A Coevolutionary Algorithm for Tyre Model Parameters Identification

[1]Cabrera, J.A.; [1]Ortiz, A.; [2]Estebanez, B.; [1]Nadal, F. and [1]Simon, A.

[1]*Mechanical Engineering Department, Malaga University*

C/ Pedro Ortiz Ramos s/n 29071 Malaga- SPAIN

[2]*System Engineering and Automation Department, Malaga University*

C/ Pedro Ortiz Ramos s/n 29071 Malaga- SPAIN

Abstract

The problem of tyre model coefficients identification using minimum test data is studied in this work. To obtain these tyre model parameters, an intense research effort by the automotive community has been made and there are different methods to fit the values of these parameters. This problem is addressed in this work through a coevolutionary algorithm that interactively searches the optimum tyre model parameters and new test data in disagreement with the tyre model. The algorithm is composed of two stages: the estimation phase, which finds out the tyre model parameters which can predict actual tyre test data, and the exploration phase, which finds out new test data which have the most disagreement with the response of the current model. The feasibility of the methodology is demonstrated comparing the obtained results with other known techniques.

Introduction

Genetic algorithms (GAs) are examples of search procedures that use a computer and actions based on the mechanics of natural genetics and selection to produce ‘simulated evolution’. In this way, GAs try to copy biological evolution using genetic operators which are a mathematical idealization of biological processes of natural selection.

Genetic algorithms (GAs) have demonstrated their efficiency in search and optimization problems in many application domains. Many implementations exist which use these kinds of algorithms and huge engineering problems have been solved using GAs with different topologies, function objectives and so on, but they often have some problems like premature convergence, stagnation and others. To solve these problems many researchers have developed new strategies based on GAs. One of these is called coevolutionary algorithms (CAs). CAs are searching procedures, also based on genetic and natural selection, in which mutual interactions among individuals of one or more different populations happen. These reciprocal actions yield the evolution of individuals to better ones. Due to that, coevolution is used as a powerful tool for optimization of multidimensional problems with multiple local minima. We make use of these features of CAs to solve a problem of parameter identification in tyre models. The most common tyre models are based on an empirical formula with many parameters which are fitted to test data, so an optimization method is necessary. To carry this out, we use a coevolutionary algorithm which consists of two phases in this approach. These two phases are: the estimation phase of the tyre model parameters which are predicted by the tyre test data, and the exploration phase, which searches tyre test data which increase the information level of tyre behaviour. We also compare the obtained results with the coevolutionary algorithms proposed in this work with two other existing methods in bibliography and a statistical study will be made to obtain a conclusion of the proposed method.

Background

The use of coevolution was introduced by [1],[2], as an optimization procedure on sorting networks in 1991. Since then many different versions have been carried out. In this approach, both a population of sorting networks represented the ‘host’ and other test cases represented the ‘parasite’ which evolved using coevolution.

All those variants were grouped in two broad classes in [3]. However, it was mentioned that it was difficult to make a general classification due to conflicting schemes, inconsistent terminology and specific procedures created ‘ad hoc’ to solve every problem. Nevertheless, many researchers classified all types of coevolution relationships in three classes: competitive (predator-prey), cooperative (symbiosis) and non-symmetric systems (host-parasite).

Coevolutionary strategies have some drawbacks due to their complex coevolutionary dynamics. Many researchers [4,5,6,7,8,9] have focused their investigations on this subject to call attention to these pathologies. In literature, the most common inconvenience we found for proper coevolutionary performance is the red queen effect, disengagement, cycling, defocusing and intransitive superiority. To understand them, it is necessary to know the distinction between the notions of objective fitness versus subjective fitness. Objective fitness is the absolute fitness metric used in classical evolutionary algorithms and subjective fitness is the fitness measured in each population in the coevolutionary strategy and sometimes it is not correlated with the objective fitness. A coevolving individual only knows its subjective fitness and it cannot know if the absolute fitness is improving, so there is 'disengagement' between these two concepts in coevolutionary systems.

One of the coevolution schemes is based on a metaphor of intelligent reasoning proposed in [7]. This author proposed an estimation and exploration algorithm (EEA). In this case, the EEA is applied to a modelling problem, i.e., the algorithm searches the best mathematical model that satisfies an output set for a given input set of a physical model. However, we apply the EEA to an optimization problem in which the algorithm finds the input that makes the known tyre model fit a known output set with minimal error.

Other authors use coevolutionary strategies to solve optimization problems. For example, in [10] two populations are used. The first one is to minimize the objective function regardless of constraints, and the second population is to minimize the violation of constraints regardless of the objective function. Also, in [11], the authors solve a problem of structural damage identification using a coevolutionary algorithm that interactively searches for damage scenarios and optimum physical tests. A coevolutionary differential evolution is used to solve constrained optimization problems in [12-13].

On the other hand, tyre models are used to calculate the tyre forces and moments as responses to the wheel motion that may be given in terms of various slip quantities. We can distinguish theoretical models based on the physics of the tyre construction [14-15], and empirical or semi-empirical models which are solely based on experimental results [16-18]. Also, combinations of both approaches are used in the development of the tyre model. A widely used empirical tyre model is based on the so-called Magic Formula [16].

The importance of tyre models is well-known due to the fact that the contact between the vehicle and the ground is by means of the tyres. Tyre models predict the forces generated by carcass deformation and rubber tread displacements. To achieve a proper performance of the efforts in the contact patch, some of the most popular tyre models [16, 19] need to tune their parameters. To obtain these tyre model parameters, an intense research effort by the automotive community is required so there are different methods to fit the values of these parameters. An extended method to get the Magic Formula parameters is used in MF-Tool software. This method is based on a starting values optimization technique (SVO) [19]. Basically, the algorithm needs very close starting parameter values to obtain a good result. Hence, users of this kind of algorithm have to know the physical properties of the contact between the tyre and the road. Another drawback of the SVO is the time needed to calculate the suitable parameters and the percentage of success in getting a global minimum.

These search methods, as mentioned in [20], seek local optima by hopping on the function and moving in a direction related to the local gradient. This is simply the motion of hill climbing to find the local best, climb the function in the steepest permissible direction. While SVO techniques have been improved, extended, hashed and rehashed, some simple reasoning shows their lack of robustness. First, these methods are local in scope, the optima they seek are the best in a neighbourhood of the current point. Second, SVO methods depend on the existence of derivates. Even if we allow numerical approximation of derivates, this is a severe shortcoming.

Another approach, using evolutionary algorithms, is described in [21-22] and it is applied and evaluated in [23]. They named their algorithm IOA. The IOA algorithm is based on Differential Evolution (DE) introduced by [24]. In these approaches the authors obtain better results than the ones obtained by SVO techniques. However, to improve important features in the optimization process, the optimization using co-evolution has been carried out.

In this paper we use an estimation phase to find out the tyre model parameters which can predict actual tyre test data, and an exploration phase to find out new test data which have the most disagreement with the response of the current model.

The objective of this article is to illustrate several advantages of a proposed co-evolutionary approach applied to the problem of identification of tyre parameters. Among these advantages are: the decrease of computation time, a lower tyre test data number is needed to evolve every generation of tyre model parameters. Also, if we assume that traits are heritable, tyre test data passed on by parents will be passed on by offspring and then these test data may be omitted. Hence, a higher level of efficiency will be reached.

Formulation of the problem

The problem consists of obtaining the parameters of a known tyre model, the Magic Formula [16]. This model represents the characteristics of the tyre by means of mathematical expresions, which have been calculated for pure and combined slip conditions. The parameters which are the objetive of the optimization method are defined in the Magic Formula, where the goal function will be obtained comparing the Magic Formula results with the test data for three cases: pure longitudinal force, pure lateral force and auto aligning torque.

