iterations. The inclusion (figure 9a) of CI, PMM, and PI each individually improved performance from the baseline GA at both generations 50 and 200. VPL did not affect performance at generation 50, but greatly increased performance by generation 200, and JS did not result in an improvement. Removing individually any of the GA modifications (figure 9b) reduced the fitness scores; these effects were most pronounced at generation 50, and by generation 200 the median fitness was not influenced with JS, VPL or CI removed, although the minimum fitness scores were lower.

3.3. Effects of GA modifications on Refractory Fit task

The All case increased the median scores from 0 and 0.02 to 0.63 and 0.83 at generation 50 and 200 respectively (figure 10, table 2), and once again substantially increased the minimum scores. Removing the JS and PMM modifications did not affect performance by generation 50, and removing CI did not significantly affect the performance at either generation 50 or 200. However, removing each modification substantially decreased the minimum score even if it did not significantly affect the distribution.

3.4. Effects of GA modifications on N-Queens Fit

The proposed modifications in the All case dramatically increased the median scores from 0.08 and 0.14 to 0.53 and 0.70 at generations 50 and 200, respectively (figure 11, table 3). This difference is further highlighted by the substantially different minimum values. Individually, the PMM modification matched the performance of all the modifications combined, and the VPL outperformed the All case. On the other hand, removing either of these two modifications resulted in significantly higher scores, indicating that these modifications were negatively affecting performance. Only when removing the CI modification were scores reduced, and only at generation 200.

3.5. Optimal temporal patterns of deep brain stimulation

The proposed modifications to the GA increased the rate of convergence of the GA and resulted in temporal patterns of DBS with higher fitness values than the standard GA (figure 12a). The proposed modifications also rescued populations that had converged to suboptimal patterns, as shown by the consistent and dramatic increase in fitness when the modifications were switched on after the standard GA had converged.

3.6. Optimal temporal patterns of spinal cord stimulation

Fitness scores in the SCS model were compared to the fitness of standard 40 Hz SCS, which was consistently the highest performing constant frequency pattern. The proposed modifications increased the rate of convergence and resulted in temporal patterns of SCS with higher fitness values (figure 12b). Further, when the modifications were turned on after the standard GA had plateaued, the fitness once again rapidly increased.

4. Discussion

The proposed modifications to the standard GA method increased the rate of convergence, solution consistency, and solution fitness across both a standardized set of evaluation tasks and two applications relevant to the design of temporal patterns of neural stimulation. Although the impact of individual modifications varied across tasks, we suggest applying all the proposed modifications simultaneously as this approach consistently exhibited high median and minimum fitness scores. Further, there were apparent synergistic effects of some of the proposed modifications. For example, Jumpstart provided no measurable utility when implemented alone, but when it was removed from the complete set, the rate of convergence was negatively affected. Since we did not observe every possible combination of the individual modifications, it is possible that a limited combination of modifications may, depending on the task, perform better than applying all modifications together.

There are several factors that may have contributed to the large variability in the minus cases (figures 9 and 10). Some combination of at least three of our innovations may drive solutions toward a low scoring local maximum, and this may be overcome by modifications that drive more iterations. Since we did not test exhaustively all combinations of our modifications, we do not know which combinations contribute to the behaviors, but the VPL and PI modifications appear to have opposite effects on the range of solutions. This suggests the variability is due to poorly scoring, high frequency local maxima since our modifications cause the GA to focus on high frequency patterns in early generations. Additionally, most of the search space for both Interpulse Interval Fit and Refractory Fit result in scores of 0, so the solution space surrounding the local maxima likely had scores of 0 which would cause the poorly scoring local maxima to dominate selection.

