Evolving Racetrack Knowledge in a Racing Game

Sam Redfern

Discipline of Information Technology, College of Engineering & Informatics, National University of Ireland, Galway, Ireland.
sam.redfern@nuigalway.ie

Abstract. This paper discusses the use of a Genetic Algorithm (GA) to evolve waypoint-based racing lines for computer-controlled cars in an online game. One goal is the reduction of developer effort, in order to automatically produce varied and effective racing lines for numerous different vehicle configurations on various racetracks. Another goal is to assess the design of fitness functions in order to evolve varying strategies and driving styles for the computer-controlled drivers. The effectiveness of this approach is assessed, and the evolved driving strategies on a number of specific racetracks are analysed.

Keywords: Computational Intelligence; Genetic Algorithms; Game A.I.; Vehicle Simulation

1 Introduction

In recent years, various Computational Intelligence (CI) techniques have been applied to computer racing games, with the dual purpose of improving the games themselves and of using the games as an appropriate platform to research CI techniques. Togelius et al. [1] provide a useful overview of recent work in the area, while making the point that CI techniques have been "conspicuously absent" in commercial games.

Racing games, like many other computer game genres, don't necessarily require CI techniques in order to produce better-performing computer-controlled agents, since the game can easily enough beat the human player through subtle cheating [1]. Rather, the motivation is often to make the behaviour of the computer player more interesting, varied and believable. Evolutionary algorithms may be used as search-based design tools rather than as traditional optimisers.

Modern racing games are increasingly sophisticated, employing complex physical driving models and collision systems – as a result, it is increasingly difficult to produce effective computer agents through the use of scripted AI – this is one of the underlying reasons for the increased interest in CI techniques in this area [2].

In this paper, the use of Genetic Algorithms (GA) as part of a multi-layer CI approach to evolving waypoint-based racing drivers is discussed. The platform for this research is the game Darkwind: War on Wheels, which is an online racing and combat driving game developed by the author since 2005 [3, 4]. Darkwind supports a detailed physical driving model and provides a set of demanding off-road racetracks.

Since it is an online game with hundreds of currently active players, and in which upwards of 1000 separate races are run every week, Darkwind also provides an excellent opportunity to compare computer agents with expert human players – unlike other academic CI racing platforms, which have had to make do with one or two players with only a few hours play-time experience.

Although we use a number of scripted, mimicry and evolutionary techniques in Darkwind in order to produce challenging computer opposition in combat and race situations, the focus of this paper is on the evolution of waypoints for computer agents in control of a variety of different vehicles on a variety of racetracks. From the perspective of the game developer, one of the main motivations for this research is to reduce the development effort required to produce effective drivers: the GA approach is used in its classic optimisation role. It is also a goal to efficiently produce a database of interesting and varied sets of waypoints that can be used to control different cars in a race, making the computer-controlled cars act less predictably. By varying the fitness function of the GA, we are able to produce behaviours that are optimised in different ways, and that therefore exhibit different driving styles.

2 Related Research

A number of previous researchers have used CI techniques to evolve effective and interesting computer agents for racing games. Several have used virtual range-finders as the primary input to neural networks which control the steering and acceleration/braking [5, 6] or to provide collision-warning systems [7]. Some have used pre-calculated waypoint data as inputs in addition to range-finders, but have not attempted evolution of the waypoints themselves [8]. Reasonable success has been reported at evolving track-specific drivers, although creating generalised controllers capable of performing well on multiple tracks (or on previously unseen tracks) has proven elusive [5]. At least one commercial game (Colin McRae Rally) has reportedly used neural networks to control computer drivers, although details are sketchy [9].

Recent work by Saez et al. has applied GAs to the car driving task. Their genome codified steering and acceleration decisions at 5 metre intervals on a racetrack, and showed some promise although a serious lack of robustness was reported (their system was not able to deal with noisy data or varied initial car state, and was unable to attempt multiple laps) [10].

In the majority of previous research, the design of the control system under evolution has been quite low-level and not human-readable. Therefore prior knowledge has not typically been used to seed the initial state of the control system, and randomised initial parameters have been used. This certainly leads to impressive-looking fitness curves, which belie the fact that the main bulk of the generations being produced are totally inadequate at their task. Since our own optimisation begins with a quite effective manually defined initial generation, we witness a far shallower fitness curve.