The general Magic Formula in pure slip conditions is expressed as:

$$\delta(x) = D \cdot \sin\left[C \cdot \arctan\left\{B \cdot x - E \cdot \left(B \cdot x - \arctan\left(B \cdot x\right)\right)\right\}\right] \tag{1}$$

where:

$$\begin{aligned} \Omega_{pure}(X) &= \delta(x) + S_v \\ x &= X + S_h \end{aligned} \tag{2}$$

B,*C*,*D* and *E* are functions of the parameters that will be optimized and they are defined in [16][18]. where Ω_{pure} is the output variable and X is the input variable. Finally, S_v and S_k are also defined in [16][18]. To obtain the goal function in the optimization method, we compute the square difference sum (mean squared quadratic error) between the Magic Formula output and the test data. In the following sections the particular cases of this study are shown.

Longitudinal Force:

The output variable is $\Omega_{pure}(X) = F_X^{Pacejka}\left(X\right)$, which represents the braking and traction force and the input variable is *X*, which represents longitudinal slip *κ*. The parameters and the goal function are shown in the following equation:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{m}\left[F_X^{Pacejka}\left(WX, \kappa_i, Fz_j\right) - F_X^{measured}\left(\kappa_i, Fz_j\right)\right]^2$$

$$WX = \left\{\begin{matrix} PCX1, \\ PDX1, PDX2, \\ PKX1, PKX2, PKX3, \\ PEX1, PEX2, PEX3, PEX4, \\ PHX1, PHX2, \\ PVX1, PVX2 \end{matrix}\right\} \tag{3}$$

n is the number of test points and *m* is the number of load cases (F_Z) where the longitudinal force is evaluated in the test procedure and *WX* are the parameters for longitudinal force defined in [16][18] function of which are expressed the parameters *B*, *C*, *D* and *E* that appear in (1).

The objective is reduced to search the *WX* parameters that define the $F_X^{Pacejka}\left(WX, \kappa_i, Fz_j\right)$ function and minimize the previous goal function in pure longitudinal slip conditions.

Lateral Force:

The lateral force is $\Omega_{pure}(X) = F_Y^{Pacejka}\left(X\right)$ and the input variable is *X*, which represents the slip angle α. The parameters and the goal function are shown in the following equation:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{g=1}^{l}\left[F_Y^{Pacejka}\left(WY, \alpha_i, Fz_j, \gamma_g\right) - F_Y^{measured}\left(\alpha_i, Fz_j, \gamma_g\right)\right]^2$$

$$WY = \left\{\begin{matrix} PCY1, \\ PDY1, PDY2, PDY3 \\ PKY1, PKY2, PKY3, \\ PEY1, PEY2, PEY3, PEY4, \\ PHY1, PHY2, PHY3 \\ PVY1, PVY2, PVY3, PVY4 \end{matrix}\right\} \tag{4}$$

n is the number of test points and *m* is the number of load cases(F_Z) and *l* are the numbers of camber angles (γ) where the lateral force is evaluated in the test procedure and *WY* are the parameters for lateral force defined in [16][18] function of which are expressed the parameters *B*, *C*, *D* and *E* that appear in (1).

The objective is reduced to search the *WY* parameters that define the $F_Y^{Pacejka}\left(WY,\alpha_i,Fz_j,\gamma_g\right)$ function and minimize the previous goal function in pure side slip conditions.

Auto Aligning Torque:

The Auto aligning torque M_z is expressed as:

$$M_Z^{Pacejka}\left(X\right)=-tF_Y^{Pacejka}\left(X\right)+M_{zr} \tag{5}$$

The input variable is *X*, which represents slip angle α, and *t* y M_{zr} are defined in [16][18]. Finally, the goal function and the parameters are:

$$\min\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{g=1}^{l}\left[M_Z^{Pacejka}\left(WZ,\alpha_i,Fz_j,\gamma_g\right)-M_Z^{measured}\left(\alpha_i,Fz_j,\gamma_g\right)\right]^2$$

$$WZ=\begin{Bmatrix}PCY1,\\ PDY1,PDY2,PDY3,\\ PKY1,PKY2,PKY3,\\ PEY1,PEY2,PEY3,PEY4,\\ PHY1,PHY2,PHY3\\ PVY1,PVY2,PVY3,PVY4\\ QBZ1,QBZ2,QBZ3,GBZ4,QBZ5\\ QCZ1\\ QDZ1,DDZ2,QDZ3,QDZ4\\ QEZ1,QEZ2,QEZ3,QEZ4,QEZ5\\ QHZ1,QHZ2,QHZ3,QHZ4\\ QBZ9,QBZ10\\ QDZ6,QDZ7,QDZ8,QDZ9\end{Bmatrix} \tag{6}$$

n is the number of test points and *m* is the number of load cases (F_Z) and *l* is the number of camber angles (γ) where the aligning torque (pure side slip) is evaluated in the test procedure and *WZ* are the parameters for the aligning torque function of which are expressed the parameters *B*, *C*, *D* and *E* that appear in (1).

The objective is reduced to search the *WZ* parameters that define the $M_Z^{Pacejka}\left(WZ,\alpha_i,Fz_j,\gamma_g\right)$ function and minimize the previous goal function in pure side slip conditions. In this case, we only need to look for the Q*** type parameters. The P*** type parameters are the same as those for lateral force.

A coevolutionary algorithm in tyre parameter identification

In this work a coevolutionary algorithm has been developed with the aim to save the drawbacks in the previous approaches.

The coevolution algorithm is an advanced technique to solve an optimization problem breaking it in parts which are solved along an evolution in series. Every part is a population which reciprocally interacts by means of a combined fitness function. This is a representative scheme to carry out the competitive coevolution, where both populations compete to build a solution to the problem. Our coevolution algorithm, Malaga University Co-evolution Algorithm (MUCA), has two phases, exploration and estimation phases, as we can see in Figure 1.

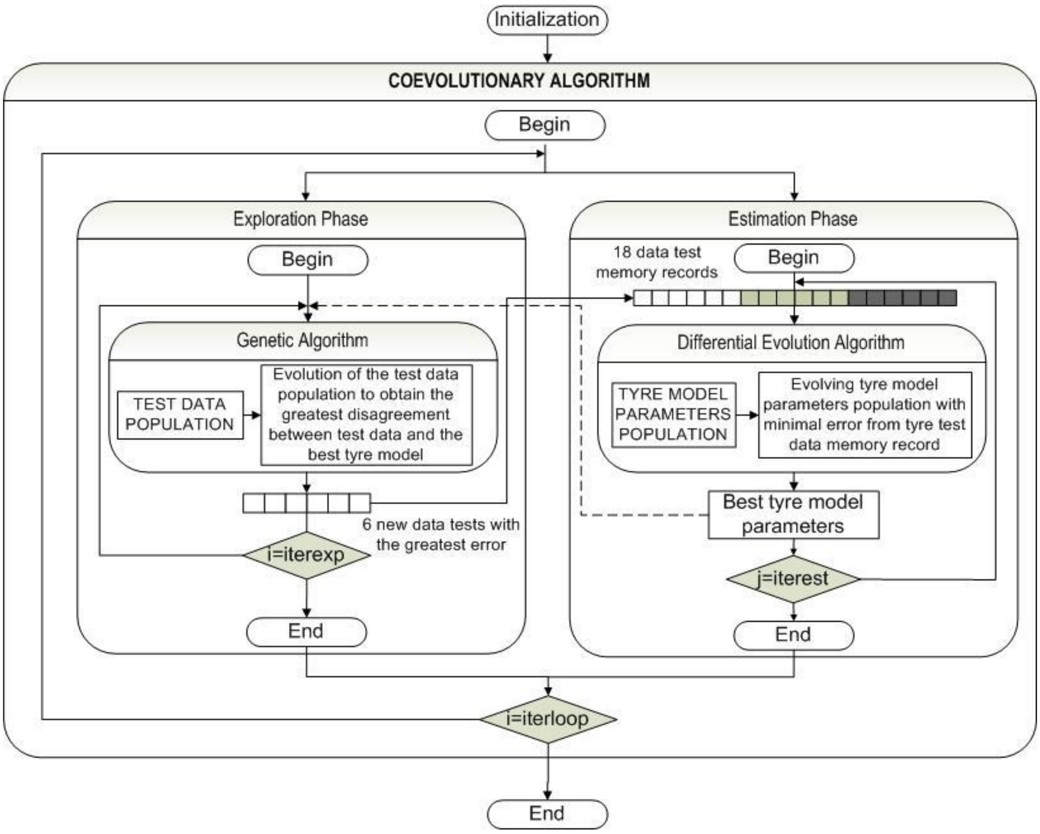

Figure 1.-Malaga University Co-evolution Algorithm (MUCA) scheme.

