

The Effects of Interaction Frequency on the Optimization Performance of Cooperative Coevolution

Elena Popovici
George Mason University
4400 University Dr MSN 2A1
Fairfax, VA 22030, USA
epopovic@gmu.edu

Kenneth De Jong
George Mason University
4400 University Dr MSN 2A1
Fairfax, VA 22030, USA
kdejong@gmu.edu

ABSTRACT

Cooperative coevolution is often used to solve difficult optimization problems by means of problem decomposition. Its performance on this task is influenced by many design decisions. It would be useful to have some knowledge of the performance effects of these decisions, in order to make the more beneficial ones. In this paper we study the effects on performance of the frequency of interaction between populations. We show them to be problem-dependent and use dynamics analysis to explain this dependency.

Categories and Subject Descriptors

I.2.m [Artificial Intelligence]: Evolutionary Computation

General Terms

Algorithms

Keywords

cooperative coevolution, dynamics, performance

1. INTRODUCTION

The idea of *cooperative coevolution* as a method for static function optimization was introduced by Potter in [12]. Initial results were promising, thus a framework for using cooperative coevolutionary algorithms was developed in [11] and then further extended in [13]. The latter work put together a hierarchical categorization of coevolutionary algorithm (CoEA) properties, pointing out the many knobs that can be adjusted in these algorithms.

A practitioner trying to use a cooperative CoEA for optimization is thus faced with a number of design decisions, such as what EA to use in each population and how to make the populations interact. It would be useful to have some knowledge of the effects of these choices, in order to make the more beneficial ones. In recent years, coevolution research has invested more effort in generating such knowledge.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '06, July 8–12, 2006, Seattle, Washington, USA.
Copyright 2006 ACM 1-59593-186-4/06/0007 ...\$5.00.

Most studies ([3], [14], [15], [5], [7]) were focused on collaboration schemes, namely how many individuals to use for evaluation, how to select these individuals and how to aggregate the outcomes of the interactions with them. In [4] run-time analysis was used to investigate the performance effects of update timing (i.e., whether the populations evolve simultaneously or take turns). The same parameter was analyzed in a different context in [10]. An initial study on the performance effects of population size and elitism was performed in [8] and then further extended in [9].

In this paper we analyze the performance effects of the frequency of interaction between populations. While previous work ([1], [2]) used various choices for this parameter, no systematic study of the effects of such choices was performed. Our hypothesis is that the performance effects are dependent on a problem property called *best-response curves*, which was introduced in [8]. We confirm this hypothesis on a family of synthetic functions from [9]. We explain the causes for the observed dependency by analyzing the run-time behavior (dynamics) of the algorithm. We then use what we have learned from the analysis of the synthetic functions to successfully predict performance on three functions common in the evolutionary computation (EC) optimization literature.

2. EXPERIMENTAL SETUP

2.1 The Domains

We start by using a family of test functions from [9], defined as follows:

$$BR_n^\alpha(x, y) = \begin{cases} 2y + \frac{\alpha-3}{2\alpha}(x-n) & \text{if } \alpha y < x + (\alpha-1)n; \\ 2x + \frac{\alpha-3}{2\alpha}(y-n) & \text{if } y > \alpha x + (1-\alpha)n; \\ n + \frac{x+y}{2} & \text{otherwise.} \end{cases}$$

$$n \in \mathbf{N}; \alpha \in [0, 1]; x, y \in [0, n].$$

The two-dimensional surfaces described by the functions for $n = 10$ and $\alpha \in \{0, 0.25, 0.90, 1\}$ are shown in Figure 1. BR_n^1 and $BR_n^{0.75}$ were initially introduced in [8] as *oneRidge_n* and *twoRidges_n*. Indeed, while at $\alpha = 0$ the surface is just a plane, for $\alpha \in (0, 1)$ two ridges appear and they get closer and closer as α increases. At $\alpha = 1$ the two ridges merge into one. For these functions the task is maximization and regardless of α there is a unique maximum $BR_n^\alpha(n, n) = 2n$.

These functions were constructed to illustrate a problem property called *best-response curves*, that was shown to highly influence the effect on optimization performance of parameters such as population size, elitism [8] and collaboration schemes [7]. Our hypothesis is that they also influence the performance effects of the interaction frequency and, as section 3 will show, the experiments confirm this hypothesis.

