

# Single and Multi-objective *in Silico* Evolution of Tunable Genetic Oscillators

Spencer Angus Thomas and Yaochu Jin

Department of Computing,  
University of Surrey  
Guildford, Surrey, GU2 7XH, UK  
{s.thomas,yaochu.jin}@surrey.ac.uk

**Abstract.** We compare the ability of single and multi-objective evolutionary algorithms to evolve tunable self-sustained genetic oscillators. Our research is focused on the influence of objective setup on the success rate of evolving self-sustained oscillations and the tunability of the evolved oscillators. We compare temporal and frequency domain fitness functions for single and multi-objective evolution of the parameters in a three-gene genetic regulatory network. We observe that multiobjectivization can hinder convergence when decomposing a period specific based single objective setup in to a multi-objective setup that includes a frequency specific objective. We also find that the objective decomposition from a frequency specified single objective setup to a multi-objective setup, which also specifies period, enable the synthesis of oscillatory dynamics. However this does not help to enhance tunability. We reveal that the use of a helper function in the frequency domain improves the tunability of the oscillators, compared to a time domain based single objective, even if no desired frequency is specified.

**Keywords:** Gene regulatory networks, *in silico* evolution, sustained oscillation, evolutionary algorithms, multiobjectivization.

## 1 Introduction

An important area of computational science is systems biology, and over recent years there have been many contributions to the field of biology from computer scientists and mathematicians. Many biological systems lack a global theoretical basis and one way to improve our understanding is to analyse the dynamics of the system *in silico*, i.e. in a computational environment. The evolutionary synthesis, or production, of these dynamics can be tested in computational simulations to investigating biological hypothesis that may be subject to experimental, theoretical and timescale limitations.

Nature is full of complex biological systems and those that are of interest to biologists and computer scientists often consist of large numbers of interacting genes. These complex networks can be broken down into smaller subnetworks

often containing repeating patterns that appear more often than they would in a random network [1, 2]. These repeating patterns are known as network motifs and are believed to be the building blocks of complex biological networks, which are modular in structure [1, 2]. Motifs that produce self-sustained oscillations are particularly important in biological systems [3, 4] and are involved in circadian rhythms [5] and the active transport of hydrogen ions [6]. The ability to accurately tune the period of a genetic oscillator is vital in biological modelling due to the range of oscillator periods observed. Biological oscillatory time scales range from seconds for neuronal and cardiac rhythms [5], to minutes for mitosis cell cycles [7], to hours for the sleep/wake cycle [8] and circadian rhythms [6], to weeks for the ovarian cycle [5] and to years for predator-prey population cycles [9]. Further examples and details about biological oscillations can be found in [5, 10, 11].

To produce oscillations, biochemical systems require negative (repressive) regulatory circuits [4], which also improves robustness to environmental perturbations [12], a necessity of many biological systems [1]. Moreover negative auto-regulation (NAR), where a gene will repress its own protein production, results in a rapid response to an input signal which is important for biological systems and are common in biology as it can help reduce noise [1]. Positive feedback loops have been demonstrated to enhance frequency tunability in biological oscillators, with little cost to amplitude, biological systems therefore often contain both positive and negative regulatory circuits [13].

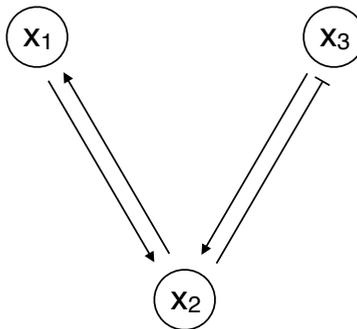
Gene regulatory networks (GRNs) can be modelled through gene-protein interactions, where genes produce proteins, which interact with genes and affect protein production. The dynamics and structure of GRNs is important in the understanding of natural evolution [3] and the analysis of GRN motifs is a growing area of importance in systems biology [1, 14]. A common way to model gene regulatory dynamics is to use differential equations [15]. To generate typical regulatory dynamics *in silico*, evolutionary algorithms (EAs) have widely been used to evolve the parameters and structure of GRNs [3, 16, 17, 18, 19]. However, it is noted that the evolution of oscillatory dynamics is non-trivial and many different objective functions have been suggested to facilitate the evolution of oscillation [3, 16].

In this work, we investigate the use of techniques such as multi-objective optimisation and multiobjectivization to improve the success rate and tunability in evolving sustained genetic oscillators, which have been used to accelerate convergence speed and obtain the global optimum [20, 21, 22].

