CONTROLLING INVESTMENT PROPORTION IN CYCLIC CHANGING ENVIRONMENTS

J.-Emeterio Navarro-Barrientos
Department of Computer Science, Humboldt-Universität zu Berlin, Unter den Linden 6 10099, Berlin, Germany
navarro@informatik.hu-berlin.de

Keywords: Genetic algorithms, investment strategies, pattern recognition.

Abstract: In this paper, we present an investment strategy to control investment proportions for environments with cyclic changing returns on investment. For this, we consider an investment model where the agent decides at every time step the proportion of wealth to invest in a risky asset, keeping the rest of the budget in a risk-free asset. Every investment is evaluated in the market modeled by stylized returns on investment (RoI). For comparison reasons, we present two reference strategies which represent agents with zero-knowledge and complete-knowledge of the dynamics of the RoI, and we consider an investment strategy based on technical analysis. To account for the performance of the strategies, we perform some computer experiments to calculate the average budget that can be obtained over a certain number of time steps. To assure for fair comparisons, we first tune the parameters of each strategy. Afterwards, we compare their performance for RoIs with fixed periodicity (stationary scenario) and for RoIs with changing periodicities (non-stationary scenario).

1 INTRODUCTION

Finding a proper method to control investment proportion is a problem that has been addressed by many researchers (Kelly, 1956; Kahneman and Tversky, 1979). Many of the proposed methods are based on *machine learning (ML)*. For example, in (Magdon-Ismail et al., 2001) the authors use neural networks to find patterns in financial time series and in (Geibel and Wysotzki, 2005), the authors propose a risk-sensitive reinforcement learning algorithm to find a policy for controlling under constraints. Other techniques from ML that are also frequently used are those based on *evolutionary computation*, like *genetic programming (GP)* and *genetic algorithms (GA)*. For a general introduction to these techniques for portfolio management and bankruptcy prediction see (Dawid, 1999). Some researchers have shown that investment strategies based on GP techniques may be profitable; however, they usually find strategies which can't be easily funded (Schulenburg and Ross, 2001). Controlling strategies that are based on a standard GA may be also difficult to explain, however, we believe that they are easier to understand than those using GP.

Genetic algorithms (GA) are stochastic search algorithms based on evolution that explore progressively from a large number of possible solutions finding after some generations the best solution for the problem. Inspired by natural selection, these powerful techniques are based on some defined evolution operators, like selection, crossover and mutation (Holland, 1975). Moreover, some researches have extended the use of GA for solving stochastic dynamic optimization problems online (Grefenstette, 1992), where most of the algorithms for changing environments are tested in problems like the knapsack problem (Yang, 2005). However, to our knowledge, no-one has applied GAs specifically to the problem of controlling the proportion of investment in environments with cyclic changing returns on investment.

This paper is organized as follows: Sec. 2 describes the investment model and Sec. 3 presents a novel approach to control investment proportions based on a GA for environments with cyclic changing time series. In Sec. 4 we present the dynamics for the risky asset and we compare the performance of the adaptive strategy with other strategies for stationary and non-stationary environments.

2 INVESTMENT MODEL

We consider an investment model (Navarro and Schweitzer, 2003) where an agent is characterized by two individual variables: (i) its *budget* $x(t)$, and (ii) its *investment proportion* $q(t)$. The budget, $x(t)$, changes in the course of time t by means of the following dynamic:

$$x(t+1) = x(t)\left[1 + r(t)\,q(t)\right] \tag{1}$$

This means that the agent at time t invests a portion $q(t)x(t)$ of its total budget. And this investment yields a gain or loss on the market, expressed by $r(t)$, the return on investment, *RoI*. Some authors assume that returns are obtained by means of continuous double auction mechanisms (LeBaron, 2001), however, in this paper we consider that the returns are not being influenced by agent's actions, this approach plays a role in more physics-inspired investment models, (Richmond, 2001; Navarro-Barrientos et al., 2008). Since $q(t)$ always represents a portion of the total budget $x(t)$, and it is bound to $q(t) \in [0,1]$. For completeness, we assume that the minimal and maximal investment proportions are described by $q_{\min}$ and $q_{\max}$, respectively.