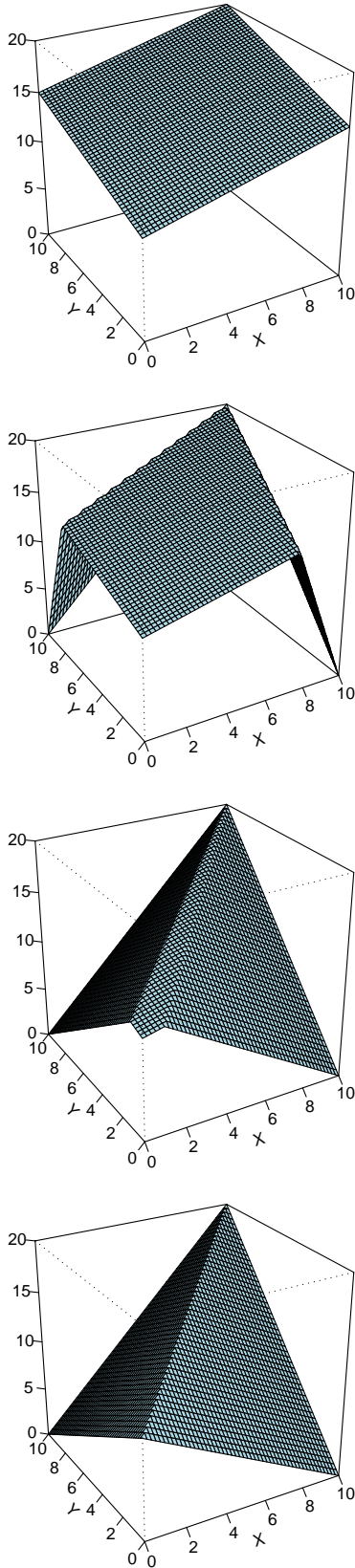


Figure 1: BR_{10}^α . Top to bottom, $\alpha = 0, 0.25, 0.90, 1$.

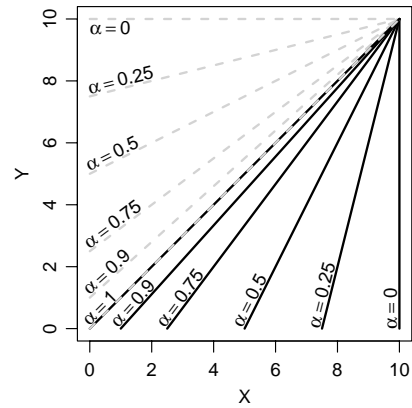


Figure 2: Best-response curves for BR_{10}^α . Black continuous lines denote $bestResponseX$ and gray dashed lines denote $bestResponseY$.

We give here a brief description of best-response curves; for more details see [8] and [7]. If $f : D_X \times D_Y \rightarrow \mathbf{R}$ is the function to maximize, we define $bestResponseX : D_Y \rightarrow D_X$, $bestResponseX(y_0) = \operatorname{argmax}_{x \in D_X} f(x, y_0)$. In other words, it returns for any y the x that produces the best function value in combination with that y . $bestResponseY$ is similarly defined. In this paper we deal only with functions for which the best response for any value is unique.

The formulas for the best-response curves of BR_n^α are:

$$bestResponseX(y) = \alpha y + (1 - \alpha)n \text{ and}$$

$$bestResponseY(x) = \alpha x + (1 - \alpha)n.$$

These formulas describe two lines that intersect in (n, n) , which is where the optimum is. Figure 2 plots them for $n = 10$ and several values of α . At $\alpha = 0$ the two best-response lines are perpendicular; as α increases, the angle between them decreases, till finally they overlap when $\alpha = 1$. The value of the function along any such line decreases from $2n$ in (n, n) as we move down and towards the left (i.e., towards smaller x and y values).

2.2 The Algorithm

We use a fairly standard two-population CoEA as our basic setup and then vary the interaction frequency and observe the changes in performance. One population evolves values for the x parameter and the other for the y parameter. In both cases individuals are single-gene real-valued numbers. In each population we run a non-overlapping generational EA with elitism of 1, tournament selection of size 2 and Gaussian mutation with sigma fixed to 0.2 altering each gene (therefore individual) with a probability of 0.90. The size of each population is 20. For evaluation we use the *single best collaboration strategy* ([12], [13]), namely, the fitness of an individual in one population is equal to the value of the function in the point obtained by coupling that individual with the latest best individual communicated by the opposite population.

The fact that the populations interact (communicate information) for evaluation purposes is the main feature of CoEAs. We call an interaction point a point in evolutionary time at which the populations interact. In this paper we only look at cases where the time between any two consecutive

interaction points is the same. We call this period of time an epoch. We vary the epoch size as a means to control the frequency of interaction (bigger epoch size equals smaller frequency and vice versa). Because we use a generational EA, we measure the epoch size in generations.

We use a *sequential update timing* [13], meaning that the populations take turns in evolving. During each epoch only one population is active, while the other one is frozen. At the end of the epoch (at the interaction point), the population that was evolving communicates its best individual to the population that was frozen and then they switch roles. The previously frozen population will be active during the following epoch and at the end of it, it will report its best individual to the other population.

At the beginning of the evolutionary process, each population is initialized uniformly random across its domain. The evolution starts with an interaction point; one population communicates a random individual (since it hasn't been evaluated yet, it doesn't have a best) to the opposite population, which is the first to become active.

For fair comparisons between interaction schemes, we keep the number of evaluations constant across experiments. One thing to note is that at the beginning of a new epoch, a population that just received information, if it is to incorporate it immediately, it should re-evaluate all its individuals. In particular, when using a generational EA, during an epoch that is n generations long, an active population requires $(n + 1) * m$ evaluations, where m is the population size. A sequential setting with k epochs requires $(n + 1) * m * k$ evaluations, as there is only one population active per epoch.