The differential equations describing the dynamics of the genetic networks studied in this work are introduced in Section 2. Various single and multi-objective fitness setups are proposed in Section 3. The description of the single- and multi-objective EAs adopted in this work is given in Section 4, followed by the experimental results and discussions in Section 5. Section 6 concludes the paper and suggests a few topics for future work.

## 2 Gene Regulatory Networks

Here we model the GRN, shown in Fig. 1, using the interaction of the genes through their protein production. The GRN contains a negative feedback loop, as this can generate sustained oscillations for interacting genes, and a positive feedback loop, which has been demonstrated to aid evolvability, robustness and tunability of the oscillator when used with a negative loop [13]. All genes in the network also use negative auto-regulation (NAR), where the protein of a gene represses its own production [3]. The use of NAR can decrease response time to an input signal [1], which is an essential function in biological systems. This functionality is important in many prokaryotic transcription networks, such as *E. coli*, which uses NAR in up to 56% of it's expressed transcription factors [23].



**Fig. 1.** A consistently regulated motif consisting of three genes with  $x_2$  as the target gene. Here arrows and lines with bar ends represent activating and repressor interactions between the genes respectively. All the genes here also have negative auto-regulations (not shown).

Here we model the protein interactions between the genes using the following differential equations,

$$\dot{x}_1 = a_{12}H_{12}(x_2) - a_{11}x_1, \quad (1)$$

$$\dot{x}_2 = a_{23}L\left(H_{23}(x_3), H_{21}(x_1)\right) - a_{22}x_2, \quad (2)$$

$$\dot{x}_3 = a_{32}H_{32}(x_2) - a_{33}x_3. \quad (3)$$

Where  $\dot{x}_i$  is the time derivative of  $x_i$ ,  $x_2$  is the target gene,  $a_{ij}$  are regulatory parameters,  $H_{ij}(x_j)$  is the Hill function and  $L(H_{23}(x_3), H_{21}(x_1))$  represents the logic function which combines the interaction of two regulatory genes to the

target gene. The Hill function can represent either an activating,  $H_{ij}^a(x_j)$ , or repressive gene interaction,  $H_{ij}^r(x_j)$ , as shown in (4).

$$H_{ij}^a(x_j) = \frac{\beta x_j^n}{\theta_i^n + x_j^n}; \quad H_{ij}^r(x_j) = \frac{\beta}{1 + (x_j/\theta_i)^n}, \tag{4}$$

where  $i$  and  $j$  represent the gene interaction pair,  $x_i$  and  $x_j$ ,  $n$  is the Hill coefficient,  $\beta$  is the production rate and  $\theta_i$  is the threshold for gene  $x_i$ . The interactions from  $x_1$  and  $x_3$  to  $x_2$  are combined using summation logic,  $L(x, y) = \frac{1}{2}(x + y)$ .

### 3 Single Objective versus Multi-objective Approach

#### 3.1 Single Objective

**Time Domain.** It is possible to produce self-sustained oscillations in (1), (2) and (3) by reducing the error between the dynamics of only the target gene, described by (2), and a desired oscillatory state of a specified period. As an oscillation can be described using a simple sine wave, we define the desired state of the target gene as

$$x_{tg}^d(t) = \sin\left(\frac{2\pi t}{T}\right), \tag{5}$$

where  $T$  is the period of the oscillator. Using this desired state we define the mean specified time domain fitness,  $f_{t_s}$ , for  $R$  number of runs of each individual as

$$f_{t_s} = \frac{1}{R} \sum_{r=1}^R \sum_{t=0}^N \left(x_{tg}^i(r, t) - x_{tg}^d(t)\right)^2. \tag{6}$$

Here  $r$  is the run number,  $N$  is the number of time steps and  $x_{tg}^i(r, t)$  is the state of the target gene for the  $i$ th generation of run  $r$  at time  $t$ . This mean squared error (MSE) method provides a simple, tunable fitness function in the time domain.