4.1. Pulse Mutation

The Pulse Mutation Method modification was highly successful in each of the three test functions. Its primary effect was to reduce the variance in the scores across trials, but at the cost of reducing the fitness of some of the top performing trials. PMM statistically matched the performance of the combined modifications on the N-Queens test, but removing it improved the GA performance. This paradoxical effect may be explained by the design of the modification. PMM prevented changes to the average number of pulses as a result of mutation, and while this seems to be beneficial for poorly performing solutions, it appears that preferentially adding pulses benefits highly performing solutions. This explanation is consistent across the solutions of all three tests, but is most exemplified by the N-Queens test, possibly because it is not based on neuromodulation principles.

4.2. Competitive Immigrants

Adding Competitive Immigrants significantly improved scores, mostly by reducing the number of poorly performing trials. This might be expected as the CI modification is only designed to maintain pattern diversity. However, it was the only innovation that failed to increase significantly Minus case scores at both sample times in one of the neuronal-based tests. The CI modification's inability to influence Refractory fit scores might be inherent to its methodology. Over many iterations, this modification limits the global influence of new

genetic material by converting the contribution of random immigrants into small, local strings of random elements. This is likely to be useful in the IPI and N-Queens Fits, where changing a subsection of the pattern can provide local benefits while minimally affecting the performance of the remainder of the pattern. However, in Refractory Fit changing one subsection to random elements can have a dramatic effect on the performance of the remainder of the pattern. In later generations of a GA, where patterns are near-optimum, such large changes are predominantly detrimental.

4.3. Variable Pattern Length

The Variable Pattern Length modification improved performance for all but the combined modifications N-Queens test. This implies that it had a strong negative interaction with at least one other modification. We suspect the negative interaction was with the PI. Both can solve the N-Queens problem alone, but neither benefited the combined modifications scores. Varying the pattern length only improves the GA performance in cases where inferences about the total solution can be drawn from smaller segments. However, because the PI depends heavily on IPIs, the combination of these strategies may be finding high performing local maxima but reducing the final fitness.

4.4. Predictive Immigrants

The Predictive Immigrants modification is clearly one of the most effective improvements. It is not surprising that the PI was highly effective at the IPI Fit function, because both operate using the interpulse intervals of a pattern. However, PI also improved performance for both the Refractory Fit and N-Queens Fit. Neither of these explicitly used interpulse intervals, and yet PI successfully increased the score when implemented individually for both.

4.5. Jumpstart

Jumpstart is seemingly the least effective of the improvements. When implemented individually, it has no measurable effect on any of the fit functions. However, this is not too surprising as JS is designed to be used when the GA gets stuck at a local maximum. In the None case, the standard GA performed poorly, and changing the population proportions using Jumpstart had negligible effect. However, removing JS significantly reduced scores at 50 generations in IPI fit and 200 generations in the Refractory fit. The Minus cases are more illustrative of Jumpstart's benefit, as these cases test whether it provides any performance improvement to GAs that retain diverse competition when approaching local maxima. In addition, Jumpstart has no effect if the algorithm continuously makes incremental improvements, so it has more of an impact if the solutions saturates near the optimum, which is more common in the Minus cases.

4.6. Optimized temporal patterns of stimulation in biophysical models

Although the arbitrary fit functions provide clear insight into the efficacy of the proposed modifications, the performance of the GA on biophysically-based computational models provided compelling support for the modified methodology. In both models, the modified GA not only reached substantially higher final fitness than the standard GA, but it did so in many fewer generations. Furthermore, when the modifications were added to the GA after

the standard methodology had converged, the algorithm rapidly converged to a new solution with much higher fitness.

In addition to the fitness scores of the final generation patterns, figure 13 shows the raster plots of a representative subset of the stimulation trains resulting from the modified and standard GA applied to the dorsal horn model. Both algorithms converged to non-regular patterns of stimulation, which emphasizes the necessity and design utility of a GA. Without the modifications, when the fitness plateaued the population had converged to a single solution. The random immigrants, while introducing diversity, will not be selected as parents for the next generation as their fitness is no longer competitive. Further, the children of the previous generation are barely distinguishable from the elites, and thus two-point crossover mating will do little to explore further the solution space. However, the modified GA identified solutions with substantially higher fitness and greater diversity of pulse timing. This diversity contributes to avoiding local maxima and thus identifying patterns with greater fitness.