Thus, in this paper we present an adaptive strategy to control proportions of investment, expressed by a method to find the most proper $q(t)$. We assume a simple dynamic for the returns allowing us to focus in the feedback of these market returns on the investment strategy (and not on the feedback of the strategies on the market). Moreover, we assume that the agent invests independently in the market, i.e. there is no direct interaction with other agents.

3 ADAPTIVE INVESTMENT STRATEGY

In this section, we present an adaptive investment strategy based on a GA for controlling proportions of investment in cyclic changing environments. For simplicity, we call this strategy *Genetic Algorithm for Changing Environments* (GACE), and we show on the following the specifications for the GA.

3.1 Encoding Scheme

We consider a population of chromosomes $j = 1,...,C$, where each chromosome j has an array of genes, g_{jk}, where $k = 0,...,G_j - 1$, and G_j is the length of the chromosome j. The length of a chromosome is assumed to be in the range $G_j \in (1, G_{\max})$, where $G_{\max}$ is a parameter that specifies the maximal allowed number of genes in a chromosome. The values of the genes could be binary, but for programming reasons we use real values (Michalewiçz, 1999). Moreover, each chromosome j represents a *set of possible strategies* of an agent, where each g_{jk} is an investment proportion.

3.2 Fitness Evaluation

Each chromosome j is evaluated after a given number of time steps by a *fitness function* $f_j(\tau)$ defined as follows:

$$f_j(\tau) = \sum_{k=0}^{G_j-1} r(t)\,g_{jk}\,; \quad k \equiv t \bmod G_j, \tag{2}$$

where τ is a further time scale in terms of generations. When a generation is completed, the chromosomes' population is replaced by a new population of better fitting chromosomes with the same population size C. Since the fitness of a chromosome tends to be maximized, negative $r(t)$ should lead to small values of g_{jk}, and positive $r(t)$ should lead to larger values of g_{jk}. Because of this, we consider the product of $r(t)g_{jk}$ as a performance measure, which is in accordance with our investment model, Eq. (1). Noteworthy, in this approach the GA tries to find the chromosomes leading to larger profits. A different approach would be to implement a GA to find the chromosomes that minimize the loss, in which case, we would have a different fitness function. Note that we treat directly returns on investment and not price movements, because our goal is to evaluate the fitness of the strategies for the RoI and not the accuracy of the prediction of the next RoI.

3.3 Selection of a New Population

If we assume that chromosomes have fixed length, $G_j = G_{\max}$, then the most proper number of time steps that have to elapse in order to evaluate all chromosomes' genes is $t_{\text{eval}} = G_{\max}$. However, this previous assumption corresponds to the ideal case where the agent knows the periodicity of the returns and sets the length of all chromosomes to this value. In this paper, we assume that the agent **doesn't know** neither the **periodicity** nor the **dynamics** of the RoI. Thus, we assume that the chromosomes have different length. Different approaches may be proposed to know after how many time steps a new generation of chromosomes should be obtained, however, we find that the best approach was to choose the number of time steps for evaluation accordingly to the length of the best chromosome in the population.

3.3.1 Elitist and Tournament Selection

After calculating the fitness of each chromosome according to Eq. (2), we first find the best chromosomes from the current population by applying elitist selection, which copies directly the best s percentage to the new population. Afterwards, a tournament selection of size two is done by randomly choosing two pairs of chromosomes from the current population and then selecting from each pair the one with the higher fitness. These two chromosomes are not simply transferred to the new population, but undergo a transformation based on the genetic operators' crossover and mutation.

3.3.2 Crossover and Mutation Operators

The limitations of conventional crossover in GA with variable length has already been addressed by some authors, where neural networks or hierarchical tree-structures are used to determine which genes should be exchanged between the chromosomes (Harvey, 1992). For simplicity, we propose a modification of the standard GA crossover operator that better suits our demands. Thus, we propose the use of a crossover operator called *Proportional Exchange Crossover* (PEC) operator, which randomly selects the range of genetic information to be exchanged between two chromosomes and contracts(extends) the genetic information from the largest(shortest) to the shortest(largest) chromosome, respectively. Algorithm 1 shows the PEC algorithm for all pair of parent-chromosomes being selected via tournament selection. Note that a chromosome j is saved in an array with indexes in the range $[0,..,G_j - 1]$. The shortest and largest parent-chromosomes are denoted by pa_s with size G_s and pa_l with size G_l, resp., and $R \in \mathbb{N}$ is the size proportion between these two parent-chromosomes. The cross-points for the shortest and largest parent-chromosomes are denoted by $cp_s \in \mathbb{N}$ and $cp_l \in \mathbb{N}$, respectively. The breeding between the two parent-chromosomes results in a short and a large children-chromosomes denoted by ch_s and ch_l, respectively.