**Frequency Domain.** One difficulty with the MSE method in the time domain is that an individual solution that produces a sustained oscillation that is out of phase with the desired state, given in (5), will lead to a high MSE. This is a consequence of the simple fitness function defined in (6). This high MSE may lead to these individuals being removed from the population, possibly in favour of solutions with a lower MSE value that do not produce oscillations. In order to avoid this potential loss of desired solutions, a fitness function based on the frequency of the oscillation is required. We perform a Fourier Transform on (5) to determine the frequency of the oscillation using the FFTW3 algorithm, as it has been demonstrated to be very efficient for discrete Fourier Transforms [24]. The maximum value in the frequency spectrum denotes the main frequency

component of the wave, which is equal to the integral of the curve for a pure sine wave. For the GRN dynamics there may be some chaotic behaviour initially in the temporal plane so the distribution of frequencies may not be a delta-like function as in the case for a pure wave. Therefore in order to produce a sustained oscillation we first define the desired frequency for the target gene,

$$\omega_{tg}^d = MAX \left\{ \hat{F} \left[ \sin \left( \frac{2\pi t}{T} \right) \right] \right\}, \quad (7)$$

where  $MAX$  denotes the peak in the frequency domain,  $\hat{F}[\sin(2\pi t/T)]$  is the Fourier Transform of the desired state of the target gene from (5). To determine the fitness for solution  $i$ , we apply a Fourier Transform to the target gene and calculated the MSE as in (6),

$$f_{\omega_s} = \frac{1}{R} \sum_{r=1}^R \sum_{t=0}^N \left( \hat{F} [x_{tg}^i(r, t)] - \omega_{tg}^d \right)^2, \quad (8)$$

where  $x_{tg}^i(r, t)$  is the state of the target gene. This is a slightly more complex objective setup than for the time domain, however, due to the efficiency of FFTW3, there is no noticeable effect in computational performance.

### 3.2 Multi-objective

**Time and Frequency Domains.** Here we use the MSE fitness definition for both the time and frequency domains in a multi-objective setup. Here we use two objectives,  $f_1$  and  $f_2$ , which are equivalent to (6) and (8) respectively. These objectives have the same goal due to the inverse relationship between frequency and period. This setup can therefore be considered a decomposition of the single objectives described in Section 3.1, in to a multi-objective setup. This decomposition is referred to as multiobjectivization and has been shown to aid convergence for certain problems [25, 26].

**Time Domain and Non-Specific Fourier Transform.** We also investigate a multi-objective setup which uses a Fourier Transform, but not for a specified frequency. In this method we use the MSE fitness function for the time domain as defined in (6) and a frequency domain fitness different from (8). Here we use the Fourier Transform not to specify a desired frequency, but simply to produce an oscillation. To produce an oscillation of a non-specific frequency we use the following fitness function with no need for a desired state

$$f_{\omega_u} = \frac{1}{R} \sum_{r=1}^R \frac{1}{MAX \{ \hat{g}(r, \omega) \}} \int \hat{g}(r, \omega) d\omega, \quad (9)$$

where  $\hat{g}(r, \omega)$  is the Fourier Transform of the target gene dynamics for the  $r$ th set of random initial conditions,

$$\hat{g}(r, \omega) = \hat{F} [x_{tg}^i(r, t)] . \quad (10)$$

The oscillator is tuned using the time domain fitness function, whereas the Fourier Transform is to improve the oscillator success rates and remove the non-oscillatory solutions. In this setup the two objectives are given by (6) and (9) for  $f_1$  and  $f_2$  respectively. Again this is an example of multiobjectivization as the addition of the frequency objective, although it does not specify a frequency, is a helper function that could aid the algorithm in converging to oscillatory solutions. This could help avoid solutions with a low fitness in the time domain objective, such as an equilibrium that corresponds to the point of highest gradient for the oscillations. As with objective decomposition, multiobjectivization through adding helper functions has been shown to aid convergence [25, 27] and may also provide more non-dominated solutions with no extra cost to functional evaluation [22].

## 4 The Evolutionary Algorithms

To ensure fairness of the comparison of the single and multi-objective fitness setups, we use the same crossover and mutation operators in the single and multi-objective EAs. In addition, both algorithms adopt an elitism strategy. The main difference is that in the single objective optimisation, a deterministic elitism similar to the plus strategy in evolution strategies is adopted, whereas in the multi-objective case, elitist non-dominated sorting operations are used, i.e. NSGA-II [21]. NSGA-II has been shown to be successful at solving a wide range of optimisation problems. In the following, we will present the details of the two EAs.