In the exploration phase, a test data population is evolved to search the greatest disagreement with the best tyre model parameter prediction and in the estimation phase, a tyre model parameter population is evolved to match with the test data found in the previous phase.

There is an initial stage where the variable starting values are given, i.e., tyre model parameters and test data populations are initialized randomly. The estimation population is a matrix with *NP (number of individuals)* and *D (number of genes)*, whereas the exploration population has *NX* individuals and *DX* genes.

Two different evolutionary strategies are used in each phase.

To evolve the test data population in the exploration phase, a genetic algorithm is used. The test data population is initialized randomly with *NX* individuals with *DX* genes. Each gene represents real test data obtained under the same tyre test conditions, as we can see in Figure 2. Therefore, we can see how a significant reduction of the test data is stored in each individual. In our work, we only store six data test points in each individual in the exploration phase, *DX*=6.

An elitist selection operator is used to obtain the couple for reproduction. The algorithm evolves the populations until the maximum number of iterations is exceeded and the best individual, which represents the tyre test data with the greatest error, is used in the estimation phase. The error is the squared difference between the six real data contained in each individual and the evaluated data for the best model obtained in the estimation phase. Notice that, to clarify the figure, the best model curve does not appear in Figure 2.

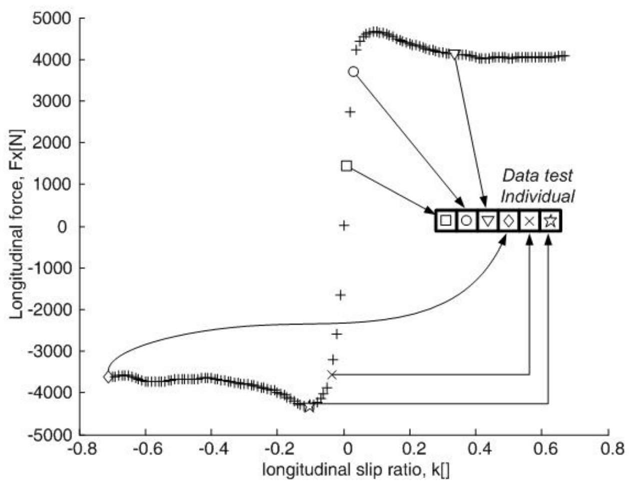

Figure 2.- Test data chromosome representation

In the exploration phase, the goal function appears in equation (7):

$$\max \sum_{i}^{DX} \sum_{j}^{m} \left[F_X^{Pacejka}\left(WX, \kappa_i, Fz_j\right) - F_X^{measured}\left(\kappa_i, Fz_j\right) \right]^2 \tag{7}$$

DX is the number of test points and *m* is the number of load cases (F_Z) where the longitudinal force is evaluated in the test procedure and *WX* are the parameters for longitudinal force defined in [16][18].

To evolve the tyre model parameters population in the estimation phase a Differential Evolution Algorithm [24] is used. The tyre model parameters population is initialized randomly with *NP* individuals with *D* genes. Each gene represents a tyre model and it is evolved using equation (3), (4), (5) and (6). To improve the performance and to avoid the drawbacks of coevolutionary strategies, a test data memory record with three test data generations is used. The goal function appears in equation (8):

$$\min \sum_{i=1}^{18} \sum_{j}^{m} \left[F_X^{Pacejka}\left(WX, \kappa_i, Fz_j\right) - F_X^{measured}\left(\kappa_i, Fz_j\right) \right]^2 \tag{8}$$

i=18 is the number of test points stored in the memory record and *m* is the number of load cases (F_Z) where the longitudinal force is evaluated in the test procedure and *WX* are the parameters for longitudinal force defined in [16][18].

In the co-evolution of both populations, between tyre model parameters and tyre test data, disengagement can exist, as we can observe in Figure 3. This pathology has been avoided with the following strategy in our coevolution process.

If the tyre model parameters are optimized using few tyre test data (i.e: six for every vertical load), the information level provided by the test data is too far from real tyre behaviour, so it is too easy for the tyre model parameters to explain them and all obtained models get an equally low subjective error in the estimation phase (Figure 3a). However, they are evolving to simulate an incomplete set of tyre properties, so the objective error keeps the same magnitude or even increases and does not improve with the iterations (Figure 3b). In the situation presented before, notice how the objective fitness does not present a monotonic decrease with the iteration increase. This situation produces disengagement between populations because the subjective error is calculated with few scattered tyre test data (six points) in the estimation phase and the objective error is calculated with the whole tyre test data, but the goal of our algorithm is to improve the objective error although it does not use the whole tyre test data in the exploration phase.

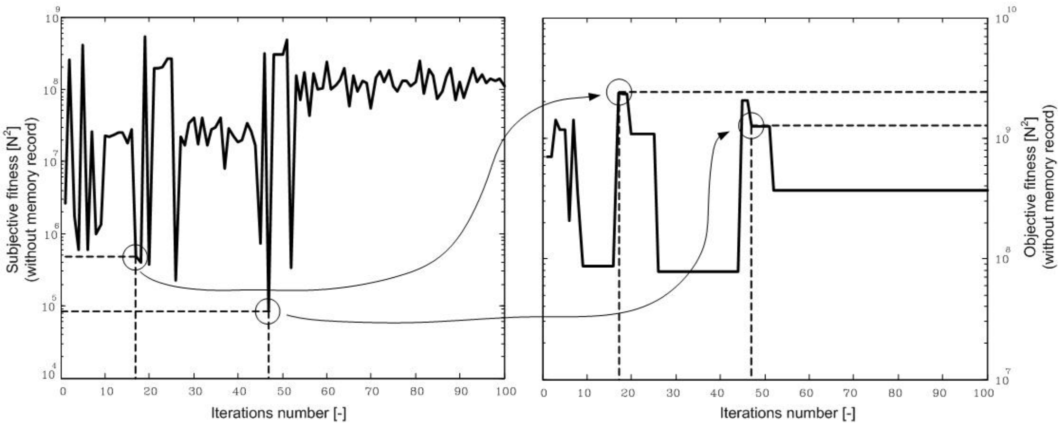

Figure 3.- Subjective and objective fitness without memory data record.

To avoid this, we make our algorithm have ‘memory’, that is, our algorithm remembers what happened before in the estimation phase for each new iteration. This way a double goal is obtained. On one hand, few tyre test data for every generation loop are used and then the time spent on the optimization process is less than the time spent on a higher number of points. And on the other side, the algorithm uses the information of previous generations and then tyre model parameters evolve to simulate a complete set of tyre properties, avoiding the disengagement between populations.

In this second situation, with the memory record, objective and subjective errors are drawn in Figure 4. Now the objective error shows a decreasing tendency with iterations, due to the fact that disengagement between populations is avoided.