Note, that different functions could be considered for the transformation of the genetic material between chromosomes with different length. For simplicity, we consider in our implementation of the Algorithm 1 in line 9 the function $extend(pa,m,R) = pa[m]$, which simple copies the genes from the short parent-chromosome to the large child-chromosome; and in line 13 the function $contract(pa,m,R) = 1/R \sum_{i=m}^{m+R} pa[i]$, which performs an average over the genetic material.

Now, to make sure that a population with chromosomes of diverse lengths is present, we introduce a mutation operator for the length of the chromosome. For this, with probability p_l a new length is drawn randomly and the genetic information of the chromosome is proportionally scaled to the new length, leading to a new enlarged or stretched chromosome. The algorithm used for the mutation of the length of the chromosome is based on the same principle as the PEC operator. After the crossover and length-mutation operators are applied, the typical gene-mutation operator is applied. This means that with a given mutation probability $p_m \in U(0,1)$, a gene is to be mutated by replacing its value by a random number from a uniform distribution $U(q_{\min}, q_{\max})$.

Algorithm 1: Proportional Exchange Crossover (PEC) operator.

1 **foreach** *pair of parent-chromosomes* **do**
2   create ch_s and ch_l with sizes G_s and G_l
3   find the cross-points:
  $cp_s \sim U(0, G_s - 1)$; $cp_l = \frac{G_l\, cp_s}{G_s}$
4   determine size proportion: $R = cp_l / cp_s$
5   with $p = 0.5$ choose side for crossover
6   **if** *crossover on the left side* **then**
7   extend genes from pa_s to ch_l:
    for $m = 0$ *to* $cp_s - 1$ **do**
8   **for** $n = 0$ *to* $R - 1$ **do**
9         $ch_l[m \cdot R + n] \leftarrow$ extend(pa_s, m, R)
10   **end**
11   **end**
12   contract genes from pa_l to ch_s:
    foreach $m = 0$ *to* $cp_s - 1$ **do**
13       $ch_s[m] \leftarrow$ contract(pa_l, m, R)
14   **end**
15   **else**
16   extend as in 9 but for $m = cp_s$ to $G_s - 1$
17   contract as in 13 but for $m = cp_s$ to $G_s - 1$
18   **end**
19   copy remaining genes in pa_s and pa_l into same positions in ch_s and ch_l, respectively.
20 **end**

3.3.3 Strategy Selection and Initialization

For every new generation, the agent takes the set of strategies g_{jk} from the chromosome j with the largest fitness in the previous generation.

$$q(t) = g_{lk} \text{ with } l = \arg\max_{j=1,..,C} f_j;\ k \equiv t \bmod G_l \quad (3)$$

For the initialization, each g_{jk} is assigned a random value drawn from a Uniform distribution: $g_{jk} \sim U(q_{\min}, q_{\max})$. And for the length of the chromosomes, each G_j is initialized randomly from a Uniform distribution of integers: $G_j \sim (1, G_{\max})$, where $G_{\max}$ is the maximal allowed chromosome length.

4 EXPERIMENTAL RESULTS

In this section, we present the environment for the agent and we analyze the performance of the adaptive strategy presented above.