### 4.1 Genetic Variations

**Simulated Binary Crossover.** The standard recombination operator used in binary genetic algorithms (GAs) is the crossover operator, in which segments are taken from the string of values of the parents to form the offspring. For real-coded GAs, however, a simulated binary crossover (SBX) operation can be used. For parent solutions  $x_i^{(1,t)}$  and  $x_i^{(2,t)}$  to produce offspring solutions  $x_i^{(1,t+1)}$  and  $x_i^{(2,t+1)}$  first a random number,  $u_i$ , between 0 and 1 is chosen and used to determine

$$\beta_{qi} = \begin{cases} (2u_i)^{\frac{1}{n+1}} & \text{if } u_i \leq 0.5 \\ \left(\frac{1}{2(1-u_i)}\right)^{\frac{1}{n+1}} & \text{otherwise .} \end{cases} \quad (11)$$

The offspring solutions are then calculated as follows:

$$x_i^{(1,t+1)} = 0.5 \left[ (1 + \beta_{qi}) x_i^{(1,t)} + (1 - \beta_{qi}) x_i^{(2,t)} \right] , \quad (12)$$

$$x_i^{(2,t+1)} = 0.5 \left[ (1 - \beta_{qi}) x_i^{(1,t)} + (1 + \beta_{qi}) x_i^{(2,t)} \right]. \quad (13)$$

In all simulations we set the recombination probability to 0.9.

**Polynomial Mutation.** For binary coded algorithms, which are encoded by a fixed length string, discrete mutation operations are used [28]. These operations use a mutation probability to determine if the value of the parameter is flipped. However for real-coded GAs, a polynomial mutation operator is used. For these operations the mutation probability is based on the number of dimensions,  $\Delta$ , in the problem  $p_m = \Delta^{-1}$ . The distribution of a spread factor is defined as,

$$P(\beta_{mi}) = 0.5 (n + 1) (1 - |\beta_{mi}|)^n, \quad (14)$$

where  $\beta_{mi}$  is given by,

$$\beta_{mi} = \begin{cases} (2u_i)^{\frac{1}{n+1}} - 1 & \text{if } u_i \leq 0.5 \\ 1 - (2(1 - u_i))^{\frac{1}{n+1}} & \text{otherwise.} \end{cases} \quad (15)$$

If a mutation in the individual occurs, the parameter value is given as

$$x' = x + (\alpha - \delta) \beta_{mi}, \quad (16)$$

where  $\alpha$  and  $\delta$  are the upper and lower bounds for the mutation values respectively. Further details of both the SBX and polynomial mutation operations can be found in [29].

## 4.2 Single-Objective Selection

For the single objective (SO) setup we use an elitist strategy known as  $(\mu + \lambda)$ , where after each generation  $\mu$  parents and  $\lambda$  offspring solutions are combined and ranked in terms of their fitness. The fittest  $\lambda$  solutions of this combination are selected as the parent population for the next generation. We adopt this strategy for both the objective problems given in (6) and (8) and use a population size of 100 for both parent and offspring solutions. This strategy ensures that good solutions are not discarded after each generation, and has been demonstrated to aid convergence in many optimisation problems [17, 21, 30]. It has been widely reported that elitist strategies can lead to premature convergence at local, rather than global, optima for some optimisation problems. However, here we are interested in the production of self-sustained oscillations rather than obtaining a global optimal solution and therefore all solutions that produce oscillatory dynamics are considered successful.

## 4.3 Multi-objective Selection

The two multi-objective (MO) setups described in Section 3.2 are solved using the elitist non-dominated operations in NSGA-II. Here, after fitness evaluation, parent and offspring solutions are combined and sorted into non-dominated