3 Evolving Racetrack Knowledge

3.1 Physical Model

In our research, a sophisticated physical model of vehicle dynamics has been employed: this includes an approximation of vehicle aerodynamics (downforce, air resistance and slipstreaming), variable tyre characteristics (lateral and longitudinal deformations, static and kinetic friction, performance degradation and damage), suspension (length, spring forces), engine performance curves, and rigid-body collision resolution. In addition, our racetracks employ a number of types of surface with varying friction and solidity, and contain not only obstacles but also in some cases quite varied terrain formations.

The complexity of this system provides a challenging game with keenly contested races and on-line lap records: on many tracks, the emphasis is on maintaining a good velocity while cornering, or on attacking a terrain feature such as a jump in an optimal fashion. On other tracks the emphasis may involve minimizing tyre-wear. This complexity means that creating effective and interesting computer drivers through scripted calculations or algorithmically defined racing lines is infeasible.

3.2 Vehicle Control Algorithm

Unlike previous CI researchers, our approach has assigned the actual steering and acceleration/deceleration decisions to a higher-level function where we believe it is more appropriately controlled. Rather than evolving low-level steering decisions, we therefore evolve representations of effective racing lines for specific track/car combinations through the use of waypoints. This provides an evolved understanding of the overall shape of the track, rather than attempting to produce a controller that navigates with a paucity of positional information or knowledge of the ‘big picture’.

Our computer drivers use these waypoints during the game to influence their behaviour as they navigate static and moving obstacles on the track. Steering behaviour is calculated based on the location of the waypoints and on the linear and angular momentum of the car, rather than being directly controlled by a neural network. We use collision-prediction in order to take evasive action, but with waypoints still providing the over-riding goals. This is quite different to previous research involving waypoints and collision prediction, which has afforded equal prominence to both in the evolution of the control system [5], and which has not allowed for evolution of the waypoints themselves [5, 8].

Fig. 1 illustrates a set of waypoints on a tight, fast corner, which has concrete blocks at its apex. The waypoints encourage the car to approach the corner from the outside of the track, and to turn into the corner early and gently (aiming for waypoint 9), thereby minimising skidding and maintaining a good speed. The blocks are avoided by the influence of waypoint 10; this waypoint also encourages the car to hit an ideal speed of 96mph, which proves low enough for this specific car with these specific tyres to exit the corner without crossing from the tarmac surface to the sandy margin and concrete barriers on the right. Waypoints 11, 12 and 13 encourage the car to stay on the tarmac, and the car exits on the ideal side of the road for the subsequent straight section and left-turning corner (out of picture). The final exit speed of 78mph is considered almost-optimal for this vehicle.

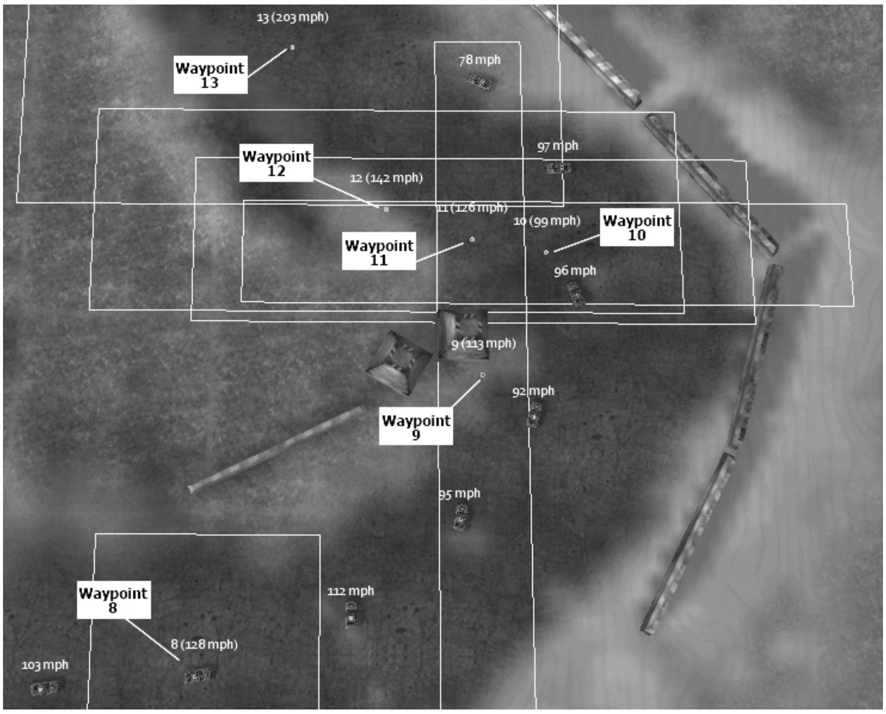

Fig. 1. An evolved set of racing waypoints on a dusty tarmac surface. The car aims for the centre of each waypoint (numbered 8 through 13) while targeting the waypoint's indicated speed. As soon as the boundary of a waypoint is crossed, the proceeding waypoint becomes the target. The position, size, and target speeds of the waypoints are evolved using a GA. The position and speed of a car over a period of 8 seconds of the game is illustrated.