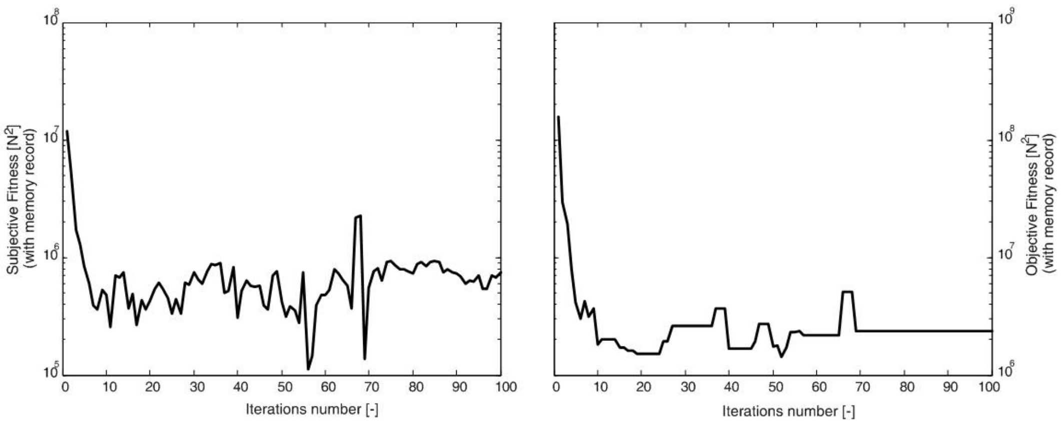

Figure 4.- Subjective and objective fitness with memory data record.

Then the main contribution of our co-evolution algorithm is that we use few tyre test data for every generation loop. This fact is possible because we use a coevolutionary strategy. In each iteration, data compete with the tyre model and vice versa. But if we want to use a simple genetic algorithm we need the whole data set to optimize tyre model parameters.

Note that, with few test data, these could not represent all the characteristics of the tyre model. The GA is not able to achieve success. In other words, with the GA, we need the whole test data set (see Figure 2). Then one of the main advantages of the MUCA is that we improve the computing time substantially because of the lower number of comparisons between test data and the real tyre model.

Hence, the MUCA obtains very accurate tyre model parameters, using the whole tyre test data, but selecting these, for every iteration, in portions of tyre test data.

Of course, all the optimization methods mentioned in this paper get the appropriate tyre model parameters but with different levels of time, cost and number of successes when the optimization is carried out. We can see all these facts in the results paragraph.

Results

This section analyses a set of results found by applying our coevolution algorithm (MUCA) developed in the previous paragraph. All examples have been programmed in a Pentium(R) D CPU 3.2 GHz and implemented in Matlab.

A coevolutionary technique has been used to optimise the parameters of the Magic Formula (MF) Tyre model Delft’96 version. To demonstrate the behaviour of the algorithm presented in this paper, the longitudinal and lateral force parameters and the auto-aligning moment for pure slip have been obtained.

The tyre model parameters have been optimized with the following tyre test data units: length (m), force (N), angle (rad), mass (kg), time (s) and slip κ(-).

The results below are obtained with a set of predefined parameters for the Differential Evolution (DE) algorithm. A discussion about the effects of these parameters in DE is shown in [25] and [26]. In the exploration phase, a Genetic

Algorithm is used with predefined parameters. The way to obtain these parameters has been carried out by means of a trial-error process.

Longitudinal force

Here, the results for pure longitudinal force are presented:

(1) For the Magic Formula Delft Tyre 96 version, the longitudinal force parameters optimised by the MUCA with the lowest error out of the 150 repetitions for the statistical study are presented in Table 1.

Table 1.- Longitudinal force coefficients in the three studied cases.

Longitudinal coefficients	**MUCA**	**SVO**	**IOA**
PCX1: C_{Fx} shape factor for longitudinal force	1.09038275	1.4146	1.39708965
PDX1: μ_x longitudinal friction at F_{znom}	-0.23049922	1.1264	1.10206790
PDX2: μ_x friction variation with load	40.07668312	-0.1817	-0.18524061
PEX1: E_{Fx} longitudinal curvature at F_{znom}	-2.321461139	0.001242	-0.45925516
PEX2: E_{Fx} curvature variation with load	-0.785631636	0.004268	-1.49950140
PEX3: E_{Fx} curvature variation with squared load	-0.228543703	0.005999	-2.46964541
PEX4: Factor in E_{Fx} curvature while driving	0.221676935	314.36	-0.90674124
PKX1: K_{Fx}/F_z longitudinal slip stiffness at F_{znom}	0.755096144	36.12	38.50310903
PKX2: K_{Fx}/F_z slip stiffness variation with load	-0.885545385	11.489	2.03196267
PKX3: Exponent in K_{Fx}/F_z slip stiffness with load	-0.002165392	0.1678	-0.59108577
PHX1: S_{hx} horizontal shift at F_{znom}	0.003768425	-0.0017105	-0.00227143
PHX2: S_{hx} shift variation with load	0.048499058	0.0030235	0.00193554
PVX1: S_{vx}/F_z vertical shift at F_{znom}	-0.069417660	0.06054	0.05759227
PVX2: S_{vx}/F_z shift variation with load	1.395445622	-0.03904	-0.02874956

(2) The adjustment that the MF for pure longitudinal force has to the tyre test data if the MUCA, SVO and IOA are used to optimise the parameters is presented in Figure 5. These tyre model curves are obtained with the longitudinal force parameters presented in Table 1. Note that in Figure 5, there are not 139 test points represented for every vertical load, with the objective to clarify the figure. We can also see how the MUCA and IOA show very similar behaviour close to the tyre test data and the SVO has a worse adjustment to the test data, especially with the vertical load of 4000 N. For this reason, the error shown in Table 2 is more than three times bigger in the SVO than in the MUCA and IOA.

(3) The sum-squared error between the test data and the longitudinal force curve, which correspond to the parameters obtained with the SVO method, with the IOA with constant control parameters and the MUCA is shown in Table 2. Also, the computing time is shown in Table 2.

(4) The algorithm parameters are: for the estimation phase, Differential Evolution Algorithm: *NP(number of individuals)*=50, *D(number of genes)*=14, *F(perturbing factor)*=0.6, *CR(crossover probability)*=0.4, *MR(mutation probability)*=0.1 and *iterest*=100. For the exploration phase *NX(number of individuals)*=5, *DX(number of genes)*=6 *CRX(crossover probability)*= 0.6, *MRX(mutation probability)*=0.1 and *iterexp*=100.

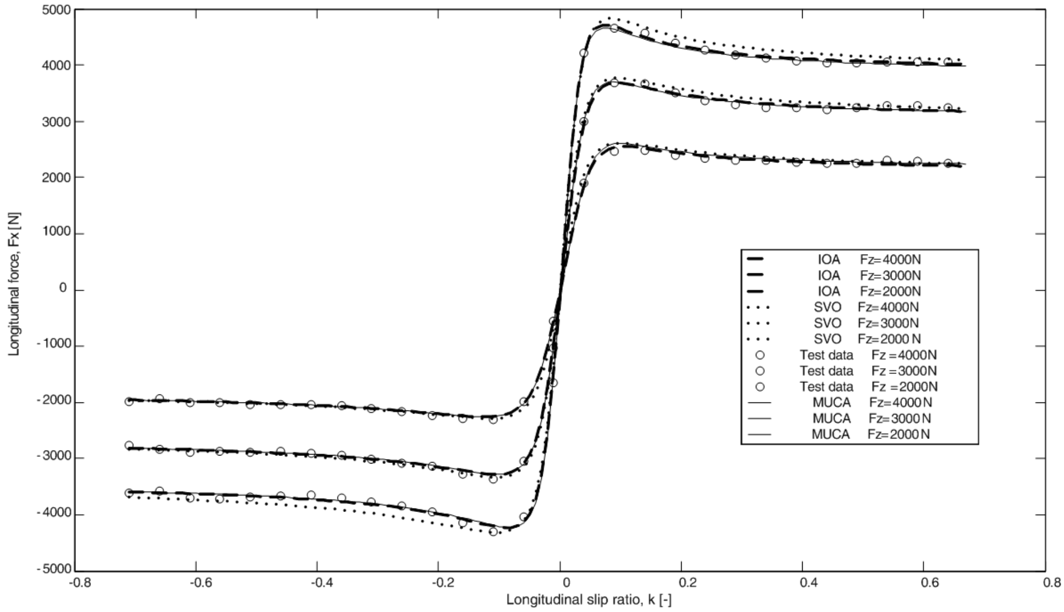

Figure 5.-Longitudinal force curves in the three studied cases.