We perform experiments with 8 settings for the epoch size, namely 1, 2, 3, 5, 8, 11, 17 and 26 generations. With a fixed budget of 2160 function evaluations, this means running for 54, 36, 27, 18, 12, 9, 6 and respectively 4 epochs (e.g., for epoch size 3 we have $(3 + 1) * 20 * 27 = 2160$). For each setting we perform 100 independent runs, 50 of which start with the X population active and 50 with the Y population active. We then repeat all of this 15 times for $\alpha \in \{0, 0.2, 0.25, 0.4, 0.5, 0.6, 0.75, 0.8, 0.9, 0.94, 0.95, 0.96, 0.98, 0.99, 1\}$. Due to space constraints, here we present data only from 9 of these settings for α , namely 0, 0.2, 0.4, 0.6, 0.75, 0.9, 0.96, 0.98 and 1. However, the results for the omitted values fit well in the trend described by the values showed.

2.3 The Results

Figure 3 summarizes performance of the 8 epoch size settings for the 9 mentioned α values. It shows boxplots¹ of best of run fitness collected over 100 independent runs. Although for any α BR_n^α has the same range, we zoom in on a different fitness interval for each plot in order to see the differences between epoch sizes.

As we increase α from 0 to 1, we observe a gradual change in the performance effects of the epoch size. We start (for $\alpha = 0$) with an upward-only slope for performance as we increase epoch size from 1 to 26. At $\alpha = 0.2$, the performance “curve” is first going upward, then level and then downward. As we further increase α , the upward part of the performance curve gradually becomes shorter until it van-

¹The boxplot format permits a concise, comparative visualization of the median (center line), the 95% confidence interval for the median (notch around the median line), the interquartile range (box), the outliers (circles) and the spread of the remaining data (dotted lines and whiskers).

ishes (at about $\alpha = 0.9$), while the downward part becomes more and more pronounced. From there to $\alpha = 1$ we see a more rapid change, with the downward part flattening out through performance decrease for low epoch sizes. A close look at the y -axis ranges of the plots shows an overall decrease in performance with increasing α . This means BR_{10}^α becomes a harder problem for our CoEA as α increases.

While each particular plot in Figure 3 nicely portrays the trend of changing performance with increasing epoch size, it is hard to tell from such a plot whether any two particular epoch sizes generate a statistically significant difference in performance. To achieve that, we use a different plotting technique portrayed in Figure 4. Each plot in this figure represents one α value and color-codes the results of all pairwise comparisons of epoch sizes. A square corresponding to epoch sizes es_i and es_j is:

- gray, if we cannot statistically significantly distinguish between the performance of es_i and that of es_j ;
- black, if there is a statistically significant difference in performance between es_i and es_j and the smaller epoch size ($\min(es_i, es_j)$) performs better; and
- white, if there is a statistically significant difference in performance between es_i and es_j and the bigger epoch size performs better (i.e., the smaller epoch size performs worse).

This definition is symmetrical, since $\min(es_i, es_j) = \min(es_j, es_i)$, therefore we only display the upper left triangle. To test statistically significant difference of medians we use the Wilcoxon test combined with Boole's inequality (the confidence level for every individual square is 99.64% for a total confidence of 90% for each image).

The image for $\alpha = 0$, for example, tells us that epoch size 1 performs worse than any other. It also tells us that epoch sizes 2 and 3 perform worse than 17, but nothing else can be distinguished. Thus, there is a benefit in increasing epoch size from 1 to 2, but to get yet another boost in performance, we would have to increase the epoch size all the way up to 17. For $\alpha = 0.2$ the black squares in the image tell us that epoch size 26 performs worse than anything else and epoch size 17 performs worse than 3 and 8. And while we cannot distinguish between 3, 5 and 8 due to gray, nor between 1 and 2, the white squares tell us that any of 3, 5 and 8 perform better than 1, and 8 also performs better than 2.

Making a parallel with Figure 3, white on the left tells us there is an upward part in the performance slope, gray triangles above the diagonal denote a flat part and black on the top shows the existence of a downward part. Reading Figure 4 along increasing α , we see that we start with some white on the left and a lot of gray, then the white and the gray areas start diminishing, as the black moves in from the top; the white finally disappears, after which the gray area starts growing again and the black area shrinks until it disappears as well. If we use the above “translation”, we discover the same message as conveyed by Figure 3, but now it is backed up by statistical significance.

Our hypothesis is that these performance effects of the interaction frequency are due to the best-response curves. The dynamics analysis in the following section confirms this hypothesis and sheds light on the observed phenomena.

3. DYNAMICS ANALYSIS

[6] introduced a technique of analyzing the dynamics of CoEAs based on best-of-generation trajectories and best-response curves. That research showed that the various pa-