4.7. Limitations and future directions

The scope of this study was limited to solutions of binary strings, as our objective was to improve the GA for the design of temporal patterns of stimulation. Binary encoding is well suited to this application, as each bit directly corresponds to the presence or absence of a pulse in the stimulation pattern. The GA could be extended to optimize stimulation amplitude or pulse duration, but the proposed modifications were designed for variables directly coded by the bit string, and it is not clear that indirectly coded variables, e.g., pulse amplitude as a binary code, would benefit from these modifications. Crossover can split binary encoded parameters, and the number of bit changes is not well correlated to the magnitude of parameter change (Ashlock 2006). A real-number encoding scheme would likely be more appropriate for these applications.

Although we constrained our modifications to the repopulation step of the GA, considerable effort has been made to improve almost every step of the GA methodology (Mukhopadhyay *et al.* 2015, Zitzler 1999), and even which parameters the GA can manipulate (Yusup *et al.* 2012). GA options now represent a parameter set so large that some groups have turned to applying one search algorithm to optimize another (Jing *et al.* 2010) and have found success, even in the field of neuroscience (Li *et al.* 2017). Our proposed modifications can also be combined with other GA improvements or variations that modify other elements of the algorithm. Two possible examples are parallel and multi-objective GAs, because neither make major adjustments to the repopulation step. In fact, we already merged our modified repopulation step into the popular NSGA-II multi-objective GA for use with the DBS model, using beta band power and average stimulation frequency as two objectives (Deb *et al.* 2002).

Both parallel and multi-objective GAs require more computation than a typical GA, but increasing computational power is continuously shifting the balance between algorithm efficiency and complexity. We intend our algorithm to be applied to computationally-intensive (nonlinear neuronal) models, and we did not consider the computational efficiency of the algorithm itself as it represents a negligible amount of the total computational

demand. We also did not consider modifications that substantially increased the number of model simulations per generations, for example, fitness sharing and objective switching (Zitzler 1999). Also, the proposed modifications intentionally bias the algorithm's exploration of the parameter space towards the realistic solution space for neural stimulation to reduce the number of iterations/simulations required to converge. However, these constraints may not be necessary with less complex applications or with additional computational power.

It is certainly our intent to test *in vivo* the patterns we developed. However, such testing is beyond the scope of this paper, as GA-designed patterns have previously been successfully applied *in vivo* (Brocker *et al.* 2017), and the success or failure of *in vivo* applications reflects on the model used by the GA rather than the algorithm itself.

5. Conclusion

Genetic Algorithms can determine optimal solutions both quickly and effectively, and binary GAs are particularly well suited to identifying optimal temporal patterns of neural stimulation. The proposed modifications: Pulse Mutation, Competitive Immigrants, Variable Pattern Length, Predictive Immigrants, and Jumpstart, substantially improved the performance of the standard GA, and their combination was highly effective across multiple test functions, including application to biophysically-based computational models of neuromodulation. This improved genetic algorithm provides a robust method for finding optimal temporal patterns of stimulation.

Acknowledgments

This work was supported by NIH R37 NS040894, the Robert Plonsey Fellowship from the Duke University Department of Biomedical Engineering, and the Duke Biomedical Engineering Departmental Second Year Fellowship and made use of the Duke Shared Compute Cluster. WMG is an inventor on licensed intellectual property on temporal patterns of neural stimulation that is owned by Duke University.