Table 2.- Sum-squared error and computing time from the longitudinal force curve to the longitudinal force test data.

$\sum_{i=1}^{n}\sum_{j=1}^{m}\left[F_{x0}^{Opt.Meth}-F_{x0}^{measured}\right]^2$	Sum squared error [N^2]		
	MUCA	**SVO**	**IOA**
Fz=[4000,3000,2000] N $-0.71 \leq \kappa \leq 0.66$ n = 139 data points m = 3 load cases	1.2949438 x 10^6	4.266528 x 10^6	1.16948126 x 10^6
Computing time	23 s.		3 min 51 s

To investigate the performance of the algorithms, we made a statistical study. In this case, we carried out 150 different runs of the whole algorithm with the following conditions: the main loop of the algorithm was run 15 times in each run, i.e., iterloop=15 (see Figure 1). The exploration and estimation phases loop was run 100 times each one, i.e, iterest=100 and iterexp=100 (see Figure 1). Each time the main loop of the algorithm was run (a global iteration), a longitudinal force error was recorded. Figure 6 shows a box plot of 15 global iteration for 150 runs. The superior line of the box is the upper quartile error value, the inferior line of the box is the lower quartile error value and the line inside the box is the median error and the rest of the data are represented outside these boxes. We can observe how the median of the error data is decreasing with the global iterations, obtaining the best result in the last iterations. So, we can see how the evolution of the algorithm improves the result in the initial global iterations. We can also observe how the box is narrower for each global iteration. This is a measure indicating that dispersion of longitudinal force error is improving with the iterations of the algorithms. And it is an indicator of how the objective fitness decreases for each global iteration.

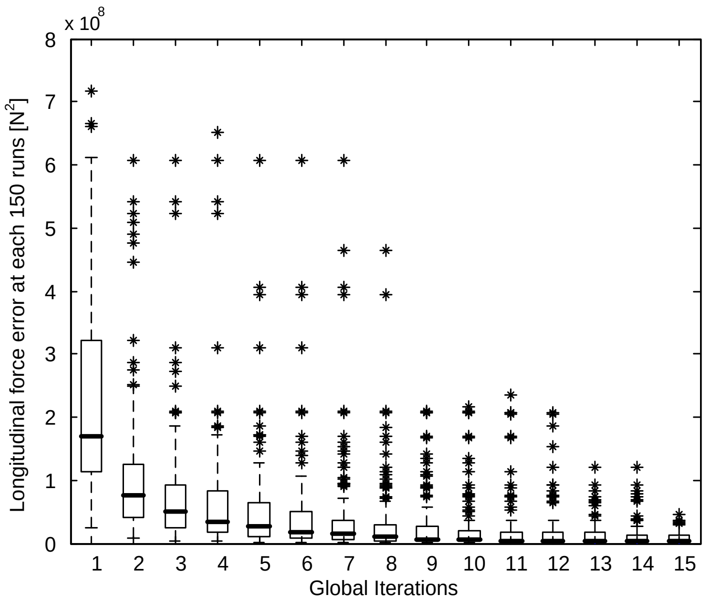

Figure 6.-Statistical parameters at 150 runs of the main algorithms. Longitudinal force error

To compare the statistical results of other methods, we also ran the IOA algorithm 150 times with 1000 global iterations and we obtained the following results that you can see in Table 3.

The Starting Value Optimization techniques (SVO) are a generic name to refer to the methods that use a starting value to initialize the optimization process. We do not dispose of the specific software created for identifying the tyre model parameters and, consequently, the statistical results in the SVO column are not available. However, the comparison with the other optimization approaches

(MUCA, IOA) will never be carried with the same initial conditions, as SVO needs a starting value of the privileged parameters (close to the optimum) and MUCA and IOA start the optimization process with whatever random values. Besides, if the SVO does not start with the specific values mentioned above, it never reaches the optimum, so the evaluations number of the goal function will be infinite or the obtained parameters are very far from the optimum.

Table 3.-Statistical comparisons among several methods.

Longitudinal force error	MUCA	SVO	IOA
Standard deviation	$8.3665x10^6$		$4.9318x10^6$
Mean	$6.6564x10^6$		$2.25606x10^6$
Best	1.2949438×10^6	4.266528×10^6	1.16948126×10^6
Worst	$4.5620x10^7$		$3.638475x10^7$
Evaluations number	4185000		41700000
Computing time	23 s.		3 min 51 s

Lateral force

Here, the results for pure lateral force are presented:

(1) For the Magic Formula Delft Tyre 96 version, the lateral force parameters optimised by the MUCA with the lowest error out of the 150 repetitions for the statistical study are presented in Table 4.

Table 4.- Lateral force coefficients in the three studied cases.

Lateral coefficients	MUCA	SVO	IOA
PCY1: C_{FY} shape factor for lateral force	1.277570645	1.2664	1.276760
PDY1: μ_y lateral friction	0.937463599	0.949	0.932775
PDY2: μ_y friction variation with load	-0.134909741	-0.13463	-0.128085
PDY3: μ_y friction variation with square camber	1.994400441	0.8261	1.019803
PEY1: Lateral E_{FY} curvature at F_{znorm}	-1.469781510	-1.3159	-1.399340
PEY2: E_{FY} curvature variation with load	-0.423934500	-0.4004	-0.074863
PEY3: Zero order camber dependency of E_{FY} curvature	0.1020335097	0.1824	0.178860
PEY4: E_{FY} curvature variation with camber	-7.571206203	-8.101	-8.252847
PKY1: K_{FY} /F_{znom} stiffness maximum value	16.655370023	-16.832	-17.36182
PKY2: Load at which K_{FY} /F_{znom} reaches maximum value	-2.167777133	2.101	2.293896
PKY3: K_{FY} /F_{znom} variation with camber	-0.128306180	-0.10954	-0.110362
PHY1: S_{hy} horizontal shift at F_{znom}	0.002307603	0.0015064	0.001696
PHY2: S_{hy} shift variation with load	0.002355101	0.0043944	0.003882
PHY3: S_{hy} shift variation with camber	0.037702414	0.03976	0.039873
PVY1: S_{vy} /F_z vertical shift at F_{znom}	0.016792719	0.005431	0.006931
PVY2: S_{vy} /F_z shift variation with load	0.268896644	0.022704	0.018685
PVY1: S_{vy} /F_z shift variation with camber	-0.614496538	-0.0771	-0.061575
PVY2: S_{vy} /F_z shift variation with camber and load	-0.397832674	-0.09083	-0.098064

(2) The adjustment that the MF for pure lateral force has to the tyre test data if the MUCA, SVO and IOA are used to optimise the parameters is presented in Figure 7 for camber angle 0º. These tyre model curves are obtained with the lateral force parameters presented in Table 4. Note that in this figure, there are not 82 test points represented for every vertical load, with the objective to clarify the figure.

