

Driving Cars by Means of Genetic Algorithms

Yago Saez¹, Diego Perez¹, Oscar Sanjuan², and Pedro Isasi¹

¹ Carlos III University, Madrid, 28911, Spain

² Oviedo University, Oviedo, 33002, Spain



Abstract. The techniques and the technologies supporting Automatic Vehicle Guidance are an important issue. Automobile manufacturers view automatic driving as a very interesting product with motivating key features which allow improvement of the safety of the car, reducing emission or fuel consumption or optimizing driver comfort during long journeys. Car racing is an active research field where new advances in aerodynamics, consumption and engine power are critical each season. Our proposal is to research how evolutionary computation techniques can help in this field. As a first goal we want to automatically learn to drive, by means of genetic algorithms, optimizing lap times while driving through three different circuits.

1 Introduction

Automatic Vehicle Guidance (AVG) has been addressed by numerous researchers, i.e. [12], [15], [17], [13], [16], [20], as an engineering problem through different approaches, and the most promising ones are being engineered on real prototypes, i.e., [13], [5], the Buick from the California PATH project or [7], or the cars which participate each year in the annual competition organized by the Defense Advanced Research Projects Agency (DARPA) for driverless cars, [6]. The study of the suspension system has been an interesting topic for researchers because it contributes to the car's handling and braking for good active safety and driving pleasure, and keeps vehicle occupants comfortable and reasonably well isolated from road noise, bumps, and vibrations. This parameter optimization has been done with Genetic Algorithm (GA) with promising results in several works, such as, [8], [9] and [24]. One of the first works on this topic was carried out at the Carnegie Mellon University by Sukthankar et al. in 1996, [16]. This work uses reasoning modules which combine high level task goals with low-level sensor constraints, proposing a Population Based Incremental Learning (PBIL) [2] for an automatic setting of the module parameters. For the simulation the system uses a program called SHIVA (Simulated Highways for Intelligent Vehicle Algorithms) which reproduces a micro-simulation of vehicles moving and interacting in a user-defined roadway. Two years later the PBIL was compared to the GA in this same framework, [3]. The results of these works were quite motivating since they showed the potential for intelligent behavior in tactical driving. Other interesting work carried out in 1996 by Pyeatt et al. from Colorado University deals with simulated race car driving [14]. In this case a study about autonomous driving was developed

based upon RARS simulator software (the one which has influenced TORCS). The results showed that the RARS simulator was adequate for developing the test framework and that neural networks can be competitive techniques for producing autonomous racing cars.

In 1998, Bernard et al. [4] from Iowa State University illustrated the power of GAs to model driver/vehicle behavior. In fact, their work determined how fast and safe a given vehicle model could be. The experimental framework was to drive through a short course without hitting a cone or lifting its wheels. In 2004, Floreano et al., [10], used a GA for tuning up a neural network which visually recognizes edges, corners and height, resembling strategies observed in simple insects which obtained results that performed equal or better than well trained human drivers tested on the same circuits. In 2005, Sun et al. [18], used a GA for optimizing the parameters of a set of Gabor filters in the context of vehicle detection from images. They successfully tested the proposed framework on real data and improved the performance of on-road vehicle detection. In the same year another interesting approach was proposed for the automated evolutionary design of driving agents, [11], [19]. This work showed how GAs can help in the task of designing an agent able to remotely operate a scale racing car. This work revealed that on long runs the agent's operated car was 5% slower than the human operated one. Working with the evolving weights of a neural network, Julian Togelius et al. compared, with their own simulation, simulated cars with evolved neural network controllers (in first-person and third-person) [23], [22]. They extended their work to a more complex case of two cars competing against each other in the same track at the same time, [21], using evolutionary strategies to solve the problem of the coevolution. Finally, an interesting study which compares neuroevolution and genetic programming in the same environment can be found in [1].