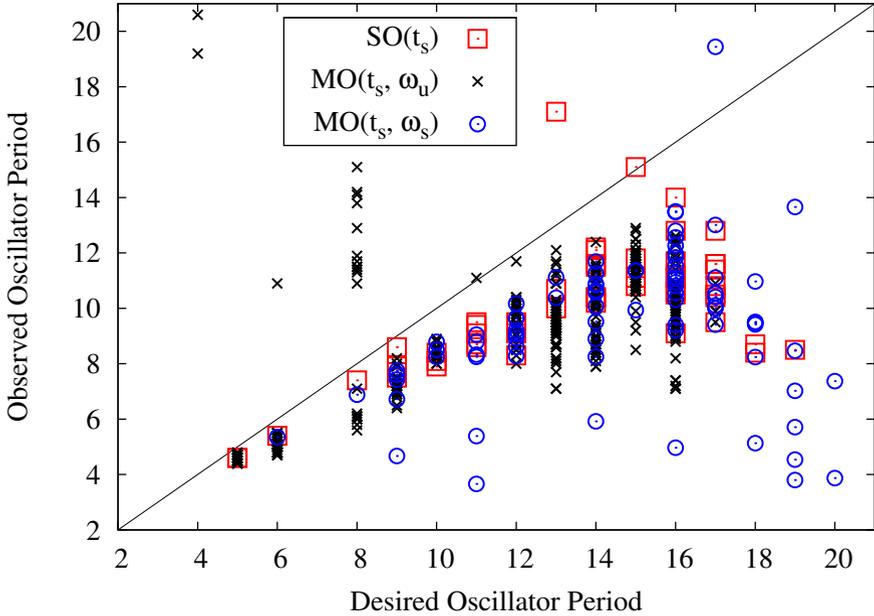
fronts, where front 1 is comprised of non-dominated solutions, front 2 is comprised of solutions that are only dominated by the solutions on front 1, etc. Next the crowding distance operation is applied to all solutions, which is the average distance between a solution and its nearest neighbours on the same non-dominated front. A new population of size  $\mu$  is then filled from the non-dominated fronts starting with the solutions on front 1, then from successive fronts if there are spaces in the new population. If there are more solutions on a front than spaces in the population the most diverse solutions, i.e. those with the largest crowding distance, are selected. Once this new population is full, two randomly selected solutions are compared in a tournament selection, with a low front winning the tournament. In the case that the solutions are from the same front, the solution with the higher crowding distance is selected as the better solution to promote the diversity of the population. The resulting solutions from the tournament selection form the mating pool for the next generation and the crossover and mutation operations described in Section 4.1 are applied to produce the next generation of offspring solutions. Further details on this algorithm and the operations used can be found in [21].

## 5 Result and Analysis

### 5.1 Success Rates: Untuned Oscillators

**Single Objective.** All objective setups are simulated 50 times for different random number seeds to investigate the success rates of the method at producing self-sustained oscillations. We run all simulations for 100 generations. Oscillations, of varying periods, are observed in 31 runs for the time domain fitness setup described in Section 3.1. For the frequency domain setup, also described in Section 3.1, no oscillations of any period were observed. Although the single objective time domain setup can potentially lead to the loss of oscillatory solutions due to phase shifts, it is still successful at producing oscillations. The frequency domain method is not only unable to avoid the potential problem of phase shift in the time domain, it is unable to lead to oscillatory dynamics at all, sustained or damped.

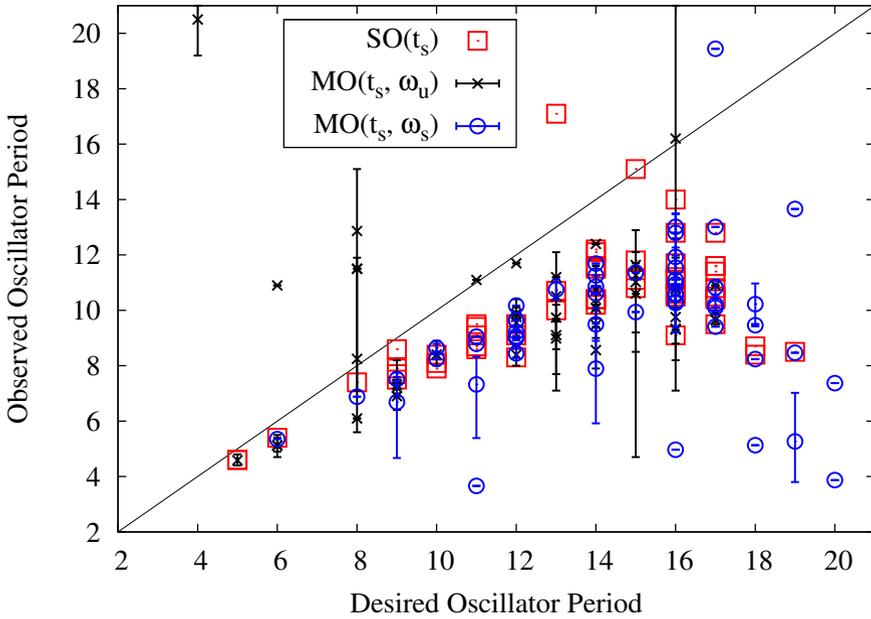
**Multi-objective.** For the multi-objective setups, the combination of the time and frequency domain single objectives described in Section 3.2, lead to 24 observed oscillations out of the 50 test cases. The decrease in observed oscillations may be due to the frequency domain objective, which is unable to produce oscillations unaided. To further test the effect of multiple objectives in oscillation production success rates, we also used the setup of the single objective time domain fitness and an untuned Fourier Transform fitness as described in Section 3.2. Here we observed 22 oscillations of the 50 simulations, showing a slight decrease compared to the other multi-objective setup.