(3) The sum-squared error between the test data and the lateral force curve, which correspond to the parameters obtained with the SVO method, with the IOA with constant control parameters and the MUCA is shown in Table 5. These errors are calculated for 5 load cases and three camber angles 0º, 4º and -4º. Also, the computing time is shown in Table 5.

(4) The algorithm parameters are: for the estimation phase, Differential Evolution Algorithm: *NP(number of individuals)*=70, *D(number of genes)*=18, *F(perturbing factor)*=0.6, *CR(crossover probability)*=0.4, *MR(mutation probability)*=0.1 and *iterest*=150. For the exploration phase *NX(number of individuals)*=10, *DX(number of genes)*=6 *CRX(crossover probability)*= 0.6, *MRX(mutation probability)*=0.1 and *iterexp*=10.

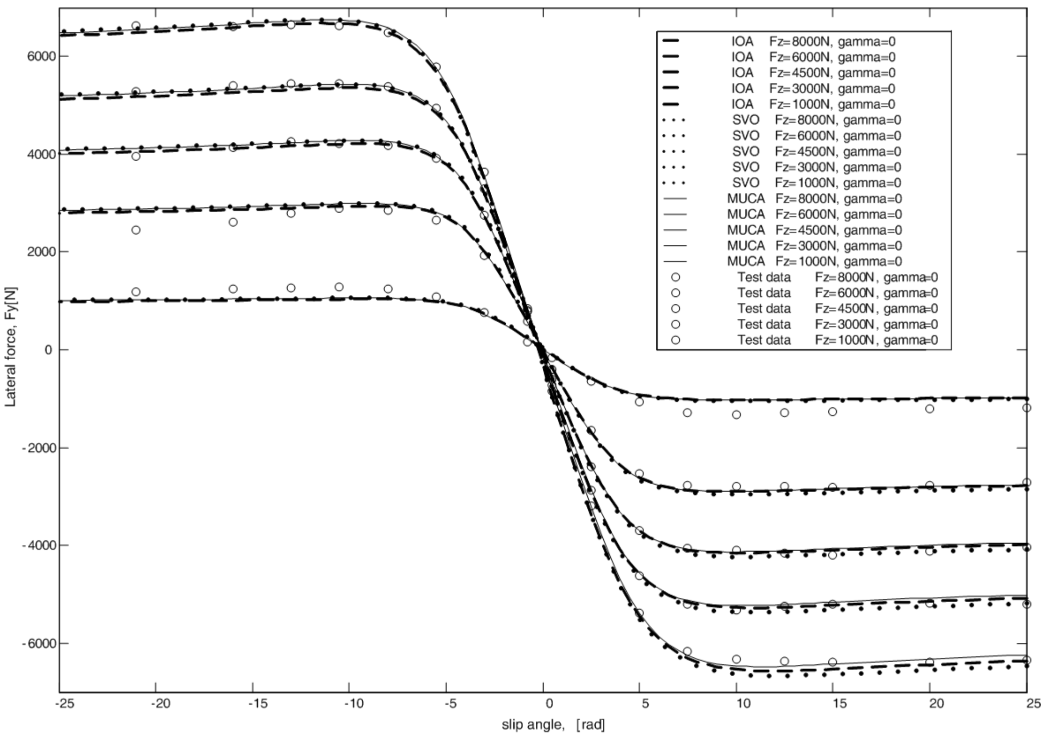

Figure 7.- Lateral force curves in the three studied cases.

Table 5.- Sum-squared error and computing time from lateral force curve to the lateral force test data.

$\sum_{i=1}^{n}\sum_{j=1}^{m}\left[F_{y0}^{Opt.Meth}-F_{y0}^{measured}\right]^2$	Sum squared error [N^2]		
	MUCA	**SVO**	**IOA**
Fz=[8000,6000,4500,3000,1000] N -25º ≤ α ≤ 25º γ=[-4,0,4] n = 82 data points m = 5 load cases	2.1002223x 10^7	2.3860189 x 10^7	1.872549 x 10^7
Computing time	52 s.		7 min 34 s

Again, to investigate the performance of the algorithms, we made a statistical study to evaluate the behaviour of the lateral force model. 150 different runs of the whole algorithm were carried out with the following conditions: the main loop of the algorithm was run 15 times in each run, i.e., iterloop=15 (see Figure 1). The exploration and estimation phases loop was run 100 times each one, i.e, iterest=100 and iterexp=100 (see Figure 1). Each time the main loop of the algorithm was run, a lateral force error was recorded and we obtained the following figure.

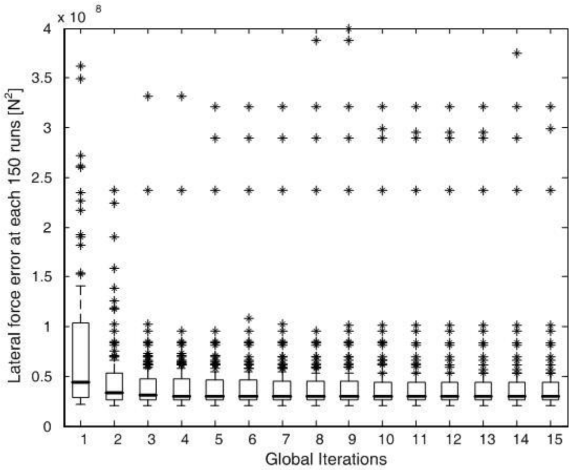

Figure 8.-Statistical parameters at 150 runs of the main algorithms. Lateral force error

Figure 8 shows a box plot for every global iteration. We can observe how the median of the error data is decreasing with the global iterations obtaining the best result in the last iterations. So we can see how the evolution of the algorithm improves the result in the initial global iterations. We can also observe how the box is narrower in each iteration. This is a measure indicating that dispersion of data is improving with the iterations of the algorithms. In this case we obtain good median values and a low level of dispersion of data in few iterations.

Table 6.- Statistical comparisons among several methods.

Lateral force error	MUCA	SVO	IOA
Standard deviation	$2.2860x10^8$		$9.03937x10^8$
Mean	$7.53708x10^7$		$8.3965x10^8$
Best	2.10022×10^7	4.266528×10^7	1.8725×10^7
Worst	$2.1216x10^9$		$1.9154x10^9$
Evaluations number	9525000		82000000
Computing time	52 s.		7 min 34 s

To compare the statistical results of other methods, we also ran the IOA algorithm 150 times with 2000 global iterations and we obtained the following results that you can see in Table 6.

Aligning torque

Here, the results for pure aligning torque are presented:

(1) For the Magic Formula Delft Tyre 96 version, the aligning torque parameters optimised by the MUCA with the lowest error out of the 150 repetitions for the statistical study are presented in Table 7.

Table 7.- Aligning torque coefficients in the three studied cases.