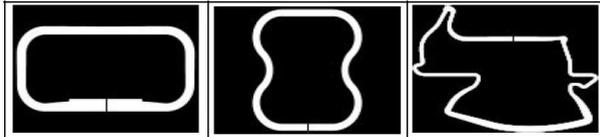
2 Problem Analysis and Design

As can be seen in the previous section, the AVG is a very difficult and widespread problem. It has been tackled from different points of view, involving, among others, robotics, artificial intelligence, computer engineering, telecommunications, signal and image processing, or control and automation techniques. Our proposal in this work is to divide the problem of the AVG starting from the optimization of a lap trajectory in a known circuit. The idea is to increase the difficulty of this scenario in future developments in order to solve the AVG problem. Thus, in this paper, we have dealt with the problem of minimizing a single lap time using Evolutionary Computation (EC) techniques, and without taking into account obstacles or consumption factors. This work will be used as a base for hereafter extensions of the goal. As we wish to reproduce human behaviour, we have designed a codification based on actions that are applicable every certain number of meters. This constant number of meters divides the circuit into segments. The actions that can be done in each segment are acceleration (value between 0 and 1, where 0 means no pressure and 1 total force) and steering (measured

with a real number between -1 and 1, where negatives values mean left turns and 0 means no turn at all). This codification encodes the individuals using 2 chromosomes: one for the acceleration value and the other for the steering. Each chromosome has the same number of genes as segments. The genes contain real values. It must be taken into account that the segment size needs to be adjusted because long segment sizes will cause shorter individuals (which is ideal for the GA) but it will offer less precision to the driver. Therefore, a trade off must be found in order to obtain a good base individual. One of the biggest problems faced when trying to optimize a car trajectory in a circuit by means of GAs is how to choose the first configuration used. This is because traditionally GA approaches evolve directly from an initially random generated configuration. As it is very difficult to obtain automatic driving evolution from individuals that have been generated by chance, we decided to develop a traditional algorithm to obtain a base trajectory which can complete a lap, although finishing it in poor times. The mechanism of this algorithm is simple, it tries, for each segment, all possible combinations of steering and acceleration (using, only for this base individual, a discretization of the allowed values with a granularity of 0,1), looking for the pair of values which maintains the vehicle closer to the center of the track. Once all segments have been evaluated we have obtained the individual taken as a basis for the GA, then the driving learning process can start. The operators used in the GA are the following:

1. Selection: a size three tournament has been implemented in all experiments.
2. Mutation: the mutation operator used is executed over each one of the genes of the final individual. Under certain mutation probability, each gene can be altered by chance so the new value will be a quantity not too far from the original.
3. Crossover: two crossover types were used in this project:
 - (a) Uniform: each one of the genes of the final individual has equal probability of coming from each one of its parents.
 - (b) By stretches: the goal of this approach is to test if a group of genes can encode a good way to, for instance, take a turn. In this case, the probability that determines the source of each gene is the same, but now these genes are exchanged in groups, instead of one by one.
4. Elitism: elitism of one individual has been used in all experiments.

The simulator used in this project to test our individuals is TORCS (The Open Racing Car Simulator), chosen because it is open source and gives us the flexibility to modify and manage the process internally. The car used in all the test is Nascar RWD (Nascar category), taking its fuel, aerodynamics and suspension configuration by default. It is also important to note that the lap time stored is from only the first lap. After the first lap the race is restarted in order to evaluate the next individual. All the experiments were conducted in three different circuits of diverse complexity, as can be seen in Figure 1:



A-Speedway	E-Track 5	Aalborg
1908,32 m Length	1.621,73 m Length	2587,54 m Length
25 m Width	20 m Width	10 m Width

Fig. 1. Circuits tested

2.1 Results with Fixed Segment Size (IT1)

In the A-Speedway circuit, the experiments were divided into 2 groups, depending on both types of crossover techniques explained before. Each experiment was executed through 1,000 generations and the segment size was fixed to 10 meters. Five experiments of each set were executed. As can be seen in Table 1, results with both crossover strategies are very similar, but the uniform case provides more stability in them. All experiments ended with lap times under 40 seconds, which is a very big improvement compared with the initial 76 seconds of the individual of reference. Most of the experiments finished with a very good tendency, and it is expected that the times keep on improving during further generations (only 1,000 were conducted). After the good results obtained in the first circuit, we tried another set of experiments on a a bit more complicated track: the E-Track5. The results were good, but the improvements found by the GA on this track (10.61 seconds on average, 14.45%) were not so good as the ones obtained for the A-Speedway (39.69 seconds on average, 50.89%). To understand the problem we observed the behaviour of the generated drivers and we concluded that the size of the segment in circuits with more turns affects the performance of the driver, because some parts of the circuit need more precision driving than others. To test the influence of this parameter we ran 2 different sets of experiments using different constant segment sizes.