**Fig. 2.** Tunability of the oscillator for single objective (SO) and multi-objective (MO) setups. The objective setup is indicated by  $t_s$ ,  $\omega_s$  and  $\omega_u$  for (6), (8) and (9) respectively. Here Included is a  $y = x$  line for clarity.

## 5.2 Success Rates: Tuned Oscillators

We also investigated the tunability of the oscillator period using the four setups detailed in Section 3. Here we simulate oscillator periods varying from 1 to 20 for 50 test cases. The oscillators are tuned according to (6) and (8), however the observed oscillations produced appear to vary in period for each test case. The periods of the successful oscillations are measured and compared to the desired oscillator period shown in Fig. 2. Here we compare the observed oscillator periods for all setups against the desired periods. The single objective frequency domain method is not included as there were no observed successful oscillations. The multi-objective (MO) setups provide more solutions than the single objective (SO) setup because the MO simulations contain an archive of solutions which cannot be compared directly and so all oscillatory solutions are plotted. For a given MO simulation there may be several solutions in the archive that produce oscillations, but not necessarily all. This leads to a range of observed periods for each simulation at each desired period. For MO simulations with more than one oscillatory solution, the observed periods are averaged to give the datum point in Fig. 3. The error bars are the maximum and minimum periods observed in that simulation and represent the spread of that values. For the MO simulations that only have one successful solution a single point only is included.



**Fig. 3.** Averages for multi-objective (MO) results. The MO points represent the average oscillator periods and the error bars represent the maximum and minimum values for each simulation; see text for details.

These results indicate that neither the single or multiple objective setups are tunable for the small three gene configuration, Fig. 1, used here. The SO setup in the time domain appears to be tunable for low period oscillations  $T \leq 9$ . At higher oscillator periods however, this setup becomes less tunable and begins to diverge from the desired period. The MO methods both demonstrate a large spread of periods for successful runs. Though the average values appear to follow a similar pattern to the single objective setup, with the exception of a few outliers, this does not indicate a tunable oscillator. Average values are used here as there is no direct way to compare archive solutions from the MO setup and so we cannot say if a solution is better than another without further analysis. Even when including all MO archive solutions only two solutions fall on the line indicating the observed period is the desired period in Fig. 2. These two solutions are at  $T = 11$  and  $T = 12$  and are the only oscillatory solutions from the archive for their respective runs. This is represented by the fact that they do not have bars on the values in Fig. 3. It is however worth noting that the SO setup was unable to produce an oscillation at the period.

### 5.3 Discussion

No objective setup investigated in this work is able to reliably produce tunable oscillations. One notable observation here was the inability for a simple frequency based objective function to produce an oscillation when an equivalent

time domain objective is successful. This may be a result of the desired state used as the Fourier Transform of a pure sine wave will produce a single peak in the frequency domain. However the dynamics of a randomly initialised GRN may initially be completely different and thus the EA is not able to find a parameter set that is able to produce a pure sine wave and therefore a single peak in the frequency plane. When considering these aspects, and that the fitness only determines the location of the peak with no width constraints, this objective is more complex than the time domain objective. This is in contrast to (9), where the width constraint comes from the minimisation of the MSE and therefore the area of the curve. Thus, for the specified frequency objective, very broad peaks that have low maximum values are considered good solutions if they peak at the correct frequency regardless of the rest of the frequency profile. Adding other objectives such as minimising the width of the peak could also be used. However incorporating this into an objective that also specifies the location of the peak, and thus the frequency, may be nontrivial. Further investigation in to the frequency domain objective setup is needed in order to tune the oscillator.

The addition of a second objective based on an unspecified frequency, (9), leads to a spread in the observed oscillator period. This is a consequence of the second objective not specifying an oscillator period. Solutions that are oscillatory, but not of the required period, will be retained if the value of (9) is low despite the value of the other objective, (6). This gives an indication as to why there is a large spread of observed oscillations and is likely to be a consequence of the selection operations in NSGA-II. The crowding distance operation used in the non-dominant sorting part of NSGA-II favours more diverse solutions to provide a wide-spread Pareto front. This, however, will ultimately lead to solutions with a low value for the unspecified frequency objective, (9), even at the cost of a more optimal time domain objective. Thus in the Pareto optimal solutions there may be many different oscillator periods representing the optimal solutions for the unspecified frequency based objective. The addition of this objective has however aided the convergence to tuned oscillators. It is able to produce solutions that are close to the desired period for a larger range than the SO setup using the same time domain objective. The improvement can be seen for periods  $T = 5, 6, 10, 11, 12$  and  $14$ , with only a slight decrease in desired period for  $T = 8$  and  $9$  compared with the SO setup (see Fig. 2). However there are potentially more solutions for this setup due to the Pareto front, thus more analysis is required for this method than for a SO method.