Two types of collision prediction operate as interrupts to the higher level waypoint-following goal. Firstly, range finders are cast into the world, with directions and distances based on the momentum of the car; where a static object is identified nearby, waypoint steering is over-ridden with collision-avoidance steering. Secondly, the momentum of all nearby moving objects (i.e., other vehicles) is assessed for a period of two seconds into the future, and where the bounding boxes overlap that of the car under control, evasive steering takes over.

3.3 Evolutionary Approach

Each GA genome consists of a set of waypoints stored as real numbers. The data for each waypoint consists of: centre position (x, y), dimensions (x, y), and target speed.

Depending on the size and complexity of the racetrack, the number of waypoints used may vary from about 20 to about 60. All genomes in a population contain the same number of waypoints.

Each candidate genome is tested 10 times, with the first lap in each sequence disregarded so as to allow the cars to achieve optimum speed by the start of a timed lap. Collision damage is ignored during training, although tyre-wear due to excessive skidding or driving over rocky ground is not. Even without damage, a car suffering a collision will obtain a very poor laptime due to lost momentum, spinning and so on – we did not wish to further penalise all subsequent laps by damaging the car's physical condition.

Our algorithm uses ranked rather than fitness-proportionate selection, in order to discourage premature convergence. Elitism is applied so that the best ranking 50% of the population is copied unaltered to the proceeding generation, while the remaining 50% is discarded and replaced with new genomes. Parents for each new genome are selected from the elite 50%, with a bias towards the highest ranked. Crossover is chosen at a random point in the gene, but only where the following additional crossover compatibility test is passed.

For a crossover point at waypoint *x* to be valid for parents *p1* and *p2*, where *p1[x]* represents waypoint *x* of parent *p1*:

1. The distance from *p1[x-2]* to *p2[x]* must be larger than the distance from *p1[x-1]* to *p2[x]*
2. The distance from *p1[x-1]* to *p2[x+1]* must be larger than the distance from *p2[x]* to *p2[x+1]*

This test ensures that the crossover point does not cause the car to turn around on the racetrack and start driving in the wrong direction.

The fitness function itself may be chosen so as to evolve drivers with different characteristics; this, and other parameters under investigation, is discussed in the next section.

4 Results

4.1 Overall Performance

A primary performance measurement involves the assessment of laptimes achieved by evolved sets of waypoints versus the original manually defined waypoints deployed by the racetrack designers. Having experimented with various mutation schema, numbers of waypoints, fitness functions and population sizes, an improvement of between 8% and 30% in average laptime over 10 laps has been observed on all racetracks and vehicle types.

The most effective GA parameters for optimising on each track and vehicle do vary somewhat, but this typically affects the efficiency of search rather than the final result. The computer has not yet beaten the best laptimes recorded by human players over the past 4 years. It should be noted that in many cases, close to 100,000 laps have been attempted by human players on each racetrack.

4.2 Influencing Driving Styles

By varying the fitness function, we can evolve driving styles varying from cautious (by defining fitness as the worst laptime of the 10 laps recorded), to optimal on average (by using the average laptime), to risk-taking (by using the best laptime). This allows different personalities to be presented in the game, and perhaps for some race-time strategy to be attempted, depending on the current state of the race.

Fig. 2 illustrates the same corner as that shown in Fig. 1. The fitness function has been changed from average laptime to best laptime, and therefore a more aggressive racing line has evolved, which encourages the car to drive closer to the dangerous apex and hold a higher speed, risking tyre-wear and spin-outs, yet performing better when successful, and yielding a higher exit speed from the corner.

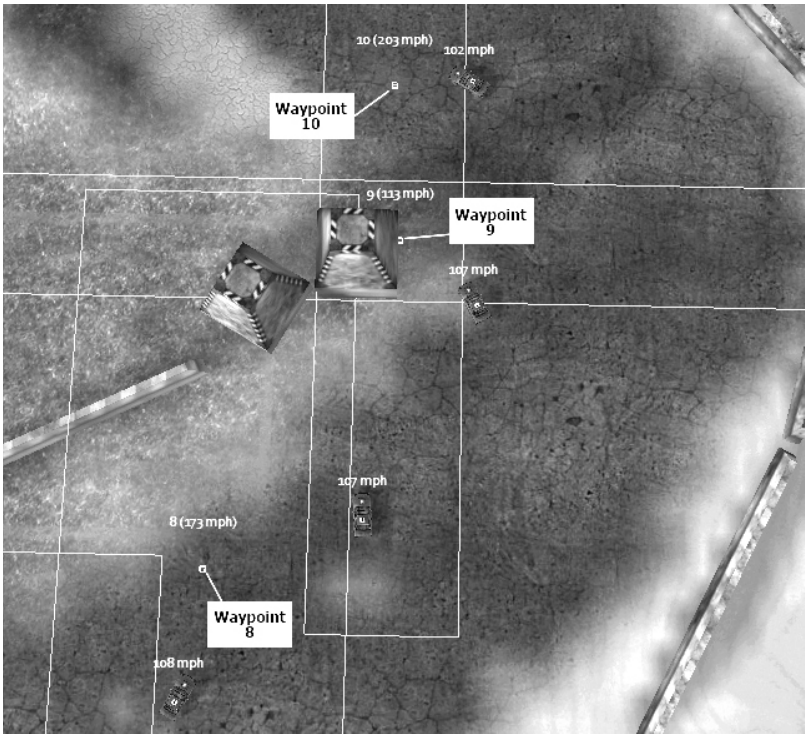

Fig. 2. The same corner is depicted as in Fig. 1. In this case, the best rather than average laptime has been used as the fitness function.

Fig. 3 shows the average fitness of the population and best laptime achieved overall using (A) best laptime as fitness function, and (B) average laptime as fitness function.

In (A), we evolve a risk-taking strategy which quickly converges on good laptimes while maintaining a poor average (since this contains frequent crashes and spin-outs). By using the average of 10 laptimes as the fitness function (B) a better result is obtained on average, in which the occurrence of disastrous errors is minimised at the expense of optimal laptime. Also shown is the fitness of the original, developer-defined (Manual) waypoints, and the best ever laptime recorded by a human over the past 4 years.

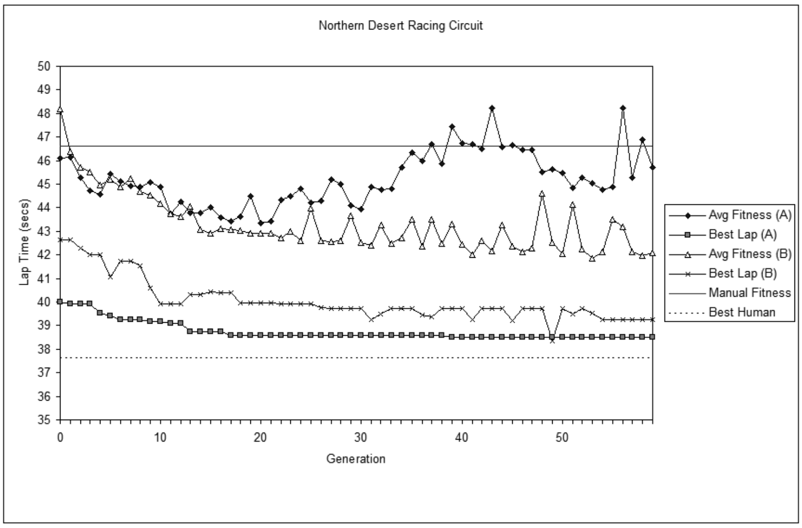

Fig. 3. The best laptime taken from 10 attempts is used as the fitness function in (A), while the average of 10 attempts is used as the fitness function in (B), yielding distinct behaviours.

4.3 Expert Knowledge of Racetracks

Most of the racetracks in the game have a number of critical features where the success or failure of a race is often decided. Only the leading human players have obtained mastery of these features, and certainly without GA optimisation it is very unlikely that computer drivers will perform well on them. In the following sub-sections we explore a few racetracks on which experiments have been run.

4.3.1 Dirt Racing Track

This is a dusty, bumpy track with many elevation changes. The final straight leading to the finishing line is a steep, bumpy hill with an adverse camber, which tends to tip the car outwards into a collision course with the finishing line. Compensating for this often causes computer drivers and novice players to oversteer or spin their car and lose their momentum. The evolved waypoints encouraged the computer driver to stay close to the left side of the track (which is the higher ground) and to avoid straying onto the sloping part of the road. This was achieved through use of a large number of tightly packed waypoints close to the left hand edge.

Halfway around this track is a large bump with a pit on one side. Taking this section badly typically means a higher jump, heavier landing, and a substantial loss of momentum. The evolved computer drivers were found to swing wide before the bump and then attack its low-point at an approximately 45 degree angle: this minimizes the height that the car jumps, and maximizes its exit speed.

On this circuit, the best laptime recorded using the evolved racing lines with a muscle-car was 32.03 seconds, which is a 10.44 seconds (24.6%) improvement on the developer-defined waypoints. The severely bumpy/hilly nature of this track made the original waypoints generalize poorly for the various vehicles – the efficiency of using GAs to produce per-vehicle optimised versions is clear. The best ever human-recorded laptime using the same vehicle is 30.87 seconds.

4.3.2 Northern Speedway

This is a broadly circular circuit with a very loose surface (desert sand) – it's an easy track to drive but a difficult one to drive optimally. On this circuit, the evolved racing lines included power-slides (drifts) on the two tightest bends – the waypoints were found to be positioned several metres inside the apex, so that the cars used understeering to their advantage to navigate the corners at maximal speed while staying close to the inside of the track.

Overall, the best time achieved by the computer when controlling a fast vehicle was 24.01 seconds, a reduction of 2.67 seconds (9.9%) when compared with the time achieved using the original, developer-defined waypoints. The best-ever human laptime on this track using the same car is 21.69 seconds.

4.3.3 Northern Foothills Racetrack

This is a tight, fairly smooth racetrack with sharp corners. The centre of the track has a good surface but this is typically quite narrow and the margins are sandy – it is very detrimental to a laptime if a car strays onto the margins. The computer-controlled cars are notoriously ineffective on this circuit, due to the difficulty experienced during development in manually defining an optimal set of waypoints.

On this circuit, the best laptime recorded using the evolved racing lines with a road-car was 56.93 seconds, which is an 11.98 seconds (17.4%) improvement on the developer-defined waypoints. The best ever human-recorded laptime is 53.56 seconds.

One hairpin corner in particular is difficult to drive well due to its sandy surface, and can be exited at above 50mph if taken well and if the narrow tarmac track-centre is successfully targeted on the exit. Taken badly, a very low exit speed or a spin-out are common. Fig. 4 illustrates an evolved set of waypoints being used in-game on this corner.

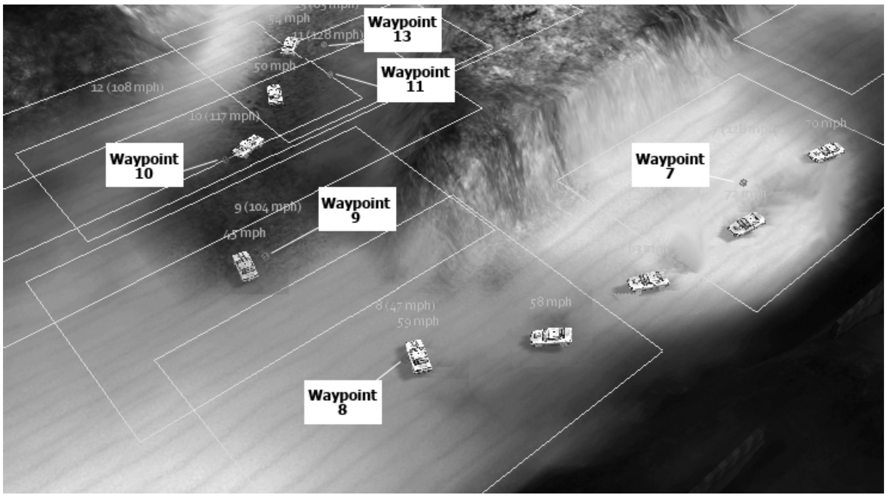

Fig. 4. A difficult hairpin on the Northern Foothills Racetrack. To drive this corner well, a car must target the narrow tarmac centre-line on the exit; this is difficult to do at high speed since the corner itself is so severe and surfaced with sand. The evolved waypoints succeed in obtaining an exit speed of 45mph in this sample run. Note that the strangely-positioned waypoint 12 is 'junk' data that never affects the racing line due to its overlap with waypoint 11.

5 Conclusions & Future Work

The research described in this paper shows that a GA approach to evolving racetrack knowledge has great potential – with improvements of up to 30% over the laptimes achieved when developer-defined waypoints were used. Through the use of varied fitness functions, this approach also enables the evolution of different driving styles and therefore more interesting opponents for the human players.

Our evolved waypoints have not yet beaten the best laptimes recorded by human players over the past 4 years. Observation of the computer driving clearly shows that this is due to the waypoint approach providing a generic understanding of a racetrack that doesn't take into account unique situations during specific test-runs. A clear example of this is on our "Dry Lake Racetrack" which is a fast circuit with concrete blocks positioned in the middle of the ideal racing line: the evolved waypoints cut directly through some of the blocks, which works well at high speeds due to lateral motion of the car on the dusty surface, yet fails at low speeds – the car suffers a head-on collision with the blocks. Some further work is needed here, probably in the form of further A.I. layers that deal with transient conditions. The current collision-prediction systems are an example of an additional layer already in use in this way.

One of the interesting results observed is the variety of different racing lines evolved for different vehicles on a racetrack. Minimum-distance racing lines were often evolved for slower vehicles, while faster vehicles preferred narrower turns which afford maximum speeds. The extent to which bumps and terrain-cover changes were avoided also varies by vehicle type and tyre characteristics.

One of the ideas that we are currently experimenting with is the use of split-times rather than complete lap-times in order to define sub-genome fitness values and use these to control crossover decisions during the breeding process.

The Darkwind game allows certain car characteristics to be modified by the players: for example, tyre pressures and suspension stiffness. We intend to design a GA-based system for optimising vehicle configurations on a per-racetrack basis. A similar approach has shown good performance when applied to car settings in a Formula One simulator [11].

Future work will also focus on multiple-car (i.e., racing) situations rather than single-car experiments. In this case, one of the characteristics that may be evolved for each waypoint is the decision about whether it is safe to undertake evasive steering – at certain dangerous parts of some racetracks, adherence to the racing line is critically important in order to avoid collisions with obstacles, and colliding with other cars would be a lesser problem. Another avenue worthy of investigation is that of the modification of steering behaviour in order to make optimal use of slipstreaming with nearby cars of similar momentum.

References

1. Togelius, J., Lucas, S.M., De Nardi, R.: Computational Intelligence in Racing Games. Studies in Computational Intelligence 71, 39—69 (2007)
2. Cardamone, L., Loiacono, D., Lanzi, P.L.: On-line Neuroevolution Applied to The Open Racing Car Simulator. In: 11th Congress on Evolutionary Computation, pp. 2622--2629. IEEE Press, New York (2009)
3. Redfern, S.: Psychic Software and Darkwind: War on Wheels. In: New Age Gamer Magazine 10(6), pp.32--35. South Africa (2007).
4. Smith, Q.: Road Rage: Darkwind Provides a Ride Like No Other. In: PC Gamer Magazine 209, Jan 2010, p.96. Future Publishing, London (2010).
5. Togelius, J., Lucas, S.M.: Evolving Robust and Specialized Car Racing Skills. In: Proc. IEEE Congress on Evolutionary Computation. IEEE Press, New York (2006)
6. Togelius, J., Lucas, S.M.: Evolving Controllers for Simulated Car Racing. In: Proc. IEEE Congress on Evolutionary Computation, pp.1906--1913. IEEE Press, New York (2005)
7. Stanley, K.O., Kohl, N., Sherony, R., Miikkulainen, R.: Neuroevolution of an Automobile Crash Warning System. In: Proc. Genetic And Evolutionary Computation Conference, pp. 1977--1984. ACM, New York (2005)
8. Chaperot, B., Fyfe, C.: Improving Artificial Intelligence In a Motocross Game. In: IEEE Symposium on Computational Intelligence And Games. IEEE Press, New York (2006)
9. Cardamone, L., Loiacono, D., Lanzi, P.L.: Evolving Competitive Car Controllers for Racing Games with Neuroevolution. In: 11th Congress on Evolutionary Computation. IEEE Press, New York (2009)
10. Saez, Y., Perez, D., Sanjuan, O., Isasi, P.: Driving Cars by Means of Genetic Algorithms. In: Proc. 10th Int. Conf. On Parallel Problem Solving From Nature, pp.1101--1110. Springer: Heidelberg (2008).
11. Wloch, K., Bentley, P.J.: Optimising the Performance of a Formula One Car using a Genetic Algorithm. In: Proc. 8th Int. Conf. On Parallel Problem Solving From Nature, pp.702--711. Springer: Heidelberg (2004).