Appendix A. Predictive Immigrants Design

There were several aspects of the Predictive Immigrants (PI) modification that we could manipulate individually, so we examined interactions of these methodologies to determine the best strategy for the GA. We have four versions: PI-1 is the base version, Banding and Repeat are additions to PI-1, and PI-2 is PI-1 with both Expand and Repeat added. Averaging the medians across the different fit functions showed differences in the different PI versions after 50 and 200 generations (figure A1). When added individually (figure A1a), PI-2 did not seem to be any superior to PI-1 + Repeat. However, when the PI versions were added to the GA with the other modifications also on (figure A1b), Repeat appeared to have a negative interaction with the other modifications, producing a worse score at generation 50 than PI-1. However, PI-2 produces the best scores after both 50 and 200 generations, indicating that Repeat and Banding may have a synergistic effect when the other modifications are present.

Figure A1.

Results from the combined test functions for each of the four Predictive Immigrant (PI) versions. PI-1 is the base PI function, and PI-2 is PI-1 with both Banding and Repeat turned on. Since there are only three test functions, the bounds of the box and the line are at the values of the three scores. **(a)** Each PI version added individually to the base GA with no other modifications (mods). **(b)** Each PI version added to the GA with all other modifications (Mods) turned on.

References

- Ashlock, D. Evolutionary computation for modeling and optimization. Guelph, Ontario: Springer; 2006.
- Benabid, aL, Chabardes, S., Mitrofanis, J., Pollak, P. Deep brain stimulation of the subthalamic nucleus for the treatment of Parkinson's disease. *Lancet Neurol.* 2009; 8(1):67–81. [PubMed: 19081516]
- Brocker DT, Swan BD, Turner DA, Gross RE, Tatter SB, Miller Koop M, Bronte-Stewart H, Grill WM. Improved efficacy of temporally non-regular deep brain stimulation in Parkinson's disease. *Experimental Neurology.* 2013; 239(1):60–67. [PubMed: 23022917]
- Brocker DT, Swan BD, So RQ, Turner DA, Gross RE, Grill WM. Optimized temporal pattern of brain stimulation designed by computational evolution. *Science Translational Medicine. Science Translational Medicine.* 2017; 9(371)
- Davis, L. Handbook of Genetic Algorithms. London: International Thomson Computer; 1991.
- Deb, K. Multi-objective optimization using evolutionary algorithms. Chichester: John Wiley & Sons; 2009.
- Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation.* 2002; 6(2):182–197.
- De Ridder D, Vanneste S, Plazier M, van der Loo E, Menovsky T. Burst spinal cord stimulation: Toward paresthesia-free pain suppression. *Neurosurgery.* 2010; 66(5):986–990. [PubMed: 20404705]
- Elhossini A, Areibi S, Dony R. Strength Pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization. *Evol Comput.* 2010; 18(1):127–56. [PubMed: 20064026]