As can be seen in Table 1, the results of the experiments with longer segments are quite worse than the ones with shorter stretches. In the first case, the vehicles do not have enough precision to drive through the circuit because the size of the segment. Therefore, it is complicated to obtain a good performance in the lap with large segments, even for the individual of reference. In contrast, shorter segments allow drivers to have more precision when driving through the track. Finally, we decided to see what happens in the Aalborg track. This circuit

Table 1. Results in A-Speedway and E-Track 5

Circuit	Pop. size	Mutation	Crossover	Segments	Base ind. time	GA time(mean)	σ
A-SpeedWay	100	0.005-0.01	uniform	191	76.064 secs	36.479 secs	2.99
A-SpeedWay	100	0.005-0.01	stretches	191	76.064 secs	36.369 secs	4.51
E-Track 5	100	0.005-0.01	uniform	163	129.992 secs	112.468 secs	17.871
E-Track 5	100	0.005-0.01	uniform	326	74.108 secs	63.493 secs	3.915

is the most difficult one of the three tracks tested: it is longer, it has more turns and it is much narrower than the others. The base individual has been generated using a segment size of five meters, in order to provide a high precision for the driving, and ends a lap in 288.044 seconds. The GA was configured using uniform crossover and, like the previous circuits, different combinations of mutation values were tested on several experiments. The best of them got 178.478 seconds, so a reduction of almost 2 minutes from the base time was obtained.

2.2 Evaluating the Results with Other Simulator Bots

However, the results presented so far, even reducing the base individual times, need to be validated against real drivers. As real drivers need to be trained, and can be biased, our proposal is to compare the results with the lap times performed by the simulator bots. The simulator provides several programmed bots that have been used to check whether our lap times are really competitive or not. Figure 2 shows the fitness curve of the best individuals of A-Speedway compared with simulator bots' lap times.

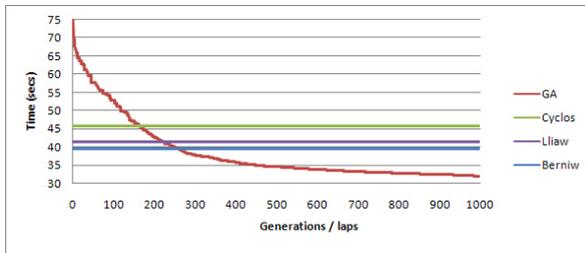


Fig. 2. A-Speedway best individual vs. bots (fitness curve)

In this figure can be clearly seen that the final lap time of our experiments is much better in A-Speedway than the best times performed by the bots. Another interesting comparison can be made by taking the lap times performed by human players, with nearly one hour of experience, in this circuit. The best human time obtained was 35.510 seconds (while ours is 31.962 seconds) so our GA individual is even better than a human driver. Analyzing E-Track 5, our lap times are worse than the bot ones, with a difference ranging between 12 and 19 seconds. As the difficulty grows, it becomes worse. For the Aalborg track, the simulator bots perform, depending on their skills, from 96.440 to 78.970 seconds, and the GA's best achieved result is 178.478 seconds. Allowing the GA to run during more generations would have decreased the lap time, but it is important to consider an improvement in the GA in order to evolve faster.

2.3 Variable Segment Size (IT2)

The results obtained in the first part of this research were promising but, in the most complex circuits, the lap time reduction could not reach the times

performed by the simulator bots. In order to improve the results and take a step forward in the problem, some modifications were proposed:

- **Variable sized segments:** the most important modification implemented is the adaptation of the individual to variable segment sizes. In the case of real driving, the number of corrections of speed and steering depends greatly on the features of the track point where the car is located on. Indeed, a higher number of actions is required when the vehicle is in a bend. The point, then, is how to find a procedure to determine the size of each segment of a circuit. The idea is to create long segments for straight parts and short ones for turns, in order to provide the individual with more precision for the trajectory. Fortunately, the TORCS simulator provides us with a segment division that accomplishes this requirement, so they were used in the new codification. Nevertheless, if we did not have this information, we could obtain these segments analyzing the curve tangent of the track, considering more segments in sharp turns (higher variations of tangent value) and less in smooth stretches (lower variation).