In comparison with the above setups, the MO method, which specifies the period and frequency, should help remove the spread of observed periods. However, this is not the case, and this setup performs worse with increasing oscillator period. This cause of failure is likely to be the same as the SO specific frequency setup. However it is interesting that the addition of another objective, that specifies the period, enables this set to produce oscillations, though does not improve tunability. This demonstrates that the decomposition of a SO problem into a MO one can aid convergence for the case of the SO frequency domain objective compared with the MO time and specified frequency domain setups. Here the

SO frequency domain was unable to produce any oscillations but the addition of a time domain objective enabled oscillatory dynamics to be produced from the GRN.

## 6 Conclusions and Future Work

Both the decomposition of a single objective problem into a multiple objective problem and the use of a helper function have effects on the convergence of the problem investigated here. For the case of objective decomposition, we find that the addition of a specified frequency based method to a time based setup does not aid convergence, and performs worse in some cases. However for the case of adding a time based objective to a frequency based setup, objective decomposition enables the evolution of oscillations though does not aid tunability. We also observe that the use of an unspecified frequency based helper objective to a time based objective shows improvements in tunability at some periods compared with the single objective setup. This indicates that the effect of multiobjectivization on performance is dependent on the problem and a general statement on the effects cannot be made, which was also observed in [22, 25]. The results here also indicate that the effects of multiobjectivization may also depend on the objective setup and fitness domain used in the optimisation problem.

Objective setup in the *in silico* evolution of oscillatory dynamics of gene regulatory networks requires much more investigation, particularly the single objective frequency domain objective. Further investigation into objective setup could lead to a tunable oscillator using a small gene regulatory networks and evolutionary algorithms. Network size and loop combination have also been shown to be important in tuning oscillators [13], therefore a wider range of gene regulatory network structures, such as gene number, logic functions, auto-regulation and feedback setups should also be investigated.

**Acknowledgment.** This research is funded by an EPSRC DTC studentship.

## References

- [1] Alon, U.: Network motifs: theory and experimental approaches. *Nat. Rev. Genet.* 8, 450–461 (2007)
- [2] Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: Simple building blocks of complex networks. *Science* 298(5594), 824–827 (2002), <http://www.sciencemag.org/content/298/5594/824.abstract>
- [3] Jin, Y., Sendhoff, B.: Evolving *in silico* bistable and oscillatory dynamics for gene regulatory network motifs. In: *IEEE World Congress on Computational Intelligence Evolutionary Computation, CEC 2008*, pp. 386–391 (June 2008)
- [4] Gonze, D.: Coupling oscillations and switches in genetic networks. *Biosystems* 99(1), 60–69 (2010), <http://www.sciencedirect.com/science/article/pii/S030326470900149X>
- [5] Goldbeter, A.: *Biochemical Oscillations and Cellular Rhythms*, vol. 1. Cambridge University Press (April 1997), <http://dx.doi.org/10.1017/CB09780511608193>