4.1 Artificial Returns

We consider artificially generated returns which are driven by the following dynamics:

$$r(t) = (1-\sigma)\sin\left(\frac{2\pi}{T}t\right) + \sigma\xi, \qquad (4)$$

where the amplitude of the returns depends on the amplitude noise level $\sigma \in (0,1)$, and ξ corresponds to a random number drawn from a Uniform distribution, $\xi \in U(-1,1)$. The periodicity of the returns is drawn randomly $T \sim U(1, T_{max})$ and would be present for a number of $t' \sim U(1, t_{max})$ time steps. Thus, σ accounts for the fluctuations in the market dynamics on the amplitude of the RoI; T_{max} accounts for the largest possible periodicity and t_{max} accounts for the maximal number of time steps a periodicity can elapse. Fig

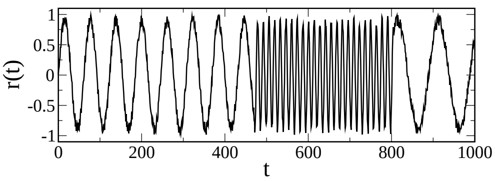

Figure 1: Periodic RoI, $r(t)$, Eq. (4) for noise level $\sigma = 0.1$, $T_{max} = 100$ and $t_{max} = 1000$.

4.2 Reference Strategies

For comparison purposes, we present in this section different strategies which are used as a reference to account for the performance of the adaptive strategy. Note that we could have considered other type of strategies which may lead to a more complete study. However, our main goal is to show the performance of GACE comparing it against the performance of other strategies for the same investment scenario.

4.2.1 Strategies with Zero/Complete Knowledge

For comparison reasons, we present in this section two strategies representing two simple behaviors for an agent: the first one, called *Constant-Investment-Proportion* (CP), represents the agent with zero knowledge and zero-intelligence; the second one, called *Square-Wave* (SW), represents the agent with complete knowledge of the environment.

Constant Investment Proportion (CP). The simplest strategy for an agent would be to take a constant investment proportion for every time step:

$$q(t) = q_{min} = \text{const.} \qquad (5)$$

Square Wave Strategy (SW). An agent using this strategy invests q_{max} during the positive cycle of the periodic return and invests q_{min} otherwise:

$$q(t) = \begin{cases} q_{max} & t \bmod T < T/2 \\ q_{min} & \text{otherwise}. \end{cases} \qquad (6)$$

Notice that this reference strategy assumes that the agent **knows** in advance the **periodicity**, T, and the **dynamics** of the returns.

4.2.2 Strategy based on Technical Analysis

We include in our study a strategy based on *technical analysis* methods, which are frequently used by traders to forecast returns. For simplicity, we chose the *Moving Least Squares* (MLS) technique and considered an agent with a memory size M to store previous received returns. This strategy fits a linear trendline to the previous M returns, to estimate the next return, $\hat{r}(t)$. Noteworthy, once the next return has been estimated, the agent still needs to perform the corresponding adjustment of the investment proportion. For this, we consider that the agent has a *risk-neutral* behavior as follows:

$$q(t) = \begin{cases} q_{min} & \hat{r}(t) \le q_{min} \\ \hat{r}(t) & q_{min} < \hat{r}(t) < q_{max} \\ q_{max} & \hat{r}(t) \ge q_{max} \end{cases} \qquad (7)$$

4.3 Results for RoI with Fixed Periodicity

To elucidate the performance of the adaptive strategy proposed in this paper and the reference strategies previously presented, we start with a simple scenario where returns have a fixed periodicity.

First, we assume that the parameters of a strategy lead to an optimal performance, if it leads to the *maximum total budget* that can be reached with this strategy during a complete period of the returns. When evaluating the strategies, we have to consider that their performance is also influenced by stochastic effects. In the case of the strategy GACE we also have to account for the different possible strategies that may evolve. This means that we have to average the simulation over a large number of trials, N, where each trial simulates an agent acting independently with the same set of parameter values. For convenience, we reinitialize the budget after each cycle of the RoI. This is done, because if the strategy performs well, the budget of the agent may reach very high values, which would lead to numerical overflows.

4.3.1 GACE Parameter Tuning

The configuration of most meta-heuristic algorithms requires both complex experimental designs and high computational efforts. Thus, for finding the best parameters for the GA, a software called *+CARPS (Multiagent System for Configuring Algorithms in Real Problem Solving)* (Monett, 2004) was used. It consists of autonomous, distributed, cooperative agents that search for solutions to a configuration problem, thereby fine-tuning the meta-heuristic's parameters.