- Feng X-J, Greenwald B, Rabitz H, Shea-Brown E, Kosut R. Toward closed-loop optimization of deep brain stimulation for Parkinson's disease: concepts and lessons from a computational model. *Journal of Neural Engineering*. 2007; 4(2):L14–L21. [PubMed: 17409470]
- Feng X-J, Shea-Brown E, Greenwald B, Kosut R, Rabitz H. Optimal deep brain stimulation of the subthalamic nucleus—a computational study. *Journal of Computational Neuroscience*. 2007; 23(3): 265–82. [PubMed: 17484043]
- Giuffrida JP, Crago PE. Functional restoration of elbow extension after spinal-cord injury using a neural network-based synergistic FES controller. *IEEE Trans Neural Syst Rehabil Eng*. 2005; 13(2):147–152. [PubMed: 16003892]
- Goldberg, DE. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston: Addison-Wesley; 1989.
- Hammond C, Bergman H, Brown P. Pathological synchronization in Parkinson's disease: networks, models and treatments. *Trends in Neurosciences*. 2007; 30(7):357–364. [PubMed: 17532060]
- Kent AR, Grill WM. Model-based analysis and design of nerve cuff electrodes for restoring bladder function by selective stimulation of the pudendal nerve. *Journal of Neural Engineering*. 2013; 10(3)
- Keserü M, Feucht M, Bornfeld N, Laube T, Walter P, Rössler G, Velikay-Parel M, Hornig R, Richard G. Acute electrical stimulation of the human retina with an epiretinal electrode array. *Acta Ophthalmologica*. 2012; 90(1):e1–e8. [PubMed: 22067614]
- Kühn AA, Kupsch A, Schneider GH, Brown P. Reduction in subthalamic 8–35 Hz oscillatory activity correlates with clinical improvement in Parkinson's disease. *Eur J Neurosci*. 2006; 23:1956–1960. [PubMed: 16623853]
- Kumaravelu K, Brocker DT, Grill WM. A biophysical model of the cortex-basal ganglia-thalamus network in the 6-OHDA lesioned rat model of Parkinson's disease. *Journal of Computational Neuroscience*. 2016; 40(2):207–229. [PubMed: 26867734]
- Jing, L., Jin-Jia, W., Chun-Liang, H., Chong-Xiao, M., Wen-Xue, H. A Novel Modified Genetic Algorithm for Training Wavelet Neural Networks; 2010 First International Conference on Pervasive Computing, Signal Processing and Applications; 2010.
- Lan N, Feng H, Crago PE. Neural network generation of muscle stimulation patterns for control of arm movements. *IEEE Transactions on Rehabilitation Engineering*. 1994; 2(4):213–224.
- Li H, Yuan D, Ma X, Cui D, Cao L. Genetic algorithm for the optimization of features and neural networks in ECG signals classification. *Scientific Reports*. 2017; 7:41011. [PubMed: 28139677]
- Man KF, Tang KS, Kwong S. Genetic algorithms: Concepts and applications. *IEEE Transactions on Industrial Electronics*. 1996; 43(5):519–534.
- Miller JP, Eldabe S, Buchser E, Johaneck LM, Guan Y, Linderöth B. Parameters of Spinal Cord Stimulation and Their Role in Electrical Charge Delivery: A Review. *Neuromodulation*. 2016; 19(4):373–384. [PubMed: 27150431]
- Mukhopadhyay A, Maulik U, Bandyopadhyay S. A Survey of Multiobjective Evolutionary Clustering. *ACM Comput Surv*. 2015; 47(4)
- Peña E, Zhang S, Deyo S, Xiao Y, Johnson MD. Particle swarm optimization for programming deep brain stimulation arrays. *J Neural Eng*. 2017; 14(1)
- Ramasubbu C, Flagg A, Williams K. Principles of electrical stimulation and dorsal column mapping as it relates to spinal cord stimulation: An overview. *Current Pain and Headache Reports*. 2013; 17(2):315. [PubMed: 23299905]
- Sankar RS, Asokan P, Saravanan R. Selection of machining parameters for constrained machining problem using evolutionary computation. *Int J Adv Manuf Technol*. 2007; 32(9):892.
- Sherman ND, Amundsen CL. Current and future techniques of neuromodulation for bladder dysfunction. *Current Urology Reports*. 2007; 8(6):448–454. [PubMed: 18042323]
- Simone DA, Sorkin LS, Oh U, Chung JM, Owens C, LaMotte RH, Willis WD. Neurogenic hyperalgesia: central neural correlates in responses of spinothalamic tract neurons. *J Neurophysiol*. 1991; 66(1):228–46. [PubMed: 1919669]
- Tan DW, Schiefer MA, Keith MW, Anderson JR, Tyler J, Tyler DJ. A neural interface provides long-term stable natural touch perception. *Science Translational Medicine*. 2014; 6(257)