- **Changes on the individual of reference:** because of the changes in segment sizes, a new problem arises when the base individual is searched again. With the new segment sizes, some of them are so short that the individual interprets them as straight segments. Then, the vehicle does not need to steer to keep centered in the track, and obtains higher speed turning with an unproper speed. Several solutions were considered, but in the end we defined a new parameter in charge of setting the maximum speed that the car was allowed to reach. Therefore, the new algorithm used to find the base individual needs only to find the best steering value to keep the car centered while keeping its velocity near (and always under) the speed limit. The application of this new algorithm brought us two main advantages: (1) the execution time spent was decreased dramatically because the search space is smaller and, (2) the lap time of the individual of reference was reduced. This improvement is an outcome of the changes made which provide more precision with the vehicle and, consequently, better lap times. Table 2 shows the new values for segments and their initial lap times.

Table 2. New parameters for the genetic algorithm

Circuit	Segments (1st Iteration)	Segments (2nd Iteration)	Base Time (1st Iteration)	Base Time (2nd Iteration)
A-Speedway	384	244	76.06 secs	60.17 secs
E-Track 5	326	334	74.67 secs	68.33 secs
Aalborg	518	371	288.04 secs	158.39 secs

It is remarkable how the new changes reduce the base times in all circuits, being the reduction in the last circuit very promising: 158.39 seconds. This lap time is much lower than the time obtained by the GA with the IT1 codification (178.478 seconds).

- **Changes on the GA:** a steady state has been implemented to provide more generations. It works creating a new individual in each generation, using the usual operators (tournament selection, uniform crossover and mutation, as

were used before), and comparing it with the worst one of the population. In the case where the new individual is better than the old one, it is substituted and the algorithm continues creating a new individual.

Once the previous modifications were made, a test set was executed in order to check if better results could be obtained. All of them were executed in the third circuit (Aalborg). As Aalborg is the most complex of the circuits tried, we considered that good results on it would mean good results in the others (A-Speedway and E-Track 5). After several tests, the mutation value chosen was 0.005. All the experiments were executed through 100,000 steady state generations. The results found are summarized in Table 3. As can be seen in this table, the final lap times are competitive, and even more if we compare them with the best times obtained in the previous iteration. The improvements in relation to IT1 are measured in a difference of almost 90 seconds on average.

Table 3. Summary of 40 experiments in Aalborg (IT2)

IT2	Generations	Base time	Reduction	Final Time
Mean	1,000	158.39 secs	43.10 secs	115.28 secs
Best result	1,000	158.39 secs	59.79 secs	98.59 secs

The improvements made in relation to IT1 are obviously due to the changes done to the base individual. However, the best lap times achieved by the GA in the IT2, are very close to the ones performed by the simulator bots. The final lap time stopped at 98.592 seconds, only 2 seconds away from the lap time of one of the bots. Attending to the mean, this approach is still 32.04 seconds (on average) away from the best TORCS simulator bot.

2.4 Variable Segment Size and Gene Value Discretization (IT3)

As the results obtained after the IT2 modifications did not reach the bots lap times, we made some additional changes to the algorithm, with the aim of performing competitive lap times. The changes made in this third iteration of the paper are the following:

Initial population mutation: in this iteration, the whole initial population is mutated before starting in order to have more diversity for the steady state algorithm. This change helps the algorithm to perform better by providing a fast fitness reduction during the first generations. It is important to note that, in this mutation, only one individual remains intact (as the individual of reference) to ensure having at least one which can complete a lap.