The *GA* was configured for periodic returns with $T = 100$ and different level of noise: $\sigma = 0.1$, and $\sigma = 0.5$. Four GA parameters were optimized: the population size C, the crossover probability p_c, the mutation probability p_m, and the elitism size s. Their intervals of definition were set as follows: $C \in \{50, 100, 200, 500, 1000\}$, $p_c \in [0.0, 1.0]$, $p_m \in [0.0, 1.0]$, and $s \in [0.0, 0.5]$. We show in Table 1 the best obtained configuration for the GA in the periodic returns previously mentioned. For clarity, we considered in these experiments chromosomes with fixed length, i.e. $G_j = T$, and no probability of length mutation, i.e $p_l = 0$.

Table 1: GACE's best pars. for RoI with fixed T.

C	p_c	p_m	s
1000	0.7	0.01	0.3

Now, to better illustrate the set of investment strategies that are being obtained using GACE, we show in Fig. 2 the RoI and the investment proportions obtained after a number of time steps. For the reader with background in signal processing techniques, Fig. 2 may sound familiar as it resembles to those figures obtained when using matched filters for signal recovery (Turing, 1960).

4.3.2 Performance Comparison

In order to assure fair comparison between the strategies, we need to find the most proper parameter values for the strategies. Note that for both strategies CP, Eq. (5) and SW, Eq. (6), we don't need to tune any parameters. However, for the strategy MLS, Eq. (7), we assume that the agent **knows** the **periodicity** T of the returns. This means that the agent needs to determine the most proper memory size, M, based on the known periodicity of the returns. For this, we performed some computer experiments using MLS with

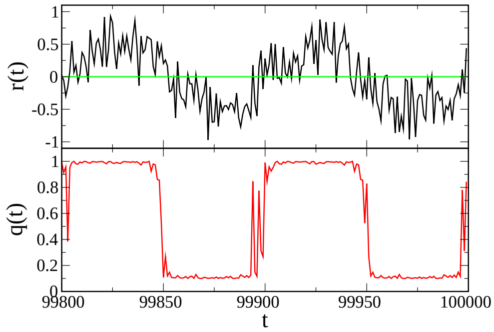

Figure 2: (Top)Return $r(t)$ and (bottom) investment control strategy $q(t)$ using GACE after $t = 10^5$ time steps, for returns with $T = 100$ and $\sigma = 0.5$.

different memory sizes for returns with different fixed periodicities, T, and no noise, finding that the most proper memory size, M, and the periodicity, T, are proportionally related by $M/T \approx 0.37$. Now, if we assume returns with no noise, we can find analytically the memory size $M^\star$ that maximizes the profits. For this, it can be shown that for periodic returns as in Eq. (4) with $\sigma = 0$, the strategy MLS, Eq. (7), estimates the next return $\hat{r}(t+1)$ as follows:

$$\hat{r}(t+1) = \frac{M+1}{M}\left[\sin(\omega t) - \sin(\omega t - \omega M)\right], \quad (8)$$

where $\omega = 2\pi/T$. Now, by calculating the average profits $\langle r q \rangle$ for the positive cycle of the returns:

$$\langle r q \rangle = \frac{T\,(M + 1 - \cos(\omega M))}{4M}. \quad (9)$$

And obtaining the derivative of $\langle r q \rangle$ w.r.t M:

$$\partial_M \langle r(t)\, q(t) \rangle = \frac{-T \sin\left(\frac{\omega}{2} M\right)^2 + \pi M \sin(\omega M)}{2M^2}. \quad (10)$$

It can be shown that by solving $\partial_M \langle r(t)\, q(t) \rangle = 0$ and using Taylor expansion to the sixth order for the sinusoidal functions, the memory size $M^\star$ that maximizes the profits corresponds to:

$$M^\star = \frac{\sqrt{\frac{3}{2}}}{\pi} T. \quad (11)$$

Consequently, the proportion $M/T \approx 0.37$ found by means of computer simulations approximates pretty well the proportion found analytically $M/T = \sqrt{\frac{3}{2}}/\pi = 0.389$.

Now, we compare the performance of the adaptive investment strategy GACE, presented in Sec. 3, with respect to the reference strategies presented in Sec. 4.2. For clarity, we assume for the moment that the strategy GACE uses fixed chromosome length, i.e. $G_j = G_{\max}$. For all strategies we consider $q_{\min} = 0.1$

and $q_{max} = 1.0$ in our experiments. These parameter values describe the behavior of the strategies CP, Eq. (5), and SW, Eq. (6). For the strategy MLS, Eq. (7), we use Eq. (11) to determine the optimal memory size and for the strategy GACE we use the parameters in Table 1.

