Applying the Multi-objective Optimization Techniques in the Design of Suspension Systems

Adrian Florea[1,*],Ioana Ileana Cofaru[1], Lucian ROMAN[2], Nicolae Cofaru[2]
[1] Computer Science and Electrical Engineering Department
[2] Industrial Engineering and Management Department
'Lucian Blaga' University of Sibiu, Romania

JDIM

Journal of Digital Information Management

ABSTRACT: *The questionable quality of the roads represents the main factor of discomfort, being directly responsible for the accidents, affecting car components, but also the security of passengerscausing death and serious injuries. According to statistics released by the World Health Organization, road accidents, in underdeveloped countries, tends to increase by 80 % in 2020 compared to 2000. In terms of road infrastructure,the low-and middle-income countries are characterized by a higher accident rate, reason for which the cars designers must approach the suspension problem slightly different and the parameters obtained by optimization algorithms should be differentfrom the same model of car depending on where they will be driven / sold.This paper presents the optimization of a quarter-car model with two degree-of-freedom using evolutionary algorithms to determine the optimal parameters for a vehicle suspension, in order to improve ride comfort. The optimization problem consists in minimizing the sprung mass acceleration and sprung mass displacement subject to several constraints that arise from kinematic considerations. The vehicle model is considered to travel at a constant speed on a random road profile generated according to the ISO 8608 standard. The design variables to be optimized are the suspension stiffness and damping coefficients. We analyzed the algorithms in multiple scenarios so we can compare their performance in terms of fast convergence and solution diversity. The results showed that the optimization algorithms find solutions in small number of iterations, with slightly better performance obtained by Fast Pareto Genetic Algorithm.*

Subject Categories and Descriptors
G.1.6 [Optimization]: I.1.2 Algorithms; **D.2.2 [Design Tools and Techniques]:** Evolutionary prototyping

General Terms
Multi-objective designs, Optimization techniques, Genetic algorithms, Car design

Keywords: suspension optimization; quarter-car; evolutionary algorithms; genetic algorithms; multi-objective

Received:
12 August 2016, Revised 29 September 2016, Accepted 3 October 2016

Highlights:

• This work argues that, due to the road infrastructure, the underdeveloped countries exhibit a higher accident rate, reason for which the car's designers must approach the suspension problem slightly different and the parameters obtained by optimization algorithms should be tuned differently and carefully chosen for the same model of car, depending on geographic place where they will be driven / sold.

• The research is focused on the optimization of a quarter-car model (QCM) with two degree-of-freedom using evolutionary algorithms (Pareto and swarm-intelligence-based) to determine the optimal parameters for a vehicle suspension, in order to improve ride comfort.

• In order to determine the optimum parameters of suspensions we developed a software application in Microsoft Visual Studio 2012 (C#), .NET Framework 4.5 using Newtonsoft.JSON and ZedGraph packages.

• We analyzed the algorithms in multiple scenarios (road profile of different roughness, different traveling speeds) so we can compare their performance in terms of fast convergence and solution diversity. As expected, travelling at high speeds showed a significant increase of the sprung mass acceleration, while a road with high degree of roughness making it much worse.

• Our application contributes to improve the quality of spare parts production that are required by the suspensions mechanism through focusing on specific directions which depends on the geographic area, the infrastructure of the region, environmental conditions, characteristics of fuels, etc. Moreover, due to vehicle's operating conditions, the manufacturers should tailor the components of the suspensions mechanism and differentiate the period in which they perform maintenance revision.

1. Introduction

The suspensions system optimization is usually considered related to its ability to provide stability, a high standard of ride comfort, safety and handling under different exploiting conditions, minimizing the passengers discomfort during movement on uneven roads. In addition, a challenge refers to anticipating the vehicle's failure when runs on poor quality roads at different speeds. The optimization study that we present in this work starts from the importance exhibited by a competitive suspensions mechanism for appreciating the vehicle quality, considering the diversifying constructive types in order to satisfy the most demanding requests of vehicle drivers. The first motivation in achieving such a study refers to the technical impact introduced by an optimized suspension which targeting issues like the possibility of increasing the payload, the average travel speed, the stability, to reducing the braking distance, and so on. Very important is also the social impact because increasing passenger comfort, improve traffic safety, reduce the number of possible accidents has been and will be a continuing concern for the design engineers of suspensions mechanism. Another motivation targets the applicability of our research work. This is due to the fact that in our city (Sibiu) operates ThyssenKrupp BilsteinCompa S.A, a powerful manufacturer of suspension elements (dampers) for west European market and, by the collaboration that we have with it, allows us testing and experimental validating the created models. Finally yet importantly, the authors experience in designing parts using evolutionary algorithms will help accelerate and refining of the optimization process. An important issue that worth mentioning is considering the road profile. Creating a template for roads, which emphasizes the surface irregularities, and integrating it in the optimization methodology provides a realistic character to the design process. Also, allows developing a customized pattern of suspensions depending on the existing infrastructure in the running area of the car.

