# An overview on the termination conditions in the evolution of game bots

A. Fernández-Ares, P. García-Sánchez, A. M. Mora, P. A. Castillo, J. J. Merelo, M. G. Arenas, and G. Romero

Dept. of Computer Architecture and Technology, University of Granada, Spain {antares,pablogarcia,amorag,pacv,jmerelo,mgarenas,gustavo}@ugr.es

Abstract. Evolutionary Algorithms (EAs) are frequently used as a mechanism for the optimization of autonomous agents in games (bots), but knowing when to stop the evolution, when the bots are good enough, is not as easy as it would a priori seem. The first issue is that optimal bots are either unknown (and thus unusable as termination condition) or unreachable. In most EAs trying to find optimal bots fitness is evaluated through game playing. Many times it is found to be noisy, making its use as a termination condition also complicated.

This paper summarizes our previous published work where we tested several termination conditions in order to find the one that yields optimal solutions within a restricted amount of time, to allow researchers to compare different EAs as fairly as possible. To achieve this, we examined several ways of finishing an EA who is finding an optimal bot design process for a particular game, Planet Wars in this case, with the characteristics described above, determining the capabilities of every one of them and, eventually, selecting one for future designs.

**Keywords:** Videogames, RTS, evolutionary algorithms, termination criteria, noisy fitness

## 1 Introduction

Evolutionary Algorithms (EAS) are one of the methods usually applied to find the best autonomous agent for playing a game, i.e. the best *bot* [1, 2] through a process mimicking the natural evolution of the species. As in any other algorithm, the termination condition is a key factor as the rest of the experimental setup since it affects the algorithmic performance, with respect to the quality of the yielded solution, and also to the amount of resources devoted to the run. The usual stopping criterion in EAs [3] is reaching a constant number of generations (or evaluations), which is normally related to a fixed *computing power budget* for carrying out the run. Another usual approach is based in a number of generations in which the best solution is not improved or the distance to the optimum is not reduced [4]. However, neither of them might be useful in certain kind of problems such as games, mainly due to the noisy nature of the fitness function [2].

Noise and optimal fitness reachability, are normally not taken into account when choosing how to stop the evolution process. Usual approach is to use a fixed

#### 2 A. Fernández-Ares et al.

number of evaluations or a fixed amount of time, usually given by the game or challenge constraints.

In this paper we present a summary of our previous work [5], where we tried to solve these issues by the introduction of novel stopping criteria for the EAs. They are compared against classical ones, and among themselves when trying to generate competitive bots for video games using Genetic Programming (GP) [6], as this method has proved to be quite flexible and has obtained good results in previous works [7].

*Planet Wars*<sup>1</sup> game was chosen in our experiment, as it is a simple Real Time Strategy (RTS) combat-based game (only one type of resource, one type of attack and one type of unit), and also it has been widely used in the literature, using different generation methods and fitness functions [8–11]. This game fulfills the next two conditions: initial position of bots is random and the decisions are stochastic, although the result of the combat is deterministic.

Summarizing, the objective of the study presented in [5] (and outlined in this work) was to find a stopping criteria that converges to optimal solutions and that is independent of the method chosen. To measure the quality of every approach, we considered time, or number of generations, needed to obtain the solution and the quality of that solution.

# 2 Methodology and experimental setup

As previously stated, in our described work [5] we proposed different termination criteria based on different EA features, such as the parameters of the algorithm (maximum number of generations) or the population (improvement, replacement or age). A *Score Function* was proposed in order to measure the quality of a generated bot (a solution or individual in the algorithm). This scoring method tries to reduce the effects of the noisy evaluation (following the guidelines of other works [12]) by computing fitness from the result of 30 different matches against an expert rival. Thus, the function considers the number of victories, turns to win and turns resisted before being defeated by the opponent (in the case of lose). The rival is *ExpGenebot* [13], and based on the improvement of the heuristics proposed by a human player.

The fitness of each individual i of the population is obtained using the next formula where N is the number of simulations<sup>2</sup>:

$$Score_i = \alpha + \beta + \gamma$$
 (1)

$$\alpha = v, \alpha \in [0, NB] \tag{2}$$

$$\beta = NB \times \frac{t_{win} + \frac{1}{N \times t_{MAX} + 1}}{\frac{t_{win}}{v + 1} + 1}, \beta \in [0, NB], t_{win} \in [0, NB \times t_{MAX}]$$
(3)

<sup>1</sup> http://planetwars.aichallenge.org/

 $<sup>^2\,</sup>$  The '1' in all denominators is used to avoid dividing by 0 and for the ratio calculation.