In our experiments we assume that the agent invests in returns with periodicity $T = 100$ for different noise levels. We consider here that the length of the chromosomes is fixed to $G_j = 100$ and a new generation of chromosomes is being obtained after a number of time steps $t_{eval} = 100$. For the computer experiments, we let the agent to use one of the strategies to invest during a number of $t = 10^5$ time steps. In order to account for the randomness of the scenario, we perform the experiment for a number of $N = 100$ trials, gathering the average budget obtained for each strategy at every 100 time steps.

Fig. 3 shows in a log-log plot the average budget, $\langle x \rangle$, for all strategies in the course of GACE's generations, τ. Except for the GACE strategy, all other strategies have a constant budget in average over each generation. This occurs because the average of the budget and the time steps to evaluate the population of chromosomes were taken at every 100 time steps, which corresponds to the periodicity of the returns $T = 100$. Noteworthy, after 4 and 300 generations GACE over-performs the strategies CP and MLS and after 400 generations it performs almost as well as the strategy SW.

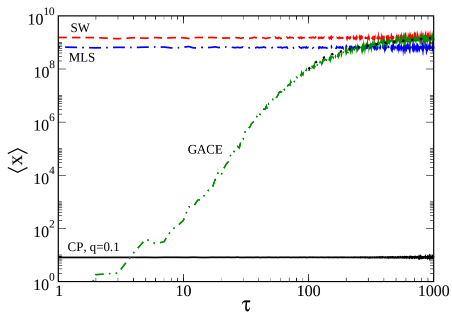

Figure 3: Average budget, $\langle x \rangle$, for different investment strategies in the course of generations τ, for returns with periodicity $T = 100$ and noise $\sigma = 0.1$.

4.4 Results for RoI with Changing Periodicity

In the previous section, we showed results for a stationary environment, now in this section we tackle a non-stationary environment.

4.4.1 GA Parameter Tuning

We used again the program +CARPS to find the best parameters for GACE now for returns with changing periodicity. The *GA* was configured for returns with a maximal periodicity of $T_{max} = 100$ and maximal elapsing time steps $t_{max} = 10^4$ for different level of noise: $\sigma = 0.1$, and $\sigma = 0.5$. In this process, we used the same intervals of definition as in Sec. 4.3.1, with the inclusion of the interval: $p_l \in [0.0, 1.0]$. The resulting best parameter values are shown in Table 2.

Table 2: GACE's best pars. for RoI with changing T.

C	p_c	p_m	s	p_l
1000	0.5	0.001	0.3	0.5

4.4.2 Performance Comparison

In this section we investigate the performance of the adaptive strategy with respect to the reference strategies in a non-stationary scenario. For this, we performed some computer experiments for returns with changing periodicity. As we did in the previous sections, we assumed for all strategies the parameter values $q_{min} = 0.1$ and $q_{max} = 1.0$ and for the strategy MLS we used Eq. (11) to calculate the memory size, M. For the strategy GACE, we used the parameter values listed in Table 2 and the length of a chromosome in the range $G_j \in (1, G_{max})$, with $G_{max} = 200$.

We show in Fig. 4 (top) the evolution of budget for different investment strategies, and (bottom) the corresponding periodicity of the returns, Eq.(4), both in the course of time. Thus, the best strategy is the strategy SW, following the strategy MLS; however, note that both strategies have total and partial knowledge about the dynamics of the returns, respectively. This previous knowledge gives some advantage to these strategies over the strategy GACE, which only needs the specification of G_{max}. We note that the strategy GACE evolves quite fast, yielding a set of investment strategies with a clear tendency to lead more gains than losses.