- Trevathan JK, Yousefi A, Park HO, Bartoletta JJ, Ludwig KA, Lee KH, et al. Computational Modeling of Neurotransmitter Release Evoked by Electrical Stimulation: Nonlinear Approaches to Predicting Stimulation-Evoked Dopamine Release. *ACS Chem Neurosci*. 2017; 8(2):394–410. [PubMed: 28076681]
- Verrills P, Sinclair C, Barnard A. A review of spinal cord stimulation systems for chronic pain. *Journal of Pain Research*. 2016; 9:481–492. [PubMed: 27445503]
- Wilson BS, Dorman MF. Cochlear implants: A remarkable past and a brilliant future. *Hearing Research*. 2008; 242(1–2):3–21. [PubMed: 18616994]
- Wongsarnpigoon A, Grill WM. Energy-efficient waveform shapes for neural stimulation revealed with a genetic algorithm. *J Neural Eng*. 2010; 7(4):046009. [PubMed: 20571186]
- Yusup N, Zain AM, Hashim SZM. Evolutionary techniques in optimizing machining parameters: Review and recent applications (2007–2011). *Expert Syst Appl*. 2012; 39(10):9909–27.
- Zhang TC, Janik JJ, Grill WM. Modeling the effects of spinal cord stimulation on wide dynamic range dorsal horn neurons: Influence of stimulation frequency and GABAergic inhibition. *Journal of Neurophysiology*. 2014; 112(3):552–67. [PubMed: 24790169]
- Zhang Q, Li H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*. 2007; 11(6):712–731.
- Zitzler, E. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Aachen: Shaker; 1999.

Figure 1.
Flowchart of a standard (solid line) and our modified (dashed line) genetic algorithm.

Figure 2.

(a) Representation of frequency bias in the two mutation methods. Random patterns were initialized with a total length of 200 and either 25 or 175 pulses. Each pattern was then mutated 200 times using either the standard point mutation or Pulse Mutation Method (PMM). (b) Comparison of the standard point mutation and PMM. In the standard method, pulses locations are randomly chosen, and the values at those locations are changed. In the PMM, the mutation type is first randomly determined, then the locations are determined, and finally the mutation is performed.

Figure 3.

Method for creation of Competitive Immigrants. A parent is selected based on its fitness and a random immigrant is created. The parent and immigrant are then crossed to produce the Competitive Immigrant that is inserted into the population. The proportion of the Competitive Immigrant from the parent increases over subsequent generations. Dark regions in the Competitive Immigrant come from the parent, and light regions come from the random immigrant.

Figure 4.

(a) The fitness levels achieved by a GA using the full-size repeat length and a substantially smaller repeat length. (b) Method for the Variable Pattern Length modification. In early generations, only a small portion of the pattern is solved with the GA, and is then duplicated to fill out the full length of the pattern. Over subsequent generations, the portion of the pattern that the GA solves is increased, until it solves for the entire pattern length. (c) The minimum frequency (Hz) that can be represented in each generation using the VPL method for a 450 bit long pattern. As the pattern length increases across generations, the minimum frequency that can be represented moves to lower frequencies.

Figure 5.

(a) Method for the performance tracking stage of the Predictive Immigrant (PI) function. Three features are tracked: the interpulse intervals, the number of pulses, and the number of repeats, and each feature is associated with the fitness of the pattern. **(b)** IPI pattern creation using the Repeats method. Probability density curves for the three tracked features (interpulse intervals, number of pulses, and number of repeats) are used to produce new patterns.

Figure 6.

Method for the Jumpstart function. When the GA is stuck at a local maximum for a preset number of generations, Jumpstart increases the proportion of the next generation that are produced from PI and CI.

Figure 7.

Simplified schematics of each of the test functions used to evaluate the modifications to the GA. **(a)** The Interpulse Interval (IPI) Fit function design. Each pattern was first broken down into its component IPIs. Points were given for IPIs that fell within pre-defined ranges. The final score was weighted by the total number of pulses and normalized to one. **(b)** The Refractory Fit function. After the first pulse, a refractory period was randomly chosen. The next pulse in the pattern was found, and if it occurred after the refractory period it elicited an action potential and a new random refractory period, but if it occurred during the refractory period it did nothing. Points were given for each pulse that elicited an action potential and were taken away for each pulse that occurred during a refractory period. **(c)** The N-Queens Fit function to fit the maximum number of queens on a chessboard without any two of them being able to attack each other. The pattern was converted to placement of the queens, and points were given for each queen that could not attack another, and taken away for each that could.