Discretization of genes values: from the beginning of the project, the values of the genes have been continuous (real value codification). However, we can consider the relevance of the values beyond the fourth or fifth floating point number insignificant. For instance, the difference between a turn value of 0.0001 and 0.0002 cannot be appreciated in the driving process and, what it is worst,

Table 4. Summary of experiments on Aalborg (IT3)

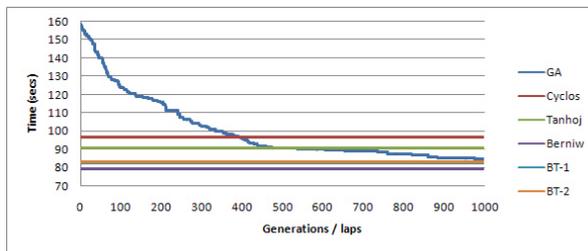
IT3	Generations	Base time	Reduction	Final Time
Mean	1,000	158.39 secs	65.05 secs	93.34 secs
Best result	1,000	158.39 secs	73.81 secs	84.57 secs

increases the search space, affecting negatively to the performance of the algorithm. In addition, a physical instrument for steering will probably not be able to work with such a precision. The problem of this approach is that, as mutation is a procedure that can change the value of a gene in any quantity, its final value will not be discrete and the search space becomes extremely large. Because of that, all the genes values have been discretized and their precision has been set to two decimal numbers.

Changes in mutation procedure: due to the previous modification, the mutation procedure has changed in order to fit in the new gene values. However, another important update has been performed in this process. All over the GA, some kind of cultural information is tracked: when an individual is mutated and the new lap time is better than the one of its parents, the values of these successful mutations are stored in a special individual. The information kept here is used as a reinforcement for the next mutation chances, performing the gene modification oriented to the successful mutations of its neighbors. This change is based on the fact that contiguous segments usually perform similar actions in this problem.

Once the changes were made, another test set was executed. The results, which are gathered in Table 4, are significantly better, obtaining competitive lap times on average, and with a best result of 84.578 seconds. This new algorithm obtains a time reduction of 22 seconds on average from the times achieved by the base individual, and its standard deviation is also lower. Comparing our times with the simulator bots, the results obtained are, on average, between the worst and the best simulator bot lap times, so the objective of achieving competitive lap times has been reached. The evolution of the best individual achieved by the IT3 GA compared to the bot lap times can be seen in Figure 3.

As can be seen, fitness value decreases smoothly from the base individual lap time, achieving competitive values before the half point of the GA execution.

**Fig. 3.** Aalborg best individual vs. simulator bots (fitness curve)

This behaviour is common in most of the experiments of this test set. Furthermore, in the last generations, the decreasing of the lap time continues, which indicates that the results will be even better with longer executions. Moreover, results on average also improve part of the times obtained by the simulator bots.

3 Conclusions and Future Work

The use of GAs has been proved to be a successful technique for the optimization of a lap trajectory in a known circuit. In the first experiments performed in this paper, the results in complex circuits showed us that we were very far away from competitive lap times. However, by analyzing and studying the algorithm, some changes were made to the GA. With these changes the GA became good enough to obtain lap times comparable with the ones achieved with the TORCS simulator bots. This was tested by means of experiment sets with a significant number of runs. The relevance of these successful results is not the fact of beating the bots of the simulator. The relevance is the discovery, in the process of doing so, of each one of the features which make this problem difficult and interesting to solve. One of the problems of this approach is that the trajectory obtained in the evolving process depends highly on the initial state of the car. If noise in sensor readings or a change in the initial position happens, the trajectory evolved would not be valid. The same happens if we need to perform more than one lap. Avoiding these problems is one of the main objectives for future developments. Two branches of future work can be the basis of our new research lines. On one hand, applying different techniques, such as evolutionary strategies and/or multi-objective optimization. These techniques can be used to find out new possibilities for solving this problem in order to obtain better lap times than any bot or human player. On the other hand, we can add complexity to this challenge; for instance, we can try to include fuel consumption, gear changing, overtaking other vehicles and, why not, a whole race planning.

References

1. Agapitos, A., Togelius, J., Lucas, S.: Evolving controllers for simulated car racing using object oriented genetic programming. In: *Procs. Conference on Genetic and Evolutionary Computation*, pp. 1543–1550. ACM, New York (2007)
2. Baluja, S., Caruana, R.: Removing the genetics from the standard genetic algorithm. In: *Prieditis, A., Russel, S. (eds.) The Int. Conf. on Machine Learning 1995*, San Mateo, CA, pp. 38–46. Morgan Kaufmann Publishers, San Francisco (1995)
3. Baluja, S., Sukthankar, R., Hancock, J.: *Prototyping intelligent vehicle modules using evolutionary algorithms* (1998)
4. Bernard, J., Gruening, J., Hoffmeister, K.: Evaluation of vehicle/driver performance using genetic algorithms, society of automotive engineers
5. Bertozzi, M., Broggi, A., Conte, G., Fascioli, A.: The experience of the argo autonomous vehicle. In: *Procs. Spie 1998* (1998)
6. Chen, Q., Ozguner, U., Redmill, K.: Ohio state university at the 2004 darpa grand challenge: Developing a completely autonomous vehicle. *IEEE Intelligent Systems* 19(5), 8–11 (2004)