Aligning torque coeficients	MUCA	SVO	IOA
QBZ1: B_{pt} trail slope factor for trail at F_{znom}	11.2675	11.266	10.47871
QBZ2: B_{pt} slope variation with load	-3.1694	2.802	-2.04739
QBZ3: B_{pt} slope variation with square load	7.67038	-0.586	6.09958
QBZ4: B_{pt} slope variation with camber	0.28477	0.405	0.39191
QBZ5: B_{pt} slope variation with absolute camber	0.12004	-0.020833	-0.01279
QCZ1: C_{pt} shape factor for pneumatic trail	1.19938	1.1908	1.18899
QDZ1: Peak trail $D''_{pt} = D^*_{pt} (F_z/F_{znom} \times R_0)$	0.11509	0.11738	0.11411
QDZ2: D''_{pt} peak variation with load	0.00992	0.009433	0.00479
QDZ3: D''_{pt} peak variation with camber	0.01970	0.13701	0.12582
QDZ4: D''_{pt} peak variation with square camber	0.10563	-2.8255	-1.10927
QEZ1: E_{pt} trail curvature at F_{znom}	-2.10223	-2.3346	-2.88862
QEZ2: E_{pt} curvature variation with load	-0.88169	1.7478	-0.83490
QEZ3: E_{pt} curvature variation with square load	3.10345	0	3.71462
QEZ4: E_{pt} curvature variation with α-t sign	0.05432	0.25	0.15337
QEZ5: E_{pt} curvature variation with camber and α-t sign	1.51871	2.1345	1.33032
QHZ1: S_{ht} horizontal shift for trail at F_{znom}	-0.00089	2.5437E-4	0.00024
QHZ2: S_{ht} shift variation with load	0.00642	3.316E-4	0.00102
QHZ3: S_{ht} shift variation with camber	0.19143	0.08772	0.12610
QHZ4: S_{ht} shift variation with camber and load	-0.05712	0.04964	0.01917
QBZ9: B_r slope factor of M_{zr} residual torque	4.06038	29.996	17.87469
QBZ10: B_r slope factor of M_{zr} residual torque	2.46784	0	5.66140
QDZ6: : $D''_{mr} = D^*_{mr} (F_z \times R_0)$ peak residual torque	-1.89367	-7.047E-4	-0.00049
QDZ7: : D''_{mr} peak factor variation with load	1.19869	1.8184E-4	0.00025
QDZ8: : D''_{mr} peak factor variation with camber	-7.60853	-0.15867	-0.18840
QDZ9: : D''_{mr} peak factor variation with camber and load	-6.38017	-0.02444	0.00415

(2) The adjustment that the MF for pure aligning torque has to the tyre test data if the MUCA, SVO and IOA are used to optimise the parameters is presented in Figure 9 for camber angle 0º. These tyre model curves are obtained with the aligning torque parameters presented in Table 7. Note that in this figure there are not 74 test points represented for every vertical load with the objective to clarify the figure.

(3) The sum-squared error between the test data and the aligning torque curve, which correspond to the parameters obtained with the SVO method, with the IOA with constant control parameters and the MUCA is shown in Table 6. These errors are calculated for 5 load cases and seven camber angles -4º, -2º ,-1º, 0º, 1º, 2º and -4º. Also, the computing time is shown in Table 8.

(4) The algorithm parameters are: for the estimation phase, Differential Evolution Algorithm: *NP(number of individuals)*=200, *D(number of genes)*=25, *F(perturbing factor)*=0.6, *CR(crossover probability)*=0.4, *MR(mutation probability)*=0.1 and *iterest*=200. For the exploration phase *NX(number of individuals)*=5, *DX(number of genes)*=6 *CRX(crossover probability)*= 0.6, *MRX(mutation probability)*=0.1 and *iterexp*=100.

Table 8.- Sum-squared error and computing time from aligning torque curve to the aligning torque test data.

$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{g=1}^{l}\left[M_{z0}^{Opt.Meth} - M_{z0}^{measured}\right]^{2}$	Sum squared error [N^2m^2]		
	MUCA	**SVO**	**IOA**
Fz=[8000,6000,4500,3000,1000] N -25º ≤ α ≤ 25º γ=[-4,-2,-1,0,1,2,4] n = 82 data points m = 5 load cases	95563	167025	108615
Computing time	5 min.		47 min 56 s

The same as in the paragraphs above, the statistical study is shown in Figure 10.

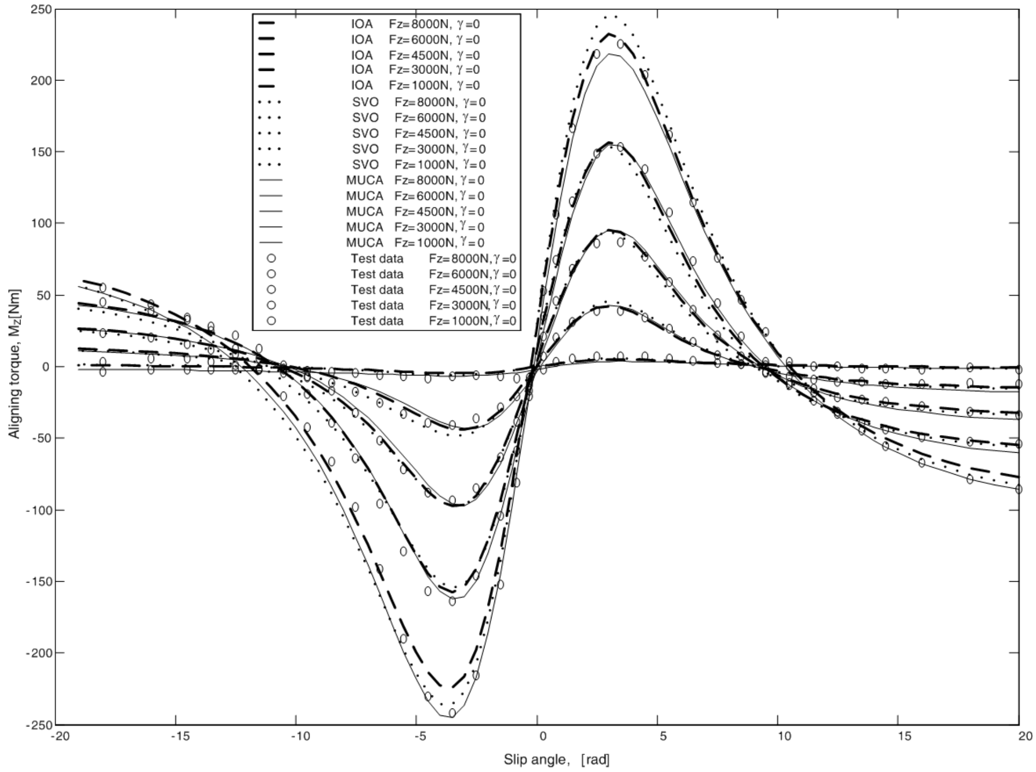

Figure 9.- Aligning torque curves in the three studied cases.

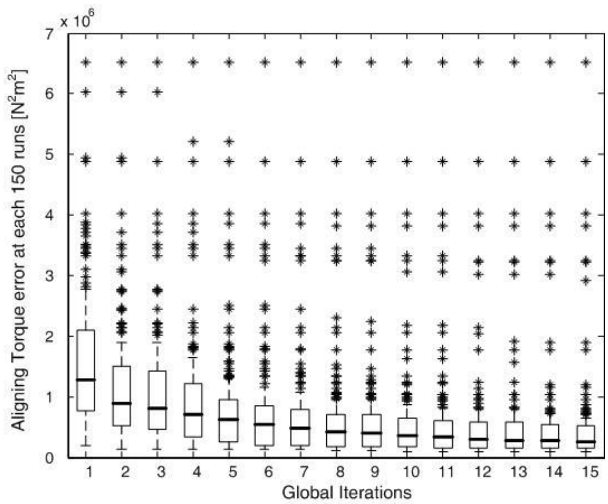

Figure 10.- Statistical parameters at 150 runs of the main algorithms. Lateral force error

To compare the statistical results of other methods, we also ran the IOA algorithm 150 times with 3000 global iterations and we obtained the following results that you can see in Table 9.

Table 9.- Statistical comparisons among several methods.

Aligning torque error	MUCA	SVO	IOA
Standard deviation	$1.0977x10^{6}$		$1.69595x10^{6}$
Mean	$5.7719x10^{5}$		$7.48416x10^{5}$
Best	9.55638×10^{4}	1.67025×10^{5}	$1.0669x10^{5}$
Worst	$8.5756x10^{6}$		$8.5473x10^{6}$
Evaluations number	54225000		519000000
Computing time	7 min		47 min 56 s

We obtain similar results as in the paragraphs above, so we can establish a good performance of the proposed algorithm in the three studied cases.