Figure 8.

Changes in performance of the GA from each of the proposed modifications: Jumpstart (JS), Variable Pattern Length (VPL), Competitive Immigrants (CI), Pulse Mutation Method (PMM), and Predictive Immigrants (PI), combined across the three test functions. A plus sign, +, indicates the “Plus” case, in which each modification was added individually, and a minus sign, –, indicates the “Minus” case, in which all modifications were turned on, and then removed individually. **(a)** The minimum fitness scores in each test function at generation 50 and 200. **(b)** The median fitness scores of each test function at generation 50 and 200.

Figure 9.

Changes in performance from the proposed modifications to the GA on the Interpulse Interval Fit function. Results from 200 trials are shown as quartile box-and-whisker plots where whiskers extend to the absolute minimum and maximum, the box covers the interquartile range of scores, and the line is the median score.

Figure 10.

Changes in performance from the proposed modifications to the GA on the Refractory Fit function. Results from the 200 trials are shown as quartile box-and-whisker plots where whiskers extend to the absolute minimum and maximum, the box covers the interquartile range of scores, and the line is the median score.

Figure 11.

Changes in performance from the proposed modifications to the GA on the N-Queens Fit function. The results from the 200 trials are shown as quartile box-and-whisker plots where whiskers extend to the absolute minimum and maximum, the box covers the interquartile range of scores, and the line is the median score.

Figure 12.

Changes in fitness across generations of the GA for the two biophysically-based models of neural stimulation. **(a)** Results from applying the GA to a network model of the basal ganglia-thalamus-cortex to optimize temporal patterns of DBS. **(b)** Results from applying the GA to a network model of the spinal dorsal horn to optimize temporal patterns of SCS. The inset shows just the results scoring between -30 and 10 . For each model, GAs were run under two conditions: modifications on (blue) and modifications off (red) for 200 generations. For the modifications off condition, after the 200 generations the modifications were turned on (blue). Also plotted is the score for a clinically standard, constant frequency stimulation (black dash): 130 Hz for (a) and 40 Hz for (b).

Figure 13.

Raster plots of a representative subset of the stimulation patterns in the final generation of standard GA with (a) modifications off and (b) modifications on applied to the network model of the dorsal horn. The y-axis gives the fitness score for each pattern.

Table 1

Statistics for the Interpulse Interval Fit results.

	Compared to None						Compared to All					
	All	+JS	+VPL	+CI	+PMM	+PI	None	-JS	-VPL	-CI	-PMM	-PI
50	+	×	×	+	+	+	-	-	-	-	-	-
200	+	×	+	+	+	+	-	-	×	-	-	-

Effect on fitnesses: +, significantly higher; ×, no change; −, significantly lower

Table 2

Statistics for the Refractory Fit results.

		Compared to None						Compared to All					
	All	+JS	+VPL	+CI	+PMM	+PI		None	-JS	-VPL	-CI	-PMM	-PI
50	+	×	+	+	+	+		-	×	-	×	×	-
200	+	×	+	+	+	+		-	-	-	×	-	-

Effect on fitnesses: +, significantly higher; ×, no change; −, significantly lower

Table 3

Statistics for the N-Queens Fit results.

		Compared to None					Compared to All					
	All	+JS	+VPL	+CI	+PMM	+PI	None	-JS	-VPL	-CI	-PMM	-PI
50	+	×	+	+	+	+	-	×	+	×	+	×
200	+	×	+	+	+	+	-	×	+	-	+	×

Effect on fitnesses: +, significantly higher; ×, no change; −, significantly lower