7. Collado, J.M., Hilario, C., de la Escalera, A., Armingol, J.M.: Self-calibration of an on-board stereo-vision system for driver assistance systems. In: Intelligent Vehicles Symposium, pp. 156–162 (2006)
8. Cambiaghi, D., Vetturi, D., Gadola, M., Manzo, L.: Semi-active strategies for racing car suspension control, paper no. 962553, university of brescia
9. Hacker, K., Kasprzak, E.M., Lewis, K.: Exploring the design tradeoffs and computational savings of executing vehicle simulations in a parallel computing environment. In: ASME Design Automation Conference (2000)
10. Floreano, D., Kato, T., Marocco, D., Sauser, E.: Coevolution of active vision and feature selection. *Biological Cybernetics* 90, 218–228 (2004)
11. Hemmi, H., Tanev, I., Joachimczak, M., Shimohara, K.: Evolution of the driving styles of anticipatory agent remotely operating a scaled model of racing car. In: *Procs. of the IEEE Congress on Evolutionary Computation*, Edinburgh, 2–4 September 2005, vol. 2, pp. 1891–1898. IEEE, Los Alamitos (2005)
12. Niehaus, A., Stengel, R.F.: Probability-based decision making for automated highway driving. In: *Vehicle Navigation and Information Systems Conference*, vol. 2, pp. 1125–1136 (1991)
13. Pomerleau, D., Jochem, T.: Rapidly adapting machine vision for automated vehicle steering. *IEEE Expert: Special Issue on Intelligent System and their Applications* 11(2), 19–27 (1996); see also *IEEE Intelligent Systems*
14. Pyeatt, L., Howe, A., Anderson, C.: Learning coordinated behaviors for control of a simulated robot (1996)
15. Siegle, G., Geisler, J., Laubenstein, F., Nagel, H.-H., Struck, G.: Autonomous driving on a road network. In: *Procs. of the Intelligent Vehicles 1992 Symposium*, Res. Center Commun., Robert Bosch GmbH, Hildesheim, pp. 403–408 (1992)
16. Sukthankar, R., Baluja, S., Hancock, J., Pomerleau, D., Thorpe, C.: Adaptive intelligent vehicle modules for tactical driving (1996)
17. Sukthankar, R., Pomerleau, D., Thorpe, C.: Shiva: Simulated highways for intelligent vehicle algorithms (1995)
18. Sun, Z., Bebis, G., Miller, R.: On-road vehicle detection using evolutionary gabor filter optimization (2005)
19. Tanev, I., Joachimczak, M., Shimohara, K.: Evolution of driving agent, remotely operating a scale model of a car with obstacle avoidance capabilities. In: *Procs. of the Conference on Genetic and Evolutionary Computation*, pp. 1785–1792. ACM, New York (2006)
20. Thorpe, C., Jochem, T., Pomerleau, D.: Automated highway and the free agent demonstration (1998)
21. Togelius, J., Lucas, S.: Arms races and car races. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) *PPSN 2006*. LNCS, vol. 4193, pp. 613–622. Springer, Heidelberg (2006)
22. Togelius, J., Lucas, S.M.: Evolving robust and specialized car racing skills. In: Yen, G.G., et al. (eds.) *Procs. IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, July 16–21, 2006, pp. 1187–1194. IEEE Press, Los Alamitos (2006)
23. Togelius, J., Simon, M.: Evolving controllers for simulated car racing. In: *Procs. IEEE Congress on Evolutionary Computation* (2005)
24. Wloch, K., Bentley, P.J.: Optimising the performance of a formula one car using a genetic algorithm. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiño, P., Kabán, A., Schwefel, H.-P. (eds.) *PPSN 2004*. LNCS, vol. 3242, pp. 702–711. Springer, Heidelberg (2004)