The reliability of the optimization technique for tyre model identification is shown in Figures 6, 8 and 10. Note that in the final global iteration, most of the 150 runs carried out, are very close to the median and there are very few cases outside the upper and lower quartile interval. Besides, a statistical study among several methods has been made. You can see it in Table 3, 6 and 9. You can observe that the number of evaluations is always lower in the MUCA than in the IOA. In addition, the mean and the standard deviation is lower in the MUCA than in the IOA for lateral force and aligning torque, being the best and the worst better in the MUCA than in the IOA for aligning torque. As many numbers of parameters and evaluations have the problem, the IOA algorithm shows more scattered behaviour than the MUCA. i.e. IOA and MUCA are dependant of the kind of objective function and this dependence is more emphasized for the IOA when the objective function has more parameters to be optimized.

Conclusions

We have presented a coevolutionary algorithm (MUCA) which is used to solve an engineering problem. Concretely, we have solved a tyre model problem, providing Magic Formula Tyre model coefficients that make the adjustment to tyre test data very accurate. The Magic Formula Tyre model for the description of steady tyre behaviour in vehicle dynamics simulations is very accurate and widely used, but the determination of the Magic Formula coefficients may be difficult. In this work we have used one new technique (MUCA) to obtain the coefficients and compare them with another two optimization strategies (SVO and IOA).

The adjustment level of the solution is higher than the one obtained with the SVO technique and very close to the one obtained with the IOA technique with few significant differences in all cases shown. The tyre model parameters obtained with the three optimization techniques are able to model dynamic tyre behaviour. However, those parameters obtained with the MUCA are gotten in less time, with less error with respect to experimental data.

We have to pay attention to the fact that, in the IOA and SVO techniques all the tyre test data are used in the optimization process, so they have more information to solve the problem than the proposed algorithm. For that reason the MUCA algorithm improves the computing time markedly.

We also pointed out a solution to improve the performance and avoid the drawbacks of coevolutionary strategies. We have used a test data memory record with three test data generations to avoid disengagement.

Due to that, coevolution is used as a powerful tool for optimization of multidimensional problems with multiple local minima.

In this paper we used an estimation phase to find out the tyre model parameters which can predict actual tyre test data and an exploration phase to find out new test data which have the most disagreement with the response of the current model.

We have obtained several advantages with this coevolutionary approach improving the obtained results in previous works. The advantage, among others, is the decrease of computing time of about ten times in the three cases studied. A lower tyre test data number is needed to evolve every generation of tyre model parameters. Also, if we assume that traits are heritable, tyre test data passed on by parents will be passed on by offspring, which allows these test data to be omitted. Hence a higher level of efficiency will be reached.

References

[1] W. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. In C. Langton, C. Taylor, J. Farmer and S. Rasmussen, editors, *Artificial Life II, SFI Studies in the Sciences of Complexity*, volume 10, pages 313-323. Addison-Wesley Publishing Co., 1991.

[2] W. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. In C. Langton, C. Taylor, J. Farmer and S. Rasmussen, editors, *Artificial Life II*, pages 264-270. Massachusetts: Addison-Wesley, 1992.

[3] J. Morrison and F. Oppacher. A general model of co-evolution for genetic algorithms.

[4] R.A. Watson and J. B. Pollack. Coevolutionary Dynamics in a Minimal Substrate. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), pp. 702-709, San Francisco, 2001.

[5] C. Bergstrom and M. Lachmann. The Red King Effect: When the slowest runner Wins the Coevolutionary Race. Proceeding of the National Academy of Sciences, USA 100(2), pp. 593-598.

[6] C.D. Rosin and R.K. Belew. New Methods for Competitive Coevolution. Evolutionary Computation, 5(1), pp. 1-29. (1997).

[7] H. Lipson, J. Bongard, V. Zykov. Coevolutionary methods in systems design and analysis. 15th International CIRP Design Seminar, Shanghai, China. (2005)

[8] S.G. Ficici. Solution Concepts in Coevolutionary Algorithms. Doctoral Thesis. (2004)

[9] Y.K. Kim, J.Y. Kim, Y. Kim. A Tournament-Based Competitive Coevolutionary Algorithm. Applied Intelligence, vol. 10, pp. 267-281. (2004).

[10] B. Liu, H. Ma, X. Zhang. and Y. Zhou. A Memetic Coevolutionary Differential Evolution Algorithm for Constrained Optimization. IEEE Congress on Evolutionary Computation (CEC 2007). (2007).

[11] Kouchmeshky, B., Aquino, W., Bongard, J.C. and Lipson, H. Coevolutionary algorithm for structural damage identification using minimal physical testing. International Journal for Numerical Methods in Engineering, vol. 69, nº 5, pp. 1085-1107. (2007)

[12] Huang, F.Z., Wang, L. and He, Q. An effective coevolutionary differential evolution for constrained optimization. Applied Mathematics and Computation, vol. 186, nº 1, pp. 340-356. (2007).

[13] He, Q. and Wang, L. An effective coevolutionary particle swarm optimization for constrained engineering design problems. Engineering Applications of Artificial Intelligence, vol. 20, nº 1, pp. 89-99. (2007).

[14] Dugoff, H., Fancher, P.S. and Segel, L. An analysis of tire traction properties and their influence on vehicle dynamic performance. *International Automobile Safety Compendium, FISITA/SAE*, Brussels, SAE No. 700377. (1970).

[15] Gim, G. and Nikravesh, P.E. Ananalytical study of pneumatic tire dynamic properties, Part 1. *Internacional Journal of Vehicle Design*, 11(6), pp. 589–618. (1991).

[16] Pacejka, H.B. and Bakker, E. The Magic Formula Tyre Model. *Vehicle System Dynamics*, 21, pp. 1–18.(1993).

[17] Bayle, P., Forissier, J.F. and Lafon, S. A new tyre model for vehicle dynamics simulations. *Proceedings of Automotive Technology International '93*, pp. 193–198. (1993).

[18] Pacejka, H.B. and Besselink, I.J.M. *Magic formula tyre model with transient properties*. Vehicle System Dynamics Supplement, 27, pp. 234-249. (1997).

[19] van Oosten, J.J.M. and Bakker, E. Determination of Magic Tyre Model Parameters. Vehicle System Dynamics. 21, pp. 19–29. (1993).

[20] Goldberg, D.E. *Genetic algorithms in search, optimization and machine learning,* Chapter 1, 'A gentle introduction to genetic algorithms', ISBN: 0201157675, Addison-Wesley. (1989).

[21] Cabrera, J.A., Ortiz, A., Carabias E. and Simon, A. *An alternative method to determine the Magic tyre model parameters using genetic algorithms*. Vehicle System Dynamics, vol. 41, nº 2, pp. 109–127. (2004).

[22] Cabrera, J.A., Ortiz, A., Carabias E. and Simon, A. *Experience with the IMMa tyre test bench for the determination of tyre model parameters using genetic techniques*. Vehicle System Dynamics, vol. 43, Supplement, pp. 253–266. (2005).

[23] Ortiz, A., Cabrera, J.A. Guerra, A.J. and Simon, A. *An easy procedure to determine optimization technique and the IMMa optimization algorithm*. Vehicle System Dynamics, vol. 44, nº 9, pp. 689–718. (2006).

[24] Storn, R.and Price, K. *Differential Evolution. A Simple and Efficient Heuristic Scheme for Global Optimization over Continuos Spaces*, Journal of Global Optimization, vol. 11, pp. 341-359. (1997).

[25] Brest, J., Greiner, S., Boskovic, B., and Mernik, M. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evolution. Comput. 10(6), pp. 646–657. (2006).

[26] Ortiz, A., Cabrera, J. A., Guerra, A. and Simon. A The IMMa optimisation algorithm without control input parameters. Vehicle System Dynamics, .vol. 47, nº 2, pp. 243-264. (2009).