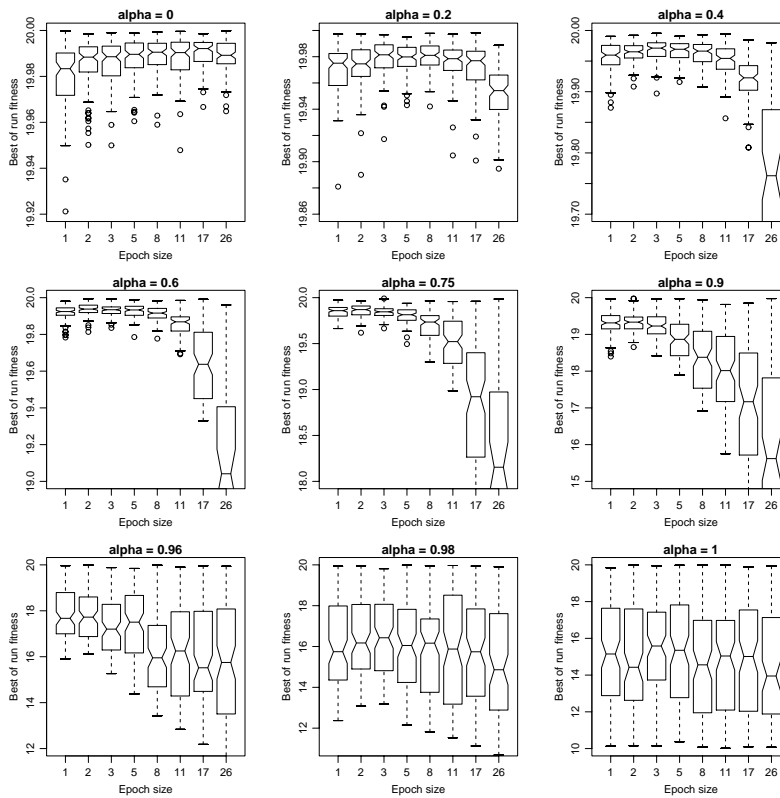


Figure 3: Best of run statistics. Maximization problems: bigger is better.

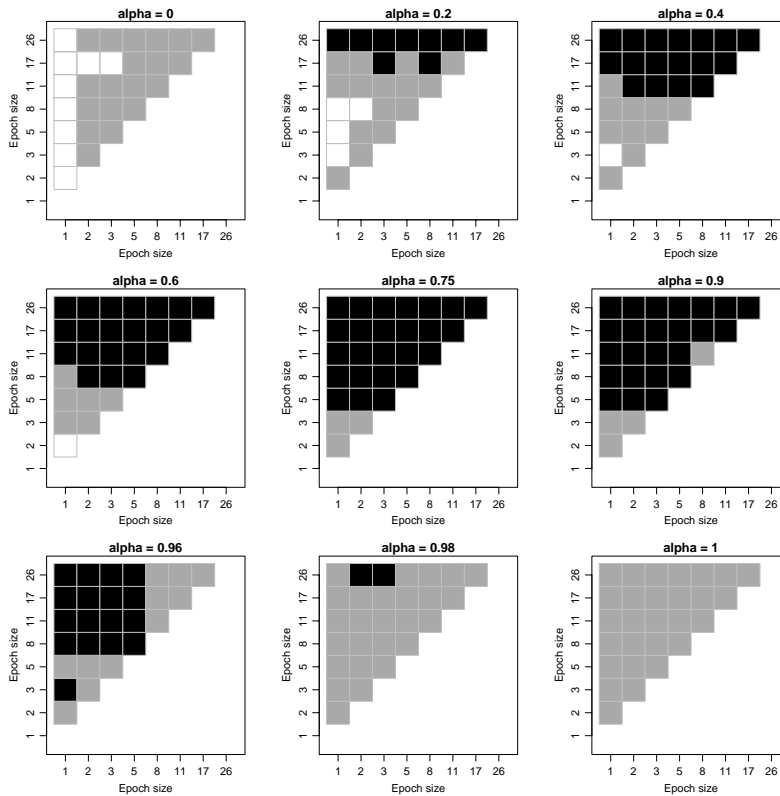


Figure 4: Statistical significance of differences between epoch sizes' performance. White - bigger epoch size performs better; black - smaller epoch size performs better; gray - undistinguishable performance.

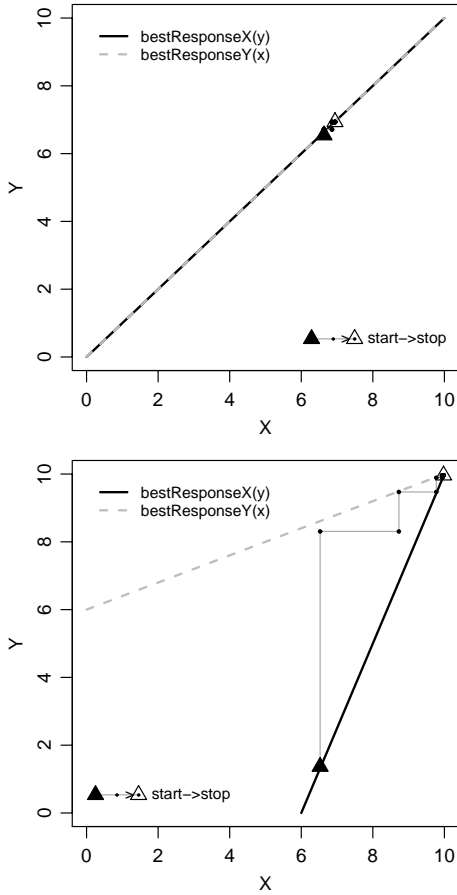


Figure 5: Examples of best-of-epoch trajectories and best-response curves. Top: $\alpha = 1$, epoch size 1. Bottom: $\alpha = 0.4$, epoch size 3.

rameters determined the degree to which the trajectories of best-of-generation individuals followed the best-response curves. This in turn influenced the areas of the search space visited by the algorithm, and thus its performance.

We perform an analysis similar to the ones in [8], [7] and [9] in order to investigate the performance effects of interaction frequency. However, in this case we are concerned with best-of-epoch individuals and their trajectories through the search space. We construct these trajectories as follows. At the end of an epoch during which the X population was active, we plot the best x individual in combination with the y individual used for its evaluation. In this case, due to the single best collaboration scheme, that y is the currently known best individual of the opposite population. At the end of the next epoch, during which the Y population is active, we plot the new best y coupled with the previous best x . We do this for every epoch and connect the points chronologically. It is straightforward to see that the trajectory thus obtained contains only vertical lines (when connecting an X epoch with the following Y epoch) and horizontal lines (when connecting a Y epoch with the following X epoch).

We then combine best-response curves with best-of-epoch trajectories. Figure 5 shows examples of best-of-epoch trajectories for individual runs, superimposed on best-response

curves. Each small black dot represents an interaction point as described above and the thin dark-gray lines connect these points chronologically. The first epoch is marked with a filled triangle and the last epoch with an empty triangle. It is immediately apparent that the trajectories are highly influenced by the best-response curves.

[8], [7] and [9] showed that optimization performance is tightly correlated with the three factors below, the first one being problem dependent and the latter two algorithm dependent:

- the relative positions of the best-response curves (mainly whether or not they overlap);
- the accuracy with which the best-of-generation trajectories follow the best-response curves; and
- the length of the trajectories (i.e., the number of interaction points).

In the case of $\alpha = 1$, high accuracy in following the best-response curves is bad for performance, since it causes the trajectory to quickly get stuck in a point on the overlap line (as can be seen in the top of Figure 5), which may or may not be close to the optimum (depending on the starting position). For $\alpha \in (0, 1)$, high accuracy causes the trajectory to climb like on a ladder towards the optimum. However, it constrains the size of the trajectory's steps to the distance between the best-response curves. The closer these are, the smaller the highly accurate steps will be. Low accuracy allows for jumps larger than the distance between the best-response curves, although smaller jumps may occasionally occur as well. Thus, high accuracy is beneficial when the best-response curves are at a big angle, but it becomes detrimental as the angle between them gets smaller.

When the trajectory is not stuck, a bigger length (more steps) will give it more time to get closer to the optimum. However, with a fixed budget, more steps usually imply smaller per-step accuracy. When the trajectory is stuck, more steps will not help, but won't hurt either.

Clearly, since trajectory length is measured in interaction points, increasing the epoch size decreases trajectory length. Intuitively, increasing the epoch size should have the effect of increasing trajectory accuracy. One way of testing this is to visually inspect trajectories. Figure 6, portraying epoch sizes 1 and 11 for $\alpha = 0.8$ seems to suggest that is the case. However, we would like a quantitative way of testing this.

For that purpose, we define two metrics that compute distance from the best-response curves:

$$brDistX(x, y) = |x - bestResponseX(y)| \text{ and}$$

$$brDistY(x, y) = |y - bestResponseY(x)|.$$

We compute $brDistX$ at interaction points marking the end of an X epoch and $brDistY$ at interaction points marking the end of a Y epoch. This gives us a measure of the accuracy of trajectories. We use it to compare the accuracy induced by different epoch sizes.

Figure 7 shows statistics of $brDistY$ ($brDistX$ behaves similarly). Each plot portrays for a certain α the accuracies for epoch sizes 1, 5 and 17. At each interaction point marking the end of a Y epoch, we plot the mean over 50 runs² of $brDistY$ for that interaction point, together with the 95% confidence interval for that mean. These plots confirm our intuition that increasing epoch size increases the accuracy of the best-of-epoch trajectories.

²In this case the ones starting with an X generation. Similar plots are obtained when averaging over the 50 runs that start with a Y generation.

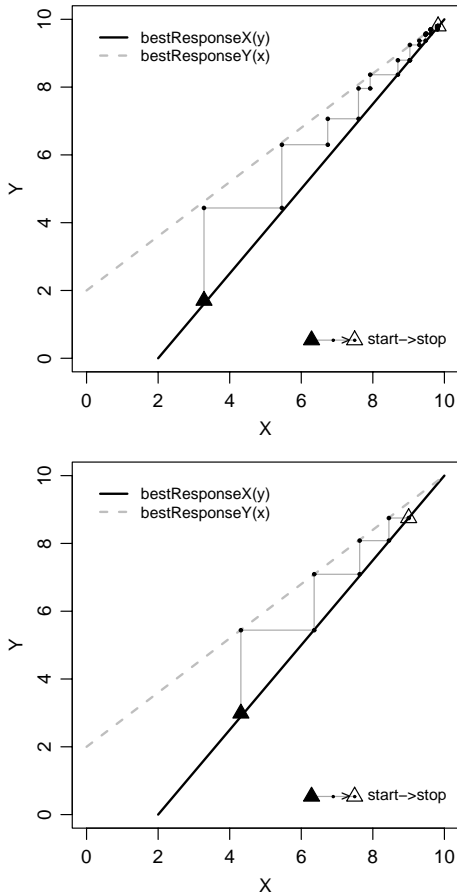


Figure 6: Higher epoch size implies fewer interaction points and also appears to increase the accuracy of following the best-response curves. $\alpha = 0.8$. Top: epoch size 1. Bottom: epoch size 11.

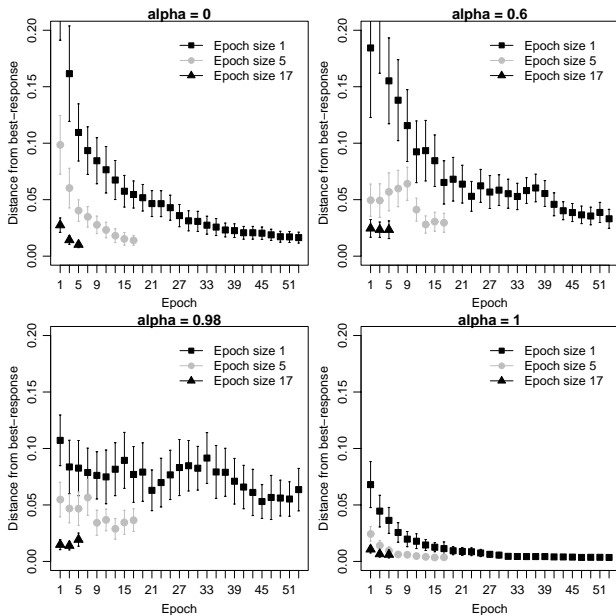


Figure 7: Accuracy in following the best-response curves.

Combining this new knowledge with the previous heuristics about how best-response curves, trajectory accuracy and trajectory length affect performance, we can now explain the results displayed by Figures 3 and 4.

At $\alpha = 0$, as the best-response curves are perpendicular, a deterministic system exactly following them would reach the optimum in just two steps. Even with an epoch size of 26 generations, the trajectories are allowed 4 steps. Thus the trajectory length plays a less important role and it is accuracy that brings in performance benefits. As we increase α , we decrease the angle between the best-response curves, and more steps are needed in order to get close to the optimum, therefore trajectory length becomes increasingly more important. As long as increasing the epoch size increases accuracy while still keeping the trajectory length above what is needed to get close to the optimum, we see improvements in performance (the upward part of the slope). Then there is a range in epoch sizes for which accuracy and trajectory length counter-balance each other and we see a flat performance trend. After that, the disadvantage of having a short trajectory overcomes the benefits of high accuracy and we see diminishing performance.

As the best-response curves get closer, we transition into the phase where high accuracy is detrimental. Thus, increasing the epoch size becomes bad for performance both through high accuracy and through short trajectory length.

At $\alpha = 1$, even the accuracy of epoch size 1 is extremely high (as can be seen in the bottom right plot of Figure 7), and so the trajectory already gets stuck. The best of run essentially has the same distribution as the best of the first epoch, namely uniform over the interval $[10, 20]$. Increasing epoch size can't make things worse than they already are.

This in depth analysis of the dynamics of best individuals confirms our hypothesis: best-response curves *do* influence the effects of the interaction frequency on performance.

4. PREDICTIVE POWER

We now test to see whether what we have learned on the BR_n^α family can be applied to other functions. In particular, what we would like to do is to predict the performance effects of the interaction frequency on various problems just by looking at their respective best-response curves.

We do this on three familiar test functions from the EC function optimization literature, *rastrigin*, *rosenbrock* and *offAxisQuadratic*. These were previously studied both in standard EC and in CoEC settings [14], [7], [9]. We reproduce them below for completeness:

$$\begin{aligned} \text{rastrigin}(x, y) &= 6 + x^2 + y^2 - 3\cos(2\pi x) - 3\cos(2\pi y), \\ &x, y \in [-5.12, 5.12]; \\ \text{offAxisQuadratic}(x, y) &= x^2 + (x + y)^2, \\ &x, y \in [-65.536, 65.536]; \\ \text{rosenbrock}(x, y) &= 100(x^2 - y)^2 + (1 - x)^2, \\ &x, y \in [-2.048, 2.048]. \end{aligned}$$

Our task in this case is minimization. All three functions have a minimum value of 0, attained in a unique point, namely $(0, 0)$ for *rastrigin* and *offAxisQuadratic*, and $(1, 1)$ for *rosenbrock*.

Figure 8 plots the best-response curves for the three functions³. We see that for *rastrigin* they look very similar

³The definitions of the best-response curves are updated for minimization. For the mathematical details of determining their formulas, see [9].

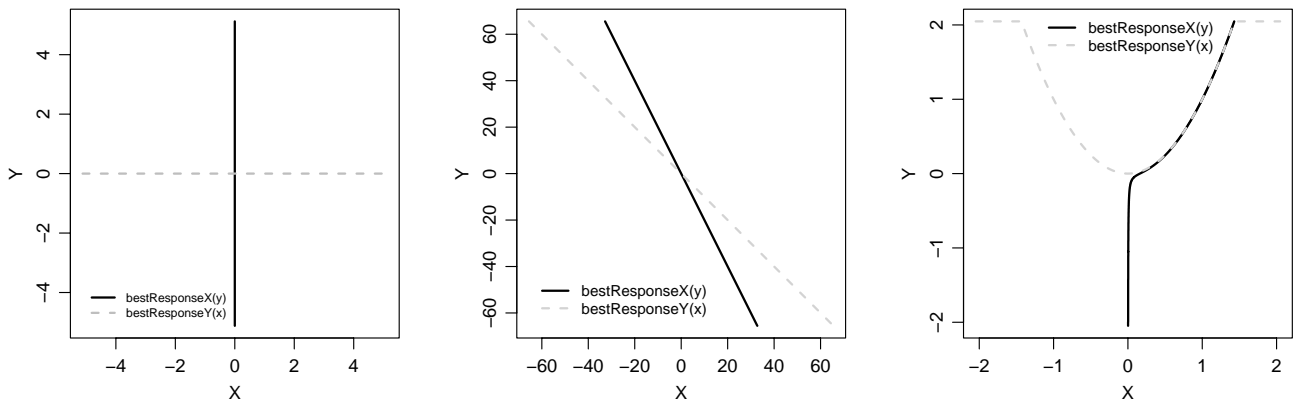


Figure 8: Best-response curves. Left: *rastrigin*, middle: *ofAxisQuadratic*, right: *rosenbrock*.

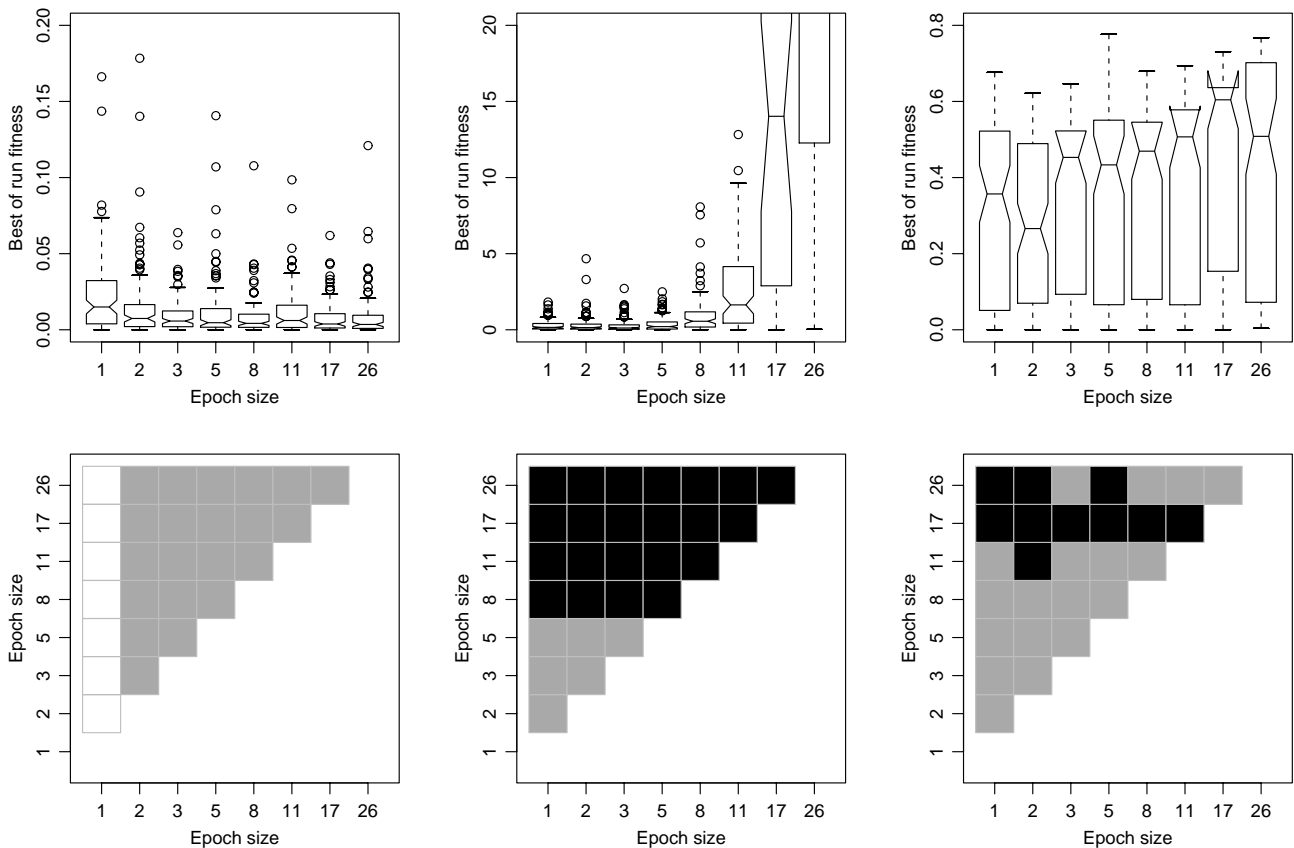


Figure 9: Top row: best of run statistics; minimization - smaller is better. Bottom row: statistical significance of differences between epoch sizes' performance (white: bigger epoch size performs better; black: smaller epoch size performs better; gray: undistinguishable performance). Columns: left - *rastrigin*, middle - *ofAxisQuadratic*, right - *rosenbrock*.

to those of BR_n^0 , namely they are perpendicular to one another and intersect at the optimum. For *offAxisQuadratic* the best-response curves also intersect at the optimum, but this time the angle between them is smaller, similar to that of $BR_n^{0.75}$. For *rosenbrock* the two best-response curves seem to overlap for some part. In fact they only do so in (1,1) (the optimum), but for $x \in [1, \sqrt{2.048}]$ the distance between $bestResponseY(x)$ and $bestResponseX^{-1}(x)$ is less than 10^{-3} . The curves become progressively closer for $x \in [0, 1]$. It looks like a mixture of $BR_n^{0.98}$ and BR_n^1 .

Based on these similarities, we expect the following effects of increasing the epoch size within a fixed budget:

- *rastrigin*: performance increases with the epoch size;
- *offAxisQuadratic*: undistinguishable performance for the first few epoch sizes and then decreasing performance;
- *rosenbrock*: similar performance for all epoch sizes, with just a few cases when higher epoch size performs worse.

We perform the same experiments as for BR_{10}^α , with the only exception that we adjust the sigma of the Gaussian mutation according to the size of these new domains (0.08 for *rosenbrock*, 0.2 for *rastrigin* and 2.6 for *offAxisQuadratic*, which is about 1/50 the size of the variables' range).

The experiments confirm our expectations, as can be seen from Figure 9. The top row shows boxplots of best of run fitness. For *rastrigin* we see some improvement in performance with increasing epoch size, for *offAxisQuadratic* some performance decay and for *rosenbrock* not much difference. The statistical significance plots on the bottom row support these findings and they look remarkably similar to the corresponding plots of BR_n^0 , $BR_n^{0.75}$, $BR_n^{0.98}$ and BR_n^1 .

5. CONCLUSIONS

To efficiently use cooperative coevolution as an optimization tool, one must understand the effect that the combination of problem properties and algorithm properties has on the system's performance. In this paper we contributed such understanding by analyzing the effects that the frequency of interaction between populations has on optimization performance. We showed that these effects are dependent on a problem property called best-response curves. We investigated the dynamics of the algorithm and explained the causes of this dependency.

For the dynamics analysis we used the technique previously employed in [10], [8], [7] and [9] and extended it by defining a quantitative metric for characterizing trajectories. The best-response problem property and the dynamics analysis of best individuals thus proved useful for understanding the behavior and optimization power of a wide range of variations of the basic coevolutionary algorithm [11].

Additionally, the best-response property helped identify certain type of problems that pose challenges to CoEAs, namely problems with overlapping best responses, such as BR_n^1 and *rosenbrock*. In the future, we would like to investigate whether for these problems performance could be improved by further increasing the interaction frequency (e.g., from every-generation to every-evaluation, by using a steady-state EA in each population).

As a further step, we plan to use the same techniques to investigate the *combined* performance effects of two or more parameters (e.g., interaction frequency, collaboration strategy, update timing). Our long-term goal is to provide coevolution practitioners with heuristics for matching the algorithm to the problem at hand.

6. REFERENCES

- [1] H. J. Blumenthal and G. B. Parker. Punctuated anytime learning for evolving multi-agent capture strategies. In *Congress on Evolutionary Computation*. IEEE Press, 2004.
- [2] J. Bongard and H. Lipson. Nonlinear system identification using coevolution of models and tests. *IEEE Transactions on Evolutionary Computation*, 9(4):361–384, 2005.
- [3] L. Bull. Evolutionary computing in multi-agent environments: Partners. In *Seventh International Conference on Genetic Algorithms*. Morgan Kaufmann, 1997.
- [4] T. Jansen and R. P. Wiegand. Sequential versus parallel cooperative coevolutionary (1+1) EAs. In *Congress on Evolutionary Computation*. IEEE Press, 2003.
- [5] L. Panait and S. Luke. Time-dependent collaboration schemes for cooperative coevolutionary algorithms. In *AAAI Fall Symposium on Coevolutionary and Coadaptive Systems*. AAAI Press, 2005.
- [6] E. Popovici and K. De Jong. Understanding competitive co-evolutionary dynamics via fitness landscapes. In *AAAI Fall Symposium on Artificial Multiagent Learning*. AAAI Press, 2004.
- [7] E. Popovici and K. De Jong. A dynamical systems analysis of collaboration methods in cooperative co-evolution. In *AAAI Fall Symposium on Coevolutionary and Coadaptive Systems*. AAAI Press, 2005.
- [8] E. Popovici and K. De Jong. Understanding cooperative co-evolutionary dynamics via simple fitness landscapes. In *Genetic and Evolutionary Computation Conference*. ACM Press, 2005.
- [9] E. Popovici and K. De Jong. The dynamics of the best individuals in co-evolution. *Journal of Natural Computing*, 2006. in print.
- [10] E. Popovici and K. De Jong. Sequential versus parallel cooperative coevolutionary algorithms for optimization. In *Proceedings of the Congress on Evolutionary Computation*. IEEE Press, 2006.
- [11] M. Potter. *The Design and Analysis of a Computational Model of Cooperative Coevolution*. PhD thesis, George Mason University, Computer Science Department, 1997.
- [12] M. Potter and K. De Jong. A cooperative coevolutionary approach to function optimization. In *Third Conference on Parallel Problem Solving from Nature*. Springer, 1994.
- [13] R. P. Wiegand. *An Analysis of Cooperative Coevolutionary Algorithms*. PhD thesis, George Mason University, Fairfax, VA, 2004.
- [14] R. P. Wiegand, W. Liles, and K. De Jong. An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In *Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, 2001. Errata available at <http://www.tesseract.org/paul/papers/gecco01-cca-errata.pdf>.
- [15] R. P. Wiegand, W. Liles, and K. De Jong. Analyzing cooperative coevolution with evolutionary game theory. In *Congress on Evolutionary Computation*. IEEE Press, 2002.