5 CONCLUSIONS

In this paper, we presented a simple investment model and some investment strategies to control the proportion of investment in cyclic changing environments. The novelty of this paper is in the adaptive investment strategy here proposed, called *Genetic Algorithm for Changing Environments* (GACE), which is a new approach based on evolution for the correct mapping of

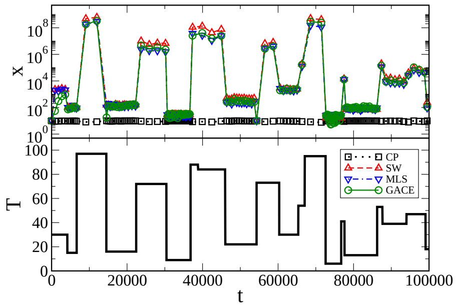

Figure 4: (Top) Budget for different strategies and (bottom) Periodicity of the returns, both in the course of time for RoI with parameters $T_{\max} = 100$, $t_{\max} = 10^4$ and $\sigma = 0.1$.

investment proportions to patterns that may be present in the returns. We analyzed the performance of GACE for different scenarios, and compared its performance in the course of time against other strategies used here as a reference. We showed that even though the strategy GACE has no knowledge of the dynamics of the returns, after a given number of time steps it may lead to large gains, performing as well as other strategies with some knowledge. This particularly is shown for long-lasting periodicities, where an ever increasing growth of budget was observed. Further work includes the analysis of the performance of the strategy GACE for real returns, and to compare the performance of GACE against other approaches from Machine Learning.

ACKNOWLEDGEMENTS

We thank to Dr. Dagmar Monett for providing us the program *+CARPS*. We also thank Frank Schweitzer and H.-D. Burkhard for helpful advice.

REFERENCES

Dawid, H. (1999). *Adaptive learning by genetic algorithms: Analytical results and applications to economic models*. Springer, Berlin.

Geibel, P. and Wysotzki, F. (2005). Risk-sensitive reinforcement learning applied to control under constraints. *J. of Artif. Intell. Res.*, 24:81–108.

Grefenstette, J. J. (1992). Genetic algorithms for changing environments. In Manner, R. and Manderick, B., editors, *Parallel Problem Solving from Nature 2.*, pages 137–144. Elsevier, Amsterdam.

Harvey, I. (1992). The sage cross: The mechanics of recombination for species with variable-length genotypes. In *Parallel Problem Solving from Nature*, volume 2, pages 269–278. Elsevier, Amsterdam.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press.

Kahneman, D. and Tversky, A. (1979). Prospect theory of decisions under risk. *Econometrica*, 47:263–291.

Kelly, J. L. (1956). A new interpretation of information rate. *AT&T Tech. J.*, 35:917–926.

LeBaron, B. (2001). Empirical regularities from interacting long and short horizon investors in an agent-based stock market. *IEEE Transactions on Evolutionary Computation*, 5:442–455.

Magdon-Ismail, M., Nicholson, A., and Abu-Mostafa, Y. (2001). Learning in the presence of noise. In Haykin, S. and Kosko, B., editors, *Intelligent Signal Processing*, pages 120–127. IEEE Press.

Michalewiçz, Z. (1999). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin.

Monett, D. (2004). +CARPS: Configuration of Metaheuristics Based on Cooperative Agents. In Blum, C., Roli, A., and Sampels, M., editors, *Proc. of the 1st Int. Workshop on Hybrid Metaheuristics, at the 16th European Conference on Artificial Intelligence, (ECAI '04)*, pages 115–125, Valencia, Spain. IOS Press.

Navarro, J. E. and Schweitzer, F. (2003). The investors game: A model for coalition formation. In Czaja, L., editor, *Proc. of the Workshop on Concurrency, Specification & Programming, CS&P '03*, volume 2, pages 369–381, Czarna, Poland. Warsaw University Press.

Navarro-Barrientos, J. E., Cantero-Alvarez, R., Rodrigues, J. F. M., and Schweitzer, F. (2008). Investments in random environments. *Physica A*, 387(8-9):2035–2046.

Richmond, P. (2001). Power law distributions and dynamic behaviour of stock markets. *The European Physical Journal B*, 20(4):523–526.

Schulenburg, S. and Ross, P. (2001). Strength and money: An LCS approach to increasing returns. In *Lecture Notes in Computer Science*, volume 1996, pages 114–137. Springer.

Turing, G. (1960). An introduction to matched filters. *IEEE T. Inform. Theory.*, 6(3):311–329.

Yang, S. (2005). Experimental study on population-based incremental learning algorithms for dynamics optimization problems. *Soft Computing*, 9(11):815–834.