- [6] Chay, T.R.: A model for biological oscillations. *Proceedings of the National Academy of Sciences of the United States of America* 78, 2204–2207 (1981), <http://ukpmc.ac.uk/abstract/MED/6264468>
- [7] Francis, M.R., Fertig, E.J.: Quantifying the dynamics of coupled networks of switches and oscillators. *PLoS ONE* 7(1), e29497 (2012), <http://dx.doi.org/10.1371/journal.pone.0029497>
- [8] Rempe, M., Best, J., Terman, D.: English A mathematical model of the sleep/wake cycle. *Journal of Mathematical Biology* 60, 615–644 (2010), <http://dx.doi.org/10.1007/s00285-009-0276-5>
- [9] Ennes, R.H., McGuire, G.C.: *Nonlinear Physics with Mathematica for Scientists and Engineers*. Birkhäuser, Boston (2001)
- [10] Fall, C.P., Marland, E.S., Wagner, J.M., Tyson, J.J.: *Computational Cell Biology*. In: Antman, S.S., Marsden, J.E., Sirovich, L., Wiggins, S. (eds.), vol. 20. Springer (July 2002)
- [11] Berridge, M., Rapp, P.: A comparative survey of the function, mechanism and control of cellular oscillators. *The Journal of Experimental Biology* 81, 217–279 (1979), <http://europemc.org/abstract/MED/390080>
- [12] Jin, Y., Meng, Y., Sendhoff, B.: Influence of regulation logic on the easiness of evolving sustained oscillation for gene regulatory networks. In: *IEEE Symposium on Artificial Life, ALife 2009*, pp. 61–68 (April 2009)
- [13] Tsai, T.Y.-C., Choi, Y.S., Ma, W., Pomeroy, J.R., Tang, C., Ferrell, J.E.: Robust, tunable biological oscillations from interlinked positive and negative feedback loops. *Science* 321(5885), 126–129 (2008), <http://www.sciencemag.org/content/321/5885/126.abstract>
- [14] Alon, U.: *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall/CRC (2006)
- [15] Ito, S., Izumi, N., Hagihara, S., Yonezaki, N.: Qualitative analysis of gene regulatory networks by satisfiability checking of linear temporal logic. In: *Proceedings of the 2010 IEEE International Conference on Bioinformatics and Bioengineering*, ser. BIBE 2010, pp. 232–237. IEEE Computer Society, Washington, DC (2010), <http://dx.doi.org/10.1109/BIBE.2010.45>
- [16] Chu, D.: Evolving genetic regulatory networks for systems biology. In: *IEEE Congress on Evolutionary Computation, CEC 2007*, pp. 875–882 (September 2007)
- [17] Jin, Y., Meng, Y.: Emergence of robust regulatory motifs from in silico evolution of sustained oscillation. *Biosystems* 103(1), 38–44 (2011), <http://www.sciencedirect.com/science/article/pii/S0303264710001693>
- [18] Sirbu, A., Ruskin, H.J., Crane, M.: Comparison of evolutionary algorithms in gene regulatory network model inference. *BMC Bioinformatics* 11, 59 (2010)
- [19] Thomas, S.A., Jin, Y.: Combining genetic oscillators and switches using evolutionary algorithms. In: *2012 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pp. 28–34 (May 2012)
- [20] Handl, J., Kell, D.B., Knowles, J.: Multiobjective optimization in bioinformatics and computational biology. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* 4(2), 279–292 (2007), <http://dx.doi.org/10.1109/TCBB.2007.070203>
- [21] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197 (2002)
- [22] Handl, J., Lovell, S.C., Knowles, J.D.: Investigations into the Effect of Multiobjectivization in Protein Structure Prediction. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) *PPSN X. LNCS*, vol. 5199, pp. 702–711. Springer, Heidelberg (2008), [http://dx.doi.org/10.1007/978-3-540-87700-4\\_70](http://dx.doi.org/10.1007/978-3-540-87700-4_70)

- [23] Silva-Rocha, R., de Lorenzo, V.: Noise and robustness in prokaryotic regulatory networks. *Annual Review of Microbiology* 64(1), 257–275 (2010), <http://www.annualreviews.org/doi/abs/10.1146/annurev.micro.091208.073229>
- [24] Frigo, M., Johnson, S.G.: The design and implementation of FFTW3. *Proceedings of the IEEE* 93(2), 216–231 (2005); Special issue on “Program Generation, Optimization, and Platform Adaptation
- [25] Brockhoff, D., Friedrich, T., Hebbinghaus, N., Klein, C., Neumann, F., Zitzler, E.: Do additional objectives make a problem harder?”. In: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO 2007, pp. 765–772. ACM, New York (2007), <http://doi.acm.org/10.1145/1276958.1277114>
- [26] Knowles, J.D., Watson, R.A., Corne, D.W.: Reducing Local Optima in Single-Objective Problems by Multi-objectivization. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D.W. (eds.) *EMO 2001*. LNCS, vol. 1993, pp. 269–283. Springer, Heidelberg (2001), <http://dl.acm.org/citation.cfm?id=647889.736521>
- [27] Jensen, M.T.: Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation. *Journal of Mathematical Modelling and Algorithms* 3, 323–347 (2004), <http://dx.doi.org/10.1023/B:JMMA.0000049378.57591.c6>, doi:10.1023/B:JMMA.0000049378.57591.c6
- [28] Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston (1989)
- [29] Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. *Complex Systems* 9, 115–148 (1995)
- [30] Mendoza, M.R., Bazzan, A.L.C.: Evolving random boolean networks with genetic algorithms for regulatory networks reconstruction. In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO 2011, pp. 291–298. ACM, New York (2011), <http://doi.acm.org/10.1145/2001576.2001617>