A moving vehicle produces vibrations due to multiple external factors like road disturbances, aerodynamic forces, vibrations of different components of the vehicle and weather conditions. These vibrations have a significant negative impact on the ride comfort which can cause motion sickness for the passengers or even damages to the vehicle itself. Because of this, it is necessary to develop highly optimized suspension mechanisms that can overcome such of issues.

The major contributions of our research focus on at least twofold:

First, it has theoretical and scientific implications: we made comparisons of optimization algorithms of suspensions. These belong to multi-objective evolutionary algorithms that help accelerating and refining of the optimization process. Also, another advantage exhibited is that we are able to provide a lot of solutions not just one. We analyzed multiple scenarios (road profile of different roughness, different traveling speeds) so we can compare the algorithms performance in terms of fast convergence and solution diversity. An important issue that worth mentioning is considering the road profile. Creating a template for roads, which emphasizes the surface irregularities, and integrating it in the optimization methodology provides a realistic character to the design process. Also, allows developing a customized pattern of suspensions depending on the existing infrastructure in the running area of the car. The results showed that the Pareto and swarm-based optimization algorithms used are able to find optimal solutions in a small number of iterations, with slightly better performance obtained by the Fast Pareto Genetic Algorithm.

Second, it has practical implications: our research has a positive impact on the quality of spare parts production that is required by the suspensions mechanism by reducing the number of replaced parts. Due to vehicle's operating conditions, the car's designers should tailor the components of the suspensions mechanism and differentiate the period in which they perform maintenance revision depending on geographic place where they will be driven / sold, the infrastructure of the region, environmental conditions, characteristics of fuels, etc. The suspensions mechanism parameters obtained by optimization algorithms should be tuned differently and carefully chosen even for the same model of car taking into account the running conditions.

The main aim of this study is to use several multi-objective evolutionary algorithms in order to provide the best solutions to optimize a suspension system, for different road conditions.

To simulate the vehicle dynamics we used a passive QCM with two-degree of freedom that is excited by a road prowith different levels of roughness. The optimization process consists in finding the optimal design variables for the stiffness and damping coefficients in order to minimize the sprung mass acceleration and sprung mass displacement. The problem is subject to several constraints that arise from kinematic considerations like the maximum vertical acceleration of the vehicle, suspensions working space (SWS) and the natural frequency of the system. The road profiles used for this study fall into the category of random road profiles and were generated according to ISO 8608. We also aimed to compare results from different algorithms to see which one gives the best results in terms of fast convergence and solution diversity. Multiple algorithms were used in this paper, multi-objective Pareto algorithms such as NSGA-II [1-2], SPEA2 [3], FastPGA [4], swarm-intelligence-based algorithms like SMPSO [5], but also traditional methods (non-Pareto) such as Vector Evaluated Genetic Algorithm (VEGA) and a classic genetic algorithm (GA) that aggregates the objectives using a weighted-sum approach.

The organization of the rest of this paper is as follows. In section 1 we shortly describe the theoretical background related to general solutions applied in suspensions optimization, whereas section 2 presents the basic components and functions of a vehicle suspension's system, the mathematical model of the dynamic system and the road profiles required for software implementation. The problem description is summarized in section 3. The next two sections (4 and 5) present the evolutionary algorithms (non-Pareto, Pareto and swarm behaviour) and how are they applied in our research. Section 6 describes the application graphical user interface.Section 7 illustrates some simulation results. Finally, section 8 suggests directions for future work and concludes the paper.

2. Related Work

There has been a lot of interest in this research field in the last decades, mainly because of the fact that auto vehicles became such an important aspect of our daily lives. With the constant growing of computational power, new methods based on computer simulation and optimization are developed, to help engineers create better and optimized designs. In [6] the authors used a gradient projection method in order to find the optimal design of a suspension system using a QCM and a road profile described by a sine wave. A major problem of this technique is that it performs a local optimization only and may possibly stuck in a local minima point, for certain starting points. In [7] the authors analysed the performance of QCM using linear and non-linear parameters. The results showed a significant difference in response between linear and non-linear models. More importantly, the results obtained with theoretical simulations models were approximately the same as that of real-world experimental, showing that simulations can be used effectively instead of a complex experimental setup. The usefulness of GA in vehicle suspension optimization was tested in [8-11]. The authors used a classic mono-objective genetic algorithm and a passive two degree-of-freedom QCM. Their experiment showed that genetic algorithms were able to find optimal suspension systems in a relatively small number of iterations. Of course, this method did not truly exploit the potential of multi-objective optimization, the objective were aggregated using a weighted sum approach. In [12] the authors present an optimization of a four-degrees-of-freedom (4-DOF) vehicle's human with seat suspension system using weighted sum genetic algorithms (with fixed weights) to determine suspension parameters. Unlike other researchers, in this paper, we performed a global optimization of a passive suspension model using various types of multi-objective evolutionary algorithms, and then we compared the performance of these algorithms.

3. Suspension Design

The suspension system is a mechanism that links the vehicle body to the tires. Its primary use is to isolate the body from forces generated by road irregularities and keep the tire in permanent contact with the road [13]. A well-designed suspension should improve ride comfort, safety and maneuverability. Usually, suspension systems are classified in three categories: passive, semi-active and active. The passive model is the most common model used in simulations due to its simplicity. To describe the vehicle's dynamics we considered a passive QCM with two-degrees of freedom, a model widely used in automotive engineering (see Fig. 1).Comparing with the active or semi-active variants of suspension system, the passive model based on springs and dampers though being relatively rigid to the excitation experienced from the road, is still widely used in low and medium priced passenger cars.

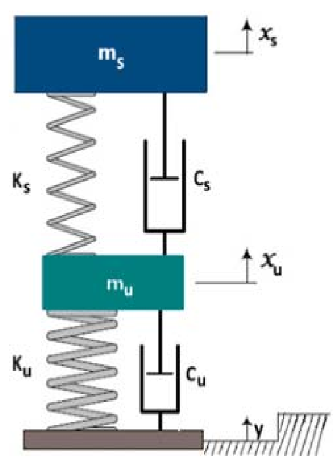

Figure 1. Quarter-car model (2 DOF)

QCM is a basic model used in simulations and performance testing. Two masses are considered: sprung mass (m_s) and the unsprung mass (m_u). The sprung mass represents 1/4 of the vehicle body mass and the unsprung mass is the mass of one wheel. The sprung mass is supported by a spring of stiffness K_s and a damper with damping coefficient of C_s. The wheel is in direct contact with the road surface through the tire which has a spring stiffness of K_u and damping of C_u. Both masses are considered to be solid and have a vertical movement represented by x_s and x_u. The free movement of these masses makes this system a two degree of freedom system. In this model, we assume that the tire is in permanent contact with the road and the roll effect is negligible. The spring stiffness K_s and damping coefficient C_s have the most significant influence on the suspension's behaviour and will represent the main point of interest in finding the optimal configuration.

3.1 Mathematical model

Using theories and formulas from mathematics and physics (laws of motion, spring theory) we can express the mechanical model in Fig. 1 through a system of differential equations known as equations of motion [14].

$$[\mathrm{M}]\ddot{x} + [\mathrm{C}]\dot{x} + [\mathrm{K}]x = \mathrm{F}, \text{ where } x = (x_s, x_u) \qquad (1)$$

This is a second order differential equation system where *M, C, K* represent the mass, damping, and stiffness matrices. *F* is the force that arise from road unevenness. Rewriting this system in matrix form we obtain:

$$\begin{bmatrix} m_s & 0 \\ 0 & m_u \end{bmatrix} \begin{bmatrix} \ddot{x}_s \\ \ddot{x}_u \end{bmatrix} + \begin{bmatrix} C_s & -C_s \\ -C_s & C_s + C_u \end{bmatrix} \begin{bmatrix} \dot{x}_s \\ \dot{x}_u \end{bmatrix} +$$

$$\begin{bmatrix} k_s & -k_s \\ -k_s & k_s + k_u \end{bmatrix} \begin{bmatrix} x_s \\ x_u \end{bmatrix} = \begin{bmatrix} 0 \\ k_u y + C_u \dot{y} \end{bmatrix} \qquad (2)$$

where y is the input road profile

From which,

$$\ddot{x}_s = \frac{1}{m_s}[-k_s(x_s - x_u) - C_s(\dot{x}_s - \dot{x}_u)] \qquad (3)$$

$$x_s = \frac{1}{m_u}[k_s(x_s - x_u) - C_s(x_s - x_u)] - [k_u(x_u - y) - C_u(x_u - y)] \qquad (4)$$

A demonstration for this formula can be found in [13]. Solving these equations will produce the values of *x* and its derivatives. We solved this system by numerical integration using the well-known integration method, Runge-Kutta [15].

3.2 Road profiles

The road is a transport structure which is subject to vehicle motion forces. Having knowledge about the forces that occur on the surface road and their frequency spectra is necessary in engineering. The irregularities found on the road surface act as input excitation for the suspension system. To simplify modelling, only vertical movements are considered. Pitch and roll are ignored. A simple method assumes that the road is a simple sinusoid of a given frequency and amplitude [7], [16]. The road profile is approximated by the following formula:

$$y = A \sin \frac{2\pi \mathrm{V}}{\lambda} t \qquad (5)$$

the meaning of the parameters is as follows:

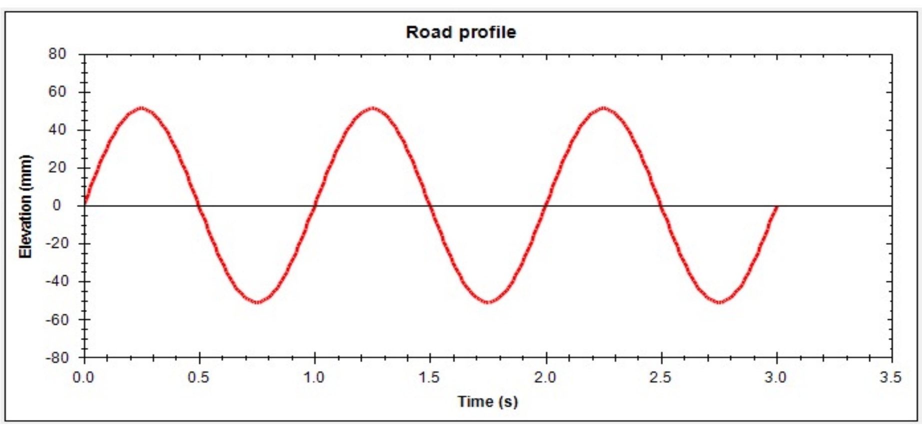

Figure 2. A sinusoidal road profile

y road elevation at time *t*

A amplitude

λ wavelength

V vehicle velocity

This profile will give a harmonic excitation to the vehicle and is useful to test the behaviour of the suspension system at certain frequencies and to study different phenomenon like resonance that can occur during the motion of the vehicle.

To simulate a more real-like road, more advanced techniques are needed. For this matter, the ISO 8608 standard proposes a classification based on power spectral density [14], [17]. The profile is characterized by different variations in elevation of the road surface measured along one track and parallel with the road. In this way, a road is a combination of a large number of longer and shorter holes of different amplitudes [18 – 20]. The ISO approximates the road profile using a PSD function:

$$G_d(\Omega) = G_d(\Omega_0)\left(\frac{\Omega}{\Omega_0}\right)^{-2} \qquad (6)$$

Where n is the spatial frequency and Ω epresents the angular spatial frequency.

Researchers demonstrated that, if a PSD function is known, it is possible to generate a profile road using a superposition of sinusoids, considering random phases which follow a uniform probabilistic distribution in the 0- 2π interval. The formula used to generate the profile is:

$$h(x) = \sum_{i=0}^{N} \sqrt{2.\, \Delta n. G_d(n)}.\mathrm{Cos}\,(2\pi.n_i.x + \varphi_i) \qquad (7)$$

L road length

n *a* vector of spatial frequencies

φ_i random phases

The values for $G_d(\Omega)$ are found in Table 1.

Table - ISO 8608		
Road class	$G_d(\Omega_0)$ $(10^{-6} m^3)$	
	Lower Limit	Upper Limit
A	-	2
B	2	8
C	8	32
D	32	128
E	128	512
$\Omega_0 = 1$ rad / m		

Table 1. Road roughness classification according to ISO 8608

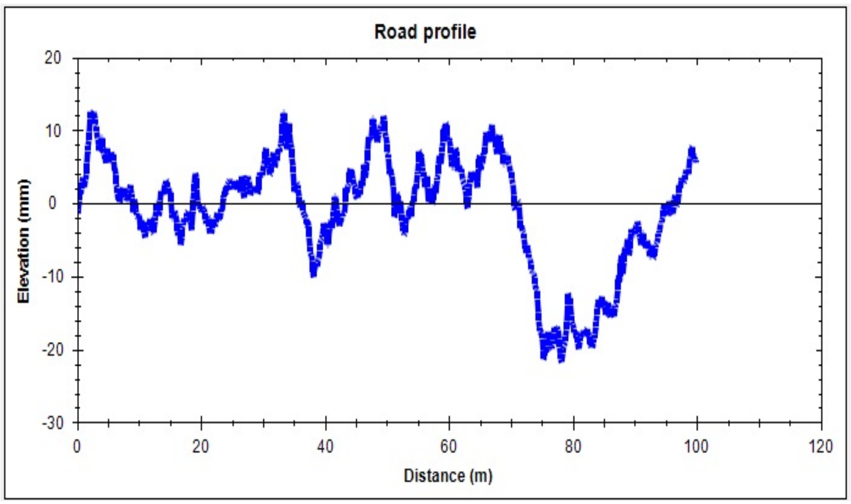

Figure 3. A sample random road profile

According to the above classification, five classes of road profiles can be distinguished from A to E where *A* class profiles contain a small degree of roughness and can be regarded as a good quality road and *E* is of the poorest quality.

4. Problem Description

The main functionality of a suspension system is to isolate vibrations produces at the wheel level by the road conditions. These vibrations can make the body of the passenger vibrate violently which can make the ride uncomfortable especially when the exposure time is high. Thus, it is necessary to design competitive suspensions mechanisms that increase the ride comfort and maneuverability. Regarding the ride comfort, designing a suspension system involves choosing the right characteristics in such a way that the vertical acceleration and vertical displacement of the sprung mass are minimized. These objectives are usually conflicting: reducing one does not also reduce the other. Moreover, the existence of some design constraints complicates even more the designer's task. In this paper, we defined three constraints [8]. The first represents the maximum value of the sprung mass acceleration that should not exceed 1g (9.8m/s²). The second constraint specifies the maximum displacement between the sprung and unsprung mass. The third constraint refers to the frequencies at which humans experience motion sickness. According to ISO 2631, humans experience motion sickness when exposed to frequencies in the interval 0.1-1Hz. Thus, the natural frequency (ω) of the suspension system should be greater than 1Hz.

$$\text{minimize } x_s \text{ and } \ddot{x}_s$$

$$\textit{Subject to}$$

$$|\ddot{x}_s| < 9.8\ m/s^2$$

$$|x_s - x_s| < 0.1\ m$$

$$\omega > 1\ Hz$$

This makes the problem a perfect candidate for multi-objective optimization.

5. Overview of Optimization Algorithms

This section presents a brief overview of the optimization algorithms used in this paper and that were software implemented for obtaining the experimental results.

5.1 Non-Pareto techniques

5.1.1 Classic GA with objective aggregation

Genetic algorithms are adaptive heuristic methods based on the principles of biological evolution: natural selection and genetic inheritance [21].

Pseudocode 1. *Genetic algorithm – the elitist approach*

1: Choose an initial population *P* generated randomly of *N* individuals

2: Evaluate the fitness of population

3: **repeat**

4: Select the best individuals

5: Generate new individuals using crossover and mutation

6:Evaluate fitness of population

7: Select the best N from the N+2 individuals (the initial populations enhanced with the new two generated). These will form the population *P* in the next generation

8: **until stop condition**

The basic principle of GA is to update a population of individuals (chromosomes) in an iterative manner, over a number of generations. At each generation, the individuals are evaluated using a fitness function. A new generation is obtained by selecting the best individuals from the current population based on their fitness. A better fitness means higher chances of reproduction, which is done through the crossover process. As in nature, an individual can suffer mutations that are applied using the mutation operator. The new generation of solutions obtained this way are usually better compared to the previous ones. Once the population does not produce any significantly better solutions, the algorithm is considered to converge and can be stopped. Because the classic GA is only suited for mono-objective problems, some modifications are needed. In [22], are proposed two weighted-sum methods (random and adaptive) where the objectives are aggregated into one by assigning a weight to each objective and sum them together. Because choosing the right weights can be a problem, random weights are generated each time an individual is created. This way, it has been observed that the algorithm covers more of the searching space and has less probability to converge prematurely. Another drawback of this technique is the impossibility of finding all solutions for problems having non-convex Pareto-optimal front.

5.1.2 Vector Evaluated Genetic Algorithm (VEGA*)*

The VEGA algorithm was proposed by Schaffer [23] and extends the classic GA to use more than one objective function. The major difference between them consists in the selection process. The selection was modified so at each generation the current population is divided into subpopulations, each subpopulation executing selection taking in consideration one objective at a time. Thus, for a number of *n* objectives, the population is divided into *n* sub-populations. Each sub-population performs the selection using an objective function. Then, the *n* sub-populations are shuffled and combined together into a new population.From this point, the algorithm follows the same steps as the original algorithm. The main disadvantage is that the population tends to converge towards solutions that are very good in one objective, but very weak in others.

5.1 Pareto techniques

5.2.1 NSGA-II algorithm

The NSGA-II algorithm [2] improves the original Non-dominated Sorting Genetic Algorithm (NSGA) in terms of complexity and preserving diversity, being a multi-objective optimization algorithm that uses elitism. Starting with an initial population, the algorithm applies genetic operators (selection, crossover, mutation) to obtain a new population. The old and new populations are combined together, than a sorting is done using non-dominance as criteria. The ranks and crowding distance (density of individuals surrounding a particular one) are used to guide the selection. An individual is better, if it has a smaller rank, or in case of equality, the one with the bigger crowding distance. NSGA-II is one of the most used algorithms in experiments due to its capacity of generating good solutions regardless of the problem.

Pseudocode 2. *NSGA-II*

1: Generate an initial population of size *N*

2: Evaluate the population

3: Assign a rank to each individual based onPareto dominance

4: Generate a new population

5: Selection

6: Crossover and mutation

7: **for i = 1 to number_of_generations**

8: **foreach parent and child**

9: Assign a rank based on Pareto dominance

10: Generate non-dominated solutions over a Pareto front

11: Add new solutions to the generation starting from the first front until *N*individuals are found

12: **end foreach**

13: Select solutions (elitism) from inferior front which are outside the crowding distance

14: Create new generation

15: Selection

16: Crossover and mutation

17: **end for**

5.2.2 SPEA-II algorithm

The Strength Pareto Evolutionary Algorithm (SPEA) uses an external population called archive that consists of non-dominated solutions found in previous iterations [24]. At each generation, non-dominated individuals are copied to the archive. For each individual a value called strength is calculated. This value is proportional to the number of solutions which the current solution dominates. The fitness is computed according to the strength of all non-dominated external solutions that dominate the current solution. SPEA2 [3] improves its previous version by employing an enhanced fitness assignment strategy as well as new techniques for archive truncation and density-based selection.

Pseudocode 3. *SPEA2 algorithm*

1: Generate initial population of size *N*

2: Create an archive

3: **for i = 1 to number_of_generations**

4: Compute fitness of individuals from population and archive

5: Copy non-dominated individuals from population to archive

6: Use the reduction operator to remove elements from archive when capacity is exceeded

7: If capacity is not exceeded use dominatedsolutions frompopulation to fill archive

8: Perform selection on combined individuals from population and archive

9: Apply crossover and mutation

10: **end for**

5.2.3 The Fast Pareto Genetic Algorithm

The FPGA algorithm [4] is a GA that uses a new ranking strategy. New operators are used to improve algorithm's convergence and to reduce the computational effort. Unlike the previous algorithms, which use a fixed value for the population's size, this algorithm dynamically adapts the size of the population at each iteration using a regulation operator. In this work, our results obtained by applying FPGA showed better performance than using NSGA-II, in terms of fast convergence to real Pareto front while maintaining a uniform distribution of non-dominated solutions.

Pseudocode 4. *FPGA algorithm*

1: t = 0

2:Create a random population P_t

3: Evaluate (P_t)

4: **While (stop condition it is not fulfilled)**

5: t++

6: P_t= selection(P_t-1)

7: O_t= crossover (P_t)

8: O_t=mutation (O_t)

9: Evaluate (O_t)

10: $CP_t = P_t \, U \, O_t$ // composite population

11: Rank(CP_t) // assign rank to each individual

12: Regulate(CP_t)

13: P_t= Generate(CP_t) // Generate new population

14: **end while**

5.3 Swarm-intelligence-based techniques

5.3.1 Speed constrained Multi-objective Particle Swarm Optimization (SMPSO)

The Particle Swarm Optimization (PSO) technique is inspired from the social behaviour of living beings, like birds, fishes, bees [25]. The search is performed by a set of particles that move using different speeds and change according to the entire system's characteristics. The set of particles is called swarm. Particles move through the search space following the '*best*' particles at that moment. To achieve this, each particle changes position and speed according to the best particle (leader) and its own best position (history). After particle's movement, the swarm is re-evaluated and new leaders are chosen. PSO enables quick finding optimum but have difficulty in avoiding local minima. Somewhat similar to GA, SMPSO [5], [26] uses an archive to store the non-dominated solutions found in the search process and a density estimator, crowding distance, for selection. Unlike GA, SMPSO does not need crossover, only mutation.

Pseudocode 5. *SMPSO algorithm*

1: Initialize swarm

2: Initialize leaders archive

3: i = 0

4: **while i<max_iterations**

5: ComputeSpeed()

6: updatePosition()

7: mutation()

8: evaluate(swarm)

9: update archive

10: update particles history

11: i++

12: **end while**

13: **return leaders**

6. Optimization Procedure

First, the suspension model is initialized. The body mass, wheel mass, tire stiffness and damping are fixed parameters and will not change during the simulation. Then, the design variables are defined, the spring stiffness K_s and the damping coefficient C_s, which are represented as real numbers. The bounds of the variables are also set in this step. The next step is to establish the design constraints. For this paper we used three constraints which are related to the maximum acceleration of the sprung mass, the relative displacement of the masses and the suspension's natural frequency. The optimization's objectives are to minimize the sprung mass acceleration and the sprung mass displacement. Now an optimization algorithm is chosen and the optimization process is started.

The algorithms used in this paper belong to the evolutionary algorithms class. All of them will follow the same basic steps. An initial set of solutions called population is generated randomly. Then, an optimization loop will use the initial population in order to generate better solutions (new populations), using specific evolutionary operators. In this loop, the objective functions are evaluated, for a particular case in which the suspension's mathematical model is solved. Based on this evaluation, a *fitness* is assigned to each solution in the population. The fitness represents the quality or the chance of the survival on an individual. After the evaluation the best candidates are chosen which will reproduce to create a new population. The selection, crossover and mutation operators are used to achieve this process. When a new population is obtained, the optimization loop will execute the same steps until a termination condition is satisfied. The algorithm output is a Pareto front, a set of solutions considered to be optimal. The entire procedure is synthesized in Fig. 4.

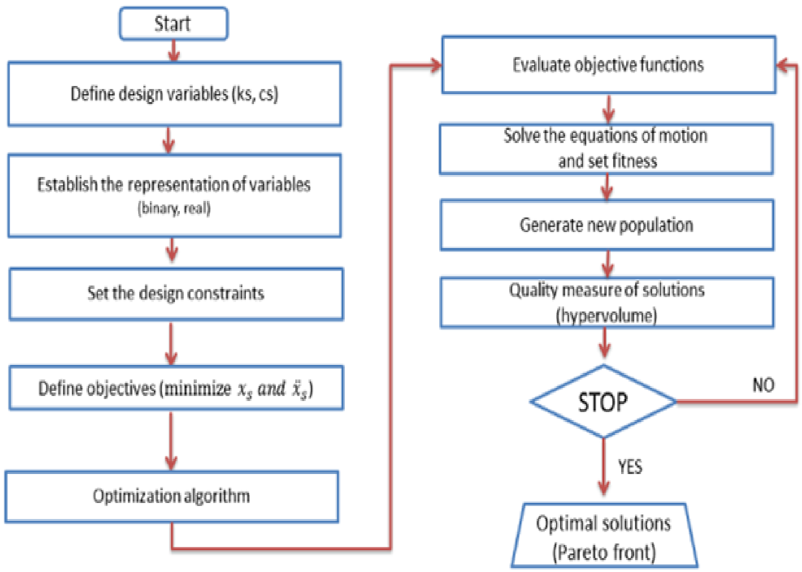

Figure 4. Optimization procedure

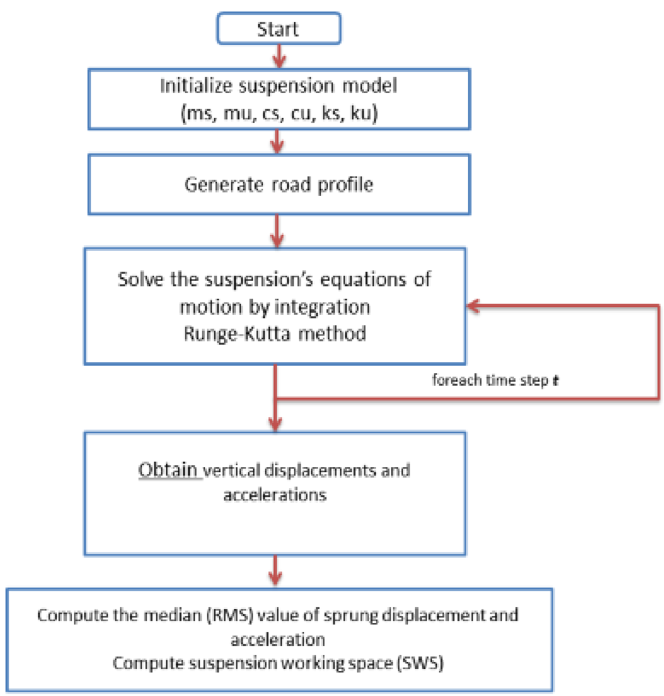

Figure 5a. Objective function (steps)

Of great importance are the objective functions used by the evolutionary algorithms to evaluate solutions. Fig. 5a shows the steps taken by such a function. The RMS value represents the Root Mean Square computed as:

$$RMS = \frac{1}{n}\sqrt{(x_1^{\,2} + x_2^{\,2} + x_n^{\,2})} \qquad (8)$$

7. Application Graphical user Interface

Our application was written in C# language under Visual Studio 2012 environment.The software solution consists in two basic projects: EVAFramework and jMetal.NET. The first contains the application graphical user interface (GUI) which allows selecting the algorithm's configuration class, the optimization problem, and facilitates viewing and comparing the obtained results and viewing the values of individuals and of objectives. The second project's goal was to provide C# implementation of jMetal (a Java framework for multi-objective optimization with metaheuristics) [28] by porting this library on .NET platform. It contains a subset of algorithms and the way they are setup, a number of problems and their subsets of solution representations, and the genetic operators from original library. For a proper run of applicationis required a performant PC, such as quad-core 2.4 GHz Intel P4 Xeon with 4GB RAM memory, Windows XP SP2 operating system or newer and 4.5 version of .NET Framework.

To start the application is launched the binary file EvaFramework.exe. Then on full screen is displayed a user-friendly GUI (see fig. 5b). This consists in: the main menu, an algorithm control panel, an optimization problem configuration control panel, a results list, an implemented metrics list, a panel designed for plotting the selected road quality.From the algorithm configuration panel (left side) the user may run the following actions:

• Selecting the optimization mode: mono / multi objective.

• Establishing the evaluation strategy of solutions: single-core, multi-core or distributed. In single-core assessment strategy the algorithm is performed just on one core. In multi-core mode,there are involved in processing all processor cores made available by the host station. In distributed way the evaluation is done on multiple workstations taking into account on their cores number.

• Choosing the optimization problem

• Setting the parameters of optimization (the population size, number of generations, the values of genetic operators,etc).

• Start/Stop the optimization procedure.

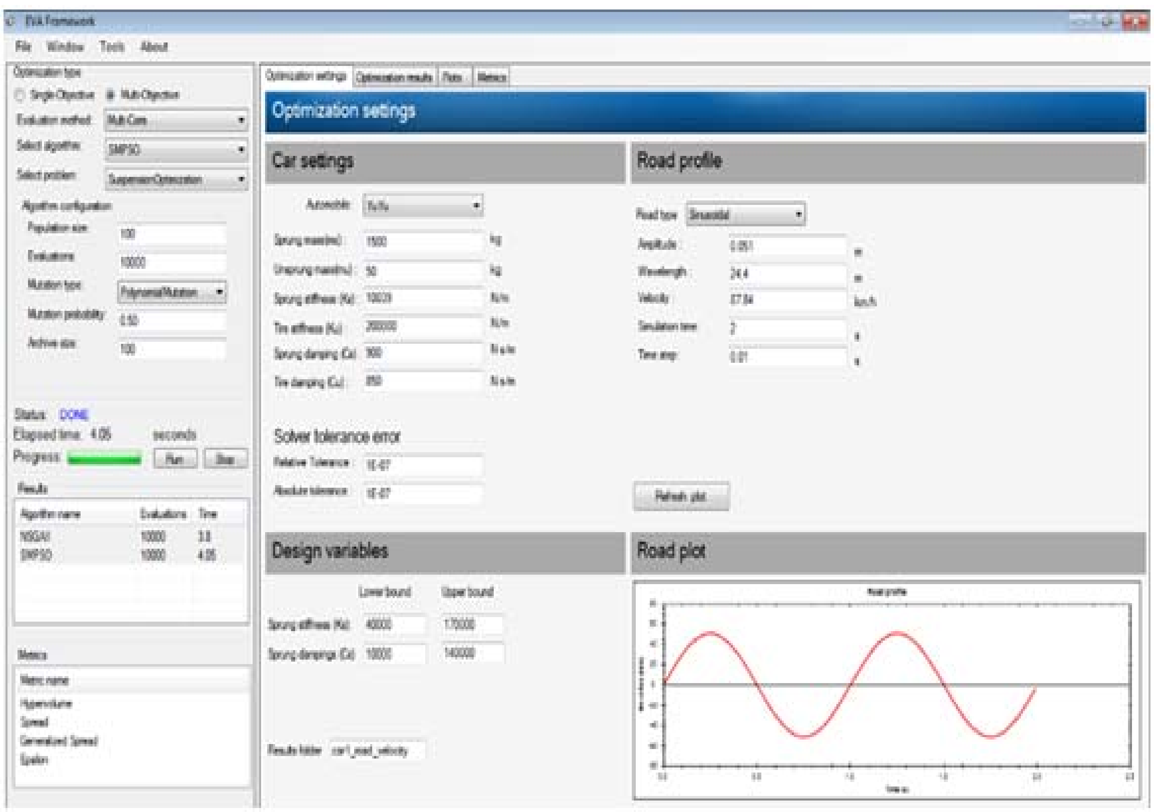

Figure 5b. Application's GUI

8. Results

In order to determine the optimum parameters of suspensions our developed software application usesNewtonsoft.JSON and ZedGraph packages. These, serialized the information in human-readable format allowing further 2D graphical representation or creating Excel worksheet, for plotting road profile or other operations. Solving the problem of optimizing the suspension has determined us to software implement a platform capable of effectively solving a variety of mono- and multi-objective optimization problems