An overview on the termination conditions

$$\gamma = \frac{t_{defeated}}{NB \times t_{MAX} + 1}, \gamma \in [0, 1], t_{defeated} \in [0, NB \times t_{MAX}]$$
(4)

The terms used are: the number of battles (NB) to test, the number of victories of the individual against ExpGenebot (v), the sum of turns used to beat ExpGenebot  $(t_{win})$  in winners simulations, the sum of turns when the individual has been defeated by ExpGenebot  $(t_{defeated})$  in losing simulations and the maximum number of turns a battle lasts  $(t_{MAX})$ .

GP algorithm evolves a binary tree formed by *decisions* (logical expressions that evaluate the current state of the game) and *actions* (the leafs of the tree: the amount of ships to send to a specific planet). This tree is evaluated in each player's planet, analysing the current state of the map/planet (decision), and how many ships send from that planet to an specific target planet (action). These target planets can be the wealthiest, the closest one, etc. owned by the player or the enemy, or neutral. The possible actions and decisions are listed in [7]. The complete set of used parameters were described in the work we are summarizing here [5].

We designed a set of five different algorithm stop criteria which were checked in the paper, namely:

- [NG] Number of Generations: it is the classical termination criteria in evolutionary algorithms: 30, 50, 100 and 200 generations.
- [A0] Age of Outliers: if the age of individual is an outlier in the comparison with the rest of the population then it would be potentially an optimal solution and the algorithm can be stopped: 1, 1.5, 2 and 2.5 times the interquartile range (IQR)
- [RT] Replacement Rate: when using an elitist strategy in which individuals are replaced only if the offspring is better, the fact that the population stops generating better individuals might be a sign of stagnation:  $\frac{n}{2}$ ,  $\frac{n}{4}$ ,  $\frac{n}{8}$ , and  $\frac{n}{16}$ .
- [FT] Fitness Threshold: a maximum value to obtain in the evolution could be set considering the top limit of the score function: 20 (as half the maximum:  $MAX_{SC}/2$ ), 30 (as half the maximum score plus half this value:  $MAX_{SC}/2 + MAX_{SC}/4$ ), and the division in four parts of the interval these values compose: 22, 24, 26 and 28.
- [FI] Fitness Improvement: if the best fitness is not improved during a number of generations, the algorithm must stop. Four possible values will be tested: 3, 7, 10 and 15 generations.

# 3 Experiments and Results

The experiments conducted involved 36 runs, each of them configured including all the defined stop criteria, and also adding an extra termination one, i.e. getting to 500 generations, in order to avoid non-ending runs.

The results showed the absence of some of the commented stop values for the *Fitness Threshold* criterion, namely 28 and 30, because they were not reached in

3

#### 4 A. Fernández-Ares et al.

any of the runs. The results also showed that all the criterion are well defined, since all the scores grow in every criterion block, so as more restrictive the criterion is (they are met with a lower probability), the higher the obtained score is. The score function worked as expected, even with the presence of noise. This is true in all the cases except in the *Age of Outliers* which is so far the criterion with the worse results, as it is also proved by the statistical test which does not find significant differences between the scores obtained by every age-based criterion and the previous and next ones. The *Replacement Rate* criterion yielded the best distribution of results, with a clear fitness improvement tendency and a very good maximum score, close to that obtained by the *Fitness Thresholds*.

In addition to these two studies, a new measure factor was computed by means of a benchmark based in battles against a different competitive bot available in the literature [13] in 100 maps (some of them used during the evolution). This test was conducted for the best individual obtained when every criterion was met in every run, thus 20x36 bots have been tested. The best results were vielded by the most restrictive of each criterion, highlighting the FT results. This is also a reinforcement to the correctness of the score function. Score and generations measures are compared in Figure 1. As it can be seen, an improvement in the fitness/score means a higher number of generations are required. This happens in almost all the cases, with some exceptions such as some of RTcriteria, which get a higher score value in less generations than other criteria. However, this happens due to the commented problem of noise. Finally Table 1 presents all the results as a summary. It also shows the completion rate of every stop criterion. Moreover, a comparative set of values was computed, considering the number of generations equal to 30 (usual in previous papers) as the standard value to relativize the rest.

The FI criterion is useful to 'detect' local optima. Increasing the restriction value of this method lets the EA more generations to escape from a local optimum, obtaining significantly better results. As the EA can quickly converge to a local optima, using this method could be equivalent to set a fixed (but unknown) number of generations, enough to detect a stagnation in the population (that can be useful in some evolutionary approaches). However, the results show that this criterion has stopped in local optima that other methods have surpassed. Finally, RT provides the best results considering all metrics: generations, score and completion rate. It is based on replacement rate, so it indirectly measures how the whole population increases their abilities, without explicitly measure the average fitness. This is useful in this kind of problems, i.e. where there is a noisy fitness function and the optimal solution is unknown.

## 4 Conclusions

Using Evolutionary Algorithms (EAs) to generate bots for playing games have two main issues: the fitness is noisy and optimal bots are either not known or unreachable. This makes it difficult to find a good stopping criterion for the EA. In [5] four different stopping criteria, based in fitness and in the population,



| Stopping  | Absolute |       |       | Relative |      |      | P    |
|-----------|----------|-------|-------|----------|------|------|------|
| Criteria  | G        | S     | V     | G        | S    | V    | п    |
| NG_030.0  | 30.00    | 16.31 | 45.92 | 1.00     | 1.00 | 1.00 | 1.00 |
| NG_050.0  | 50.00    | 17.80 | 52.72 | 1.70     | 1.09 | 1.15 | 1.00 |
| NG_100.0  | 100.00   | 19.21 | 57.25 | 3.37     | 1.18 | 1.25 | 1.00 |
| NG_200.0  | 200.00   | 20.25 | 58.39 | 6.70     | 1.24 | 1.27 | 1.00 |
| AO_1.0    | 8.83     | 13.46 | 35.89 | 0.29     | 0.83 | 0.78 | 1.00 |
| AO_1.5    | 10.33    | 14.07 | 34.83 | 0.34     | 0.86 | 0.76 | 1.00 |
| AO_2.0    | 14.61    | 14.93 | 38.08 | 0.49     | 0.92 | 0.83 | 1.00 |
| AO_2.5    | 17.17    | 15.30 | 39.92 | 0.57     | 0.94 | 0.87 | 1.00 |
| RT_n/02   | 2.00     | 10.20 | 20.58 | 0.07     | 0.63 | 0.45 | 1.00 |
| $RT_n/04$ | 5.47     | 12.21 | 28.17 | 0.18     | 0.75 | 0.61 | 1.00 |
| $RT_n/08$ | 78.64    | 18.16 | 50.94 | 2.62     | 1.11 | 1.11 | 1.00 |
| RT_n/16   | 248.21   | 21.34 | 62.92 | 8.27     | 1.31 | 1.37 | 0.66 |
| FT_20.0   | 55.08    | 20.62 | 54.22 | 1.84     | 1.26 | 1.18 | 1.00 |
| FT_22.0   | 127.56   | 22.65 | 58.25 | 4.25     | 1.39 | 1.27 | 1.00 |
| FT_24.0   | 276.71   | 24.39 | 63.39 | 9.22     | 1.50 | 1.38 | 0.77 |
| FT_26.0   | 378.88   | 26.45 | 74.75 | 12.63    | 1.62 | 1.63 | 0.22 |
| FI_03.0   | 10.31    | 13.34 | 30.00 | 0.34     | 0.82 | 0.65 | 1.00 |
| FI_07.0   | 24.56    | 15.54 | 41.39 | 0.82     | 0.95 | 0.90 | 1.00 |
| FI_10.0   | 35.47    | 16.50 | 47.94 | 1.18     | 1.01 | 1.04 | 1.00 |
| FI_15.0   | 52.22    | 17.56 | 53.00 | 1.74     | 1.08 | 1.15 | 1.00 |

Fig. 1: Average score of the best individual and average reached generations per termination criteria.

Table 1: Average results of every criterion for the three measures: Number of generations (G), Score (S), and Victories in benchmark (V); plus the Completion rate in experiments (R). Relative values are computed with respect to  $NG_{-}30.0$ .

were tested and compared with the classical approach of the fixed number of generations. This paper summarizes the contents of that previous work.

Several experiments were conducted, using different metrics based in a score function, the number of generations reached for each criterion, and the number of victories that the best yielded bots per criterion have obtained against an external rival (not the same used in the fitness computation). According to the results, initially, a stopping criterion based in *Fitness Threshold* would be the most desirable option, as it attains the best score. However, in this kind of problems, it is quite difficult to find an optimal fitness value to use (normally it is unknown). Therefore, the best option would be using a *Replacement Rate* as stopping criterion, since it is a compromise solution which relies in the population improvement without implicitly use the fitness.

As future work, new problems (and algorithms) will be addressed to validate the proposed stopping criteria, using different environments and new score functions. In addition, mechanisms to improve the EA will be used in conjunction with the proposed methods, for example, increasing the search space when a stagnation of the population is detected.

5

6 A. Fernández-Ares et al.

### Acknowledgments

This work has been supported in part by SIPESCA (Programa Operativo FEDER de Andalucía 2007-2013), TIN2011-28627-C04-02 (Spanish Ministry of Economy and Competitivity), SPIP2014-01437 (Dirección General de Tráfico), PRY142/14 (Fundación Pública Andaluza Centro de Estudios Andaluces en la IX Convocatoria de Proyectos de Investigación), PYR-2014-17 GENIL project and V17-2015 of the Microprojects program 2015 (CEI-BIOTIC Granada).

## References

- Small, R., Bates-Congdon, C.: Agent Smith: Towards an evolutionary rule-based agent for interactive dynamic games. In: Evolutionary Computation, 2009. CEC '09. IEEE Congress on. (2009) 660–666
- Mora, A.M., Montoya, R., Merelo, J.J., Sánchez, P.G., Castillo, P.A., Laredo, J.L.J., Martínez, A.I., Espacia, A.: Evolving Bots AI in Unreal. In di Chio et al., C., ed.: Applications of Evolutionary Computing, Part I. Volume 6024 of Lecture Notes in Computer Science., Istanbul, Turkey, Springer-Verlag (2010) 170–179
- 3. Bäck, T.: Evolutionary algorithms in theory and practice. Oxford University Press (1996)
- Roche, D., Gil, D., Giraldo, J.: Detecting loss of diversity for an efficient termination of eas. In: 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2013, Timisoara, Romania, September 23-26, 2013, IEEE (2013) 561–566
- Fernández-Ares, Á., García-Sánchez, P., Mora, A.M., Valdivieso, P.A.C., Guervós, J.J.M., Arenas, M.I.G., Romero, G.: It's time to stop: A comparison of termination conditions in the evolution of game bots. In: Applications of Evolutionary Computation. Volume 9028 of Lecture Notes in Computer Science., Springer (2015) 355–368
- 6. Koza, J.R.: Genetic Programming: On the programming of computers by means of natural selection. MIT Press, Cambridge, MA (1992)
- García-Sánchez, P., Fernández-Ares, A.J., Mora, A.M., Castillo, P.A., Merelo, J.J., González, J.: Tree depth influence in genetic programming for generation of competitive agents for rts games. In: EvoApplications, EvoStar. (2014) 411–421
- Mora, A.M., Fernández-Ares, A., Guervós, J.J.M., García-Sánchez, P., Fernandes, C.M.: Effect of noisy fitness in real-time strategy games player behaviour optimisation using evolutionary algorithms. J. Comput. Sci. Technol. 27(5) (2012) 1007–1023
- Fernández-Ares, A., Mora, A.M., Guervós, J.J.M., García-Sánchez, P., Fernandes, C.: Optimizing player behavior in a real-time strategy game using evolutionary algorithms. In: IEEE C. on Evolutionary Computation, IEEE (2011) 2017–2024
- R, L.C., Cotta, C., Fernández-Leiva, A.: On balance and dynamism in procedural content generation with self-adaptive evolutionary algorithms. Natural Computing 13(2) (2014) 157–168
- Nogueira-Collazo, M., C., C., Fernández-Leiva, A.: Virtual player design using self-learning via competitive coevolutionary algorithms. Natural Computing 13(2) (2014) 131–144
- Mora, A.M., Fernández-Ares, A., Guervós, J.J.M., García-Sánchez, P., Fernandes, C.M.: Effect of noisy fitness in real-time strategy games player behaviour optimisation using evolutionary algorithms. J. CST. 27(5) (2012) 1007–1023
- Fernández-Ares, A., García-Sánchez, P., Mora, A.M., Guervós, J.J.M.: Adaptive bots for real-time strategy games via map characterization. In: CIG, IEEE (2012) 417–721