# ONLINE APPENDIX TO "SIMPLE FORECASTING HEURISTICS THAT MAKE US SMART: EVIDENCE FROM DIFFERENT MARKET EXPERIMENTS"

**Mikhail Anufriev**
University of Technology Sydney,
Business School

**Cars Hommes**
CeNDEF, University of Amsterdam
and Tinbergen Institute

**Tomasz Makarewicz**
University of Bamberg

E-mail: Mikhail.Anufriev@uts.edu.au (Anufriev); C.H.Hommes@uva.nl (Hommes);
tomasz.makarewicz@uni-bamberg.de (Makarewicz)

*Journal of the European Economic Association*
*Preprint prepared on 17 May 2018 using* jeea.cls *v1.0.*

# Appendices

## E On-Line Appendix. Formal definition of Genetic Algorithms

In this Appendix we present a formal definition of the Genetic Algorithms (GA) version, which served as the cornerstone of our model. It closely follows the standard specification suggested by Haupt and Haupt (2004) and used by Hommes and Lux (2013).

### E.1 Optimization procedures: traditional and Genetic Algorithms

Consider a maximization problem where the target function $\mathcal{F}$ of $N$ arguments $\theta = (\theta^1, \ldots, \theta^N)$ is such that a straightforward analytical solution is unavailable. Instead, one needs to use a numerical optimization procedure.

Traditional maximization algorithms, like the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, iterate a candidate argument for the optimum of the target function $\mathcal{F}$ by (1) estimating the curvature around the candidate and (2) using this curvature to find the optimal direction and length of the change to the candidate solution. This so called 'hill-climbing' algorithm is very efficient in its use of the shape of the target function. On the other hand, it will fail if the target function is 'ill-behaved': non-continuous or almost flat around the optima, has kinks or breaks. Here the curvature cannot be reliably estimated. Another problem of a computational nature is that the BFGS may perform poorly for a problem of large dimensionality.

The Genetic Algorithms are based on a fundamentally different

approach and therefore can be used for a wider class of problems. The basic idea is that we have a population of arguments which compete *only* in terms of their respective function value. This competition is modeled in an evolutionary fashion: mutation operators allow for a blind-search experimentation, but the probability that a particular candidate will survive over time is relative to its functional value. As a result, the target function may be as general as necessary, while the arguments can be of any kind, including real numbers, integers, probabilities or binary variables. The only constraint is that each argument must fall into a predefined dense interval (or set) $[a_n, b_n]$.

## E.2 Binary strings

A Genetic Algorithm (GA) uses $H$ chromosomes $g_{h,t} \in \mathbb{H}$ which are binary strings divided into $N$ genes $g^n_{h,t}$, each encoding one candidate parameter $\theta^n_{h,t}$ for the argument $\theta^n$. A chromosome $h \in \{1, \ldots, H\}$ at time $t \in \{1, \ldots, T\}$ has predetermined length $L$ and is specified as

$$g_{h,t} = (g^1_{h,t}, \ldots, g^N_{h,t}), \tag{E.1}$$

such that each gene $n \in \{1, \ldots, N\}$ has its length equal to an integer $L_n$ (with $\sum_{n=1}^{N} L_n = L$) and is a string of binary entries (bits)

$$g^n_{h,t} = (g^{n,1}_{h,t}, \ldots, g^{n,L_n}_{h,t}), \quad g^{n,l}_{h,t} \in \{0, 1\} \text{ for each } j \in \{1, \ldots, L_n\}. \tag{E.2}$$

The relation between the genes and the arguments is straightforward. An integer $\theta^n$ is simply encoded by (E.2) with its binary notation. Consider now an argument $\theta^n$ which is a probability. Notice that $\sum_{l=0}^{L_n-1} 2^j = 2^{L_n} - 1$. It follows that a particular gene $g^n_{h,t}$

can be decoded as a normalised sum

$$\theta_{h,t}^{n} = \sum_{l=1}^{L_n} \frac{g_{h,t}^{n,l} 2^{l-1}}{2^{L_n} - 1}. \tag{E.3}$$

A gene of zeros only is therefore associated with $\theta_n = 0$, a gene of ones only – with $\theta_n = 1$, while other possible binary strings cover the $[0, 1]$ interval with an $\frac{1}{2^{L_n}-1}$ increment. Any desired precision can be achieved with this representation. Since $2^{-10} \approx 10^{-3}$, the precision close to one over trillion ($10^{-12}$) is obtained by a mere of 40 bits.

A real variable $\theta^n$ from an $[a_n, b_n]$ interval can be encoded in a similar fashion, by an affine transformation of a probability:

$$\theta_{h,t}^{n} = a_n + (b_n - a_n) \sum_{l=1}^{L_n} \frac{g_{h,t}^{n,l} 2^{l-1}}{2^{L_n} - 1} \tag{E.4}$$

where the precision of this representation is given by $\frac{b_n - a_n}{2^{L_n}-1}$. Notice that one can approximate an unbounded real number by reasonably large $a_n$ or $b_n$, since the loss of precision is easily undone by a longer string.

## E.3 Evolutionary operators

The core of GA are its four operators. GA iterates the population of chromosomes for $T$ periods, where $T$ is either large and predefined, or depends on some convergence criterion. First, at each period $t \in \{1, \ldots, T\}$ each chromosome has its fitness equal to a monotone transformation of the function value $\mathcal{F}$. This transformation is defined as $V\left(\mathcal{F}(\theta_{h,t})\right) \equiv V(h_{k,t}) \to \mathbb{R}^+ \cap \{0\}$. For example, a non-negative function can be used directly as the fitness. If the problem is to minimize a function, a popular choice is the exponential transformation of the function values, similar to the one used in the logit specification

of the Heuristic Switching Model (Brock and Hommes, 1997).

Chromosomes at each period can undergo the following operators: reproduction, mutation, crossover and election. These operators first generate an offspring population of chromosomes from the parent population $t$ and therefore transform both populations into a new generation of chromosomes $t+1$ (notice the division of the process).

### E.3.1 Reproduction

For the population at time $t$, GA picks subset $\mathbb{X} \subseteq \mathbb{H}$ of $\chi$ chromosomes and picks $\kappa < \chi$ of them into a set $\mathbb{K}$. The probability that the chromosome $h \in \mathbb{X}$ will be picked into $\mathbb{K}$ as its $z$th element (where $z \in \{1, \ldots, \kappa\}$) is usually defined by the power function:

$$Prob(g_z = g_{h,t}) = \frac{V(g_{h,t})}{\sum_{j \in \mathbb{X}} V(g_{j,t})}. \tag{E.5}$$

This procedure is repeated with differently chosen $\mathbb{X}$'s until the number of chromosomes in all such sets $\mathbb{K}$'s is equal to $H$. For instance, the *roulette* is reproduction with $\chi = H$ and $\kappa = 1$: GA picks randomly one chromosome from the whole population, where each chromosome has probability of being picked equal to its function value relative to the function value of all other chromosomes. This is repeated exactly $H$ times.

So called *tournaments* are often used for the sake of computational efficiency. Here, $\chi << H$. For instance, GA could divide the chromosomes into pairs and sample two offspring from each pair.

Procreation is modeled as the basic natural selection mechanism. We consider subsets of the original population (or maybe the whole population at once). Out of each such a subset, we pick a small number of chromosomes, giving advantage to these which perform

better. We repeat this procedure until the offspring generation is as large as the old one. Thus the new generation is likely to be 'better' than the old one.

### E.3.2 Mutation

For each generation $t \in \{1, \ldots, T\}$, after the reproduction has taken place, each binary entry in each new chromosome has a predefined $\delta_m$ probability to mutate: ones turned into zeros and vice versa. In this way the chromosomes represent different numbers and may therefore attain better fit.

The mutation operator is where the binary representation becomes most useful. If the bits, which are close to the beginning of the gene, mutate, the new argument will be substantially different from the original one. On the other hand, small changes can be obtained by mutating bits from the end of the gene. Both changes are equally likely! In this way, GA can easily evaluate arguments which are both far away from and close to what the chromosomes are currently encoding. As a result, GA efficiently converges to the maximum, but are also likely *not* to get stuck on a local maximum. This is clearly independent of the initial conditions, which gives GA additional advantage over hill-climbing algorithms (like BFGS), where a good choice of the initial argument can be crucial to obtain the global maximum.

### E.3.3 Crossover

Let $0 \leqslant C_L, C_H \leqslant \sum_{n=1}^{N} L_n = L$ be two predefined integers. The crossover operator divides the population of chromosomes into pairs. If $C_L < L - C_H$, it exchanges the first $C_L$ and the last $C_H$ bits between chromosomes in each pair with a predefined probability $\delta_c$. Otherwise,

the crossover operator exchanges $max\{C_L, C_H\}$ bits in each pair of chromosomes with this predefined probability $\delta_c$. This operator facilitates experimentation in a different way than the mutation operator. Typically, it is set to exchange whole arguments, that is there are $0 \leqslant \nu_L \leqslant \nu_H \leqslant N$ such that $C_L = \sum_{n=1}^{\nu_L} L_n$ and $C_H = \sum_{n=\nu_H}^{N} L_n$. This allows the chromosomes to experiment with different compositions of the individual arguments, which on their own are already successful.

### E.3.4 Election

The experimentation done by the mutation and crossover operators does not need to lead to efficient binary sequences. For instance, a chromosome which actually decodes the optimal argument should not mutate at all. To counter this effect, it is customary to divide the creation of a new generation into two stages. First, the chromosomes procreate and undergo mutation and crossover in some predefined order. Next, the resulting set of chromosomes is compared in terms of fitness with the parent population. Thus, offspring will be passed to the new generation only if it *strictly* outperforms the parent chromosome. In this way each generation will be at least as good as the previous one, what in many cases facilitates convergence.

# F On-Line Appendix. Parametrization of the forecasting heuristic

In this Appendix, we will address two issues. First, we will investigate the importance of the anchor in the forecasting heuristic both for the one-period ahead **HHST09** and for the two-period ahead **HSTV05** settings. Second, we study the allowed degree of trend extrapolation (i.e., the interval for the trend coefficient $\beta$), based on the linear feedback from **HHST09**.

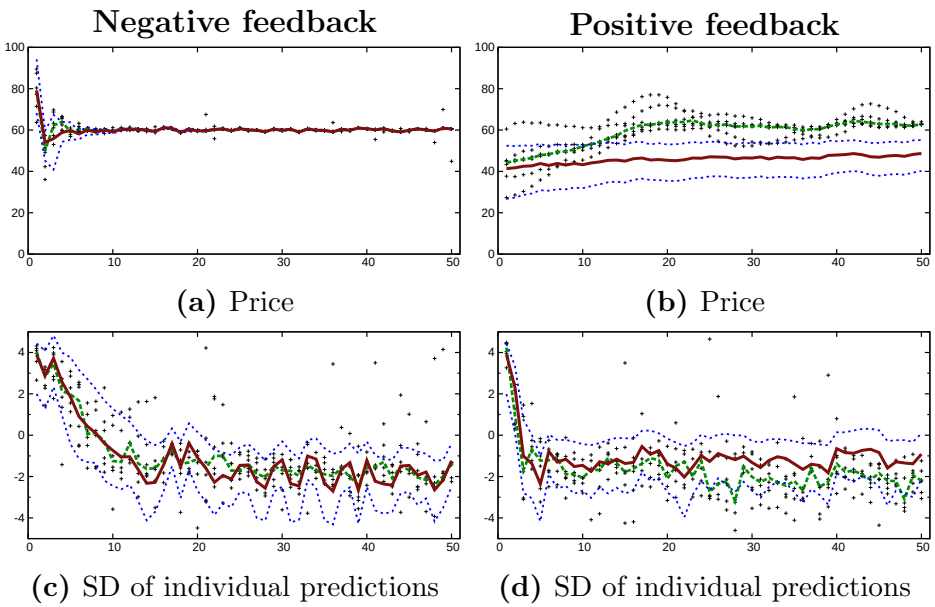
## F.1 Is the anchor important for HHST09?

**HHST09** show that most of their subjects (around 60%) use the First-Order prediction rule with heterogeneous parameter specification:

$$p^e_{i,t} = \alpha_1 p_{t-1} + \alpha_2 p^e_{i,t-1} + \alpha_3 60 + \beta(p_{t-1} - p_{t-2}) \tag{F.1}$$

where the anchor is a weighted average of the last observed price, the last forecast and the fundamental price $p^f = 60$, $\alpha$'s span a simplex ($\alpha_1 + \alpha_2 + \alpha_3 = 1$) and $\beta$ is the trend extrapolation coefficient. Our FOR heuristic (10) is a special case of (F.1) with the restriction that $\alpha_3 = 0$, which implies that the fundamental price is not used by the agents (recall footnote 16).

Experimental literature suggests that, in general, anchors and focal points are important in describing human behavior. However, **HHST09** report that the anchor coefficient $\alpha_3$ is typically significant for the subjects under negative feedback, while most of the subjects under positive feedback do not use it (only 2 out of 42 subjects have a significantly positive $\alpha_3$). Furthermore, under negative feedback




**(a)** Price **(b)** Price

**(c)** SD of individual predictions **(d)** SD of individual predictions

**Figure F.1: HHST09**: 50-period ahead Monte Carlo simulation (1000 runs) for the **GA-P1** model with the FOR (F.2) compared with the experimental data. *Upper panels:* price. *Lower panels:* degree of coordination ($\log_2$ scale). Green dashed line and black pluses show the experimental median and group observations, respectively. Red thick line is the median and blue dotted lines are the 95% confidence interval for the GA model.

prices and forecasts quickly converge to the vicinity of 60, which in practice makes the three $\alpha_k$ coefficients unidentifiable; and could make the anchor itself redundant. When designing our GA model, we therefore investigated whether the fundamental anchor has any additional explanatory power.

To simplify econometric issues, in the previous literature the anchor was set at the fundamental level, which however was not directly given to the subjects. It is more plausible that the subjects used the average of all previous prices as an anchor. We will use a GA with the FOR specified specified as

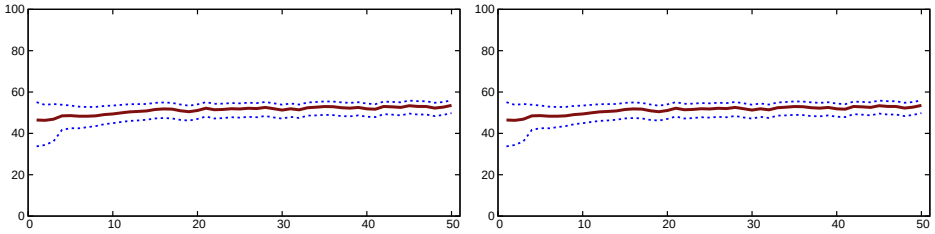
$$p^e_{i,t} = \alpha_1 p_{t-1} + \alpha_2 p^e_{i,t-1} + (1-\alpha_1-\alpha_2)\left(\frac{1}{t-1}\sum_{s=1}^{t-1} p_s\right) + \beta(p_{t-1}-p_{t-2}). \tag{F.2}$$

This extended GA model has 3 parameters (two weights within the anchor and the trend extrapolation coefficient), instead of 2 parameters (price weight and trend extrapolation coefficient) from our GA model, which is based on heuristic (10).

We run the Monte Carlo (MC) simulations exactly as in the first part of Section 5.1, but for the GA model based on (F.2) with the restriction for $\beta \in [-1.1, 1.1]$. The results are presented in Fig. F.1. We observe for the positive feedback that, in contrast to our restricted GA model without the fundamental anchor, the GA model based on FOR (F.2) does not predict oscillations at all. Instead a sluggish convergence towards the fundamental is generated, as can be seen in the stable median price, bounded by relatively narrow 95% CI. In other words, this specification misses most of the dynamics observed in half of the experimental groups. We conclude that there is no evidence for a need of a fundamental anchor, specified as a long-run average of the observed prices, in our GA model.

## F.2 Anchor and HSTV05

The **HSTV05** non-linear, two-period ahead LtF asset pricing market resulted in more pronounced oscillations than those observed in the simple linear experiment **HHST09** under positive feedback. One could therefore think that some kind of a long-run anchor might have been important for the subjects, even though they would not use it in one-period ahead forecasting setting. Furthermore, in **HSTV05** the oscillations typically arose around the fundamental price, which again suggests that the subjects tried to anchor the price changes to it. To address this issue, we run the 50-period ahead MC simulation as in Section 5.3, but where the heuristic (21) is replaced by the extended




**(a) GA-P1**, $\beta \in [-1.1, 1.1]$ **(b) GA-P3**, $\beta \in [-1.3, 1.3]$

**Figure F.2: HSTV05** with $p^f = 60$: 50-period ahead Monte Carlo simulation (1000 runs) for the **GA-P1** (*left panel*) and **GA-P3** (*right panel*) models with FOR (F.2). Price evolution is shown. Red line is the median and blue dotted lines are the 95% CI.

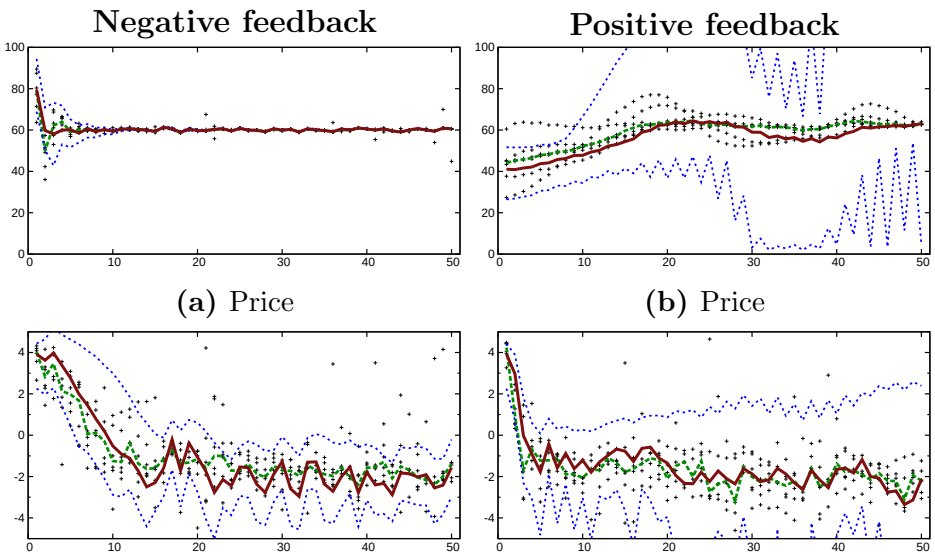
FOR heuristic (F.2) adapted for the two-period ahead setting, and where the anchor was given by the fundamental price $p^f = 60$.

Results for two parametrizations (with allowed trend extrapolation $\beta \in [-1.1, 1.1]$ and $\beta \in [-1.3, 1.3]$) are presented on Fig. F.2. As in the case of **HHST09**, we find that the GA model with the extended FOR rule generates sluggish convergence towards the fundamental price from below. Indeed, in contrast to **HHST09**, the 95% CI of the GA model's prices do not include the fundamental $p^f = 60$ even after 50 periods. This indicated that adding an anchor to the GA model would decrease its fitness to the experimental data.[1]

## F.3 Degree of trend extrapolation in HHST09

Recall that the GA requires a predefined finite interval for the optimized parameters. In the case of our GA model based on (10), the price weight is confined to $\alpha \in [0, 1]$, but *prima facie* there is no 'natural' bound for the trend extrapolation $\beta \in [\beta_L, \beta_H]$, since *a priori* we do not know the degree of trend extrapolation that people consider while forecasting prices. As mentioned in Section 4.2, we argue that

[1] We found similar results when the anchor was specified as the sample average price $\frac{1}{t-1}\sum_{s=1}^{t-1} p_s$.




**(a)** Price **(b)** Price

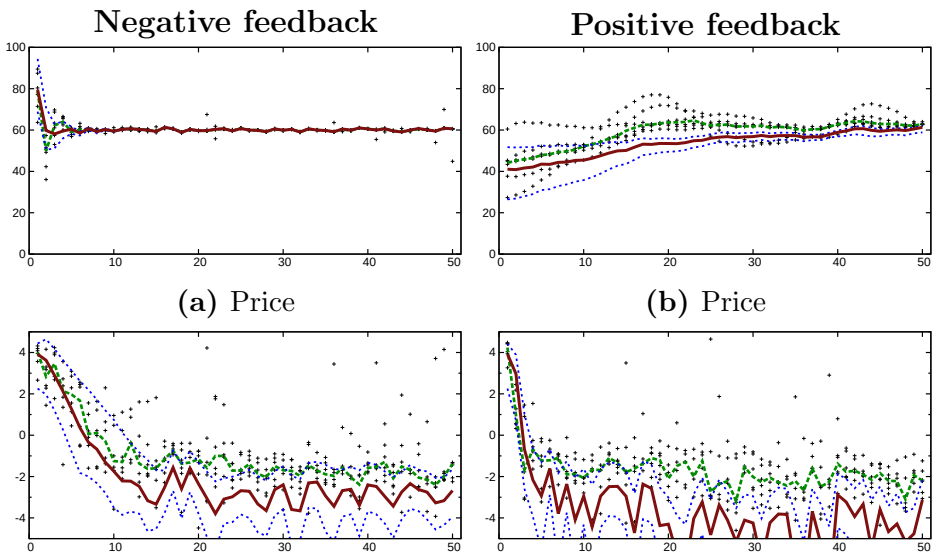
**(c)** SD of individual predictions **(d)** SD of individual predictions

**Figure F.3: HHST09**: 50-period ahead Monte Carlo simulation (1000 runs) for the model with restriction $\beta \in [-1.5, 1.5]$ compared with the experimental data. *Upper panels:* price. *Lower panels:* degree of coordination. Green dashed line and black pluses show the experimental median and group observations, respectively. Red thick line is the median and blue dotted lines are the 95% confidence interval for the GA model.

the model performs well in the **HHST09** economy if we specify the heuristic (10) with an upper bound of 1.1 to the trend coefficient $\beta$ (as in **GA-P1** and **GA-P2**).

It turns out (unsurprisingly) that the allowed trend extrapolation interval has little effect on the behavior of our GA model under negative feedback. However, a clear effect exists for the model under positive feedback: the larger the interval $\beta \in [\beta_L, \beta_H]$, the larger the amplitude of the price oscillations. We experimented with different bounds, trying to calibrate the GA model to the experimental oscillations, using the same Monte Carlo experiments as in Section 5.1.1.

Allowing for a strong trend extrapolation $\beta \in [-1.5, 1.5]$ results in a model with huge possible oscillations and little predictive power, see Fig. F.3. On the other hand, parametrization with $\beta \in [-0.5, 0.5]$ has




**(a)** Price **(b)** Price

**(c)** SD of individual predictions **(d)** SD of individual predictions

**Figure F.4: HHST09**: 50-period ahead Monte Carlo simulation (1000 runs) for the model with restriction $\beta \in [-0.5, 0.5]$) compared with the experimental data. *Upper panels:* price. *Lower panels:* degree of coordination. Green dashed line and black pluses show the experimental median and group observations, respectively. Red thick line is the median and blue dotted lines are the 95% confidence interval for the GA model.

narrow CI, but predicts small oscillations, see Fig. F.4. We found that the model with $\beta \in [-1.1, 1.1]$ is the best trade-off between in-sample fit and out-sample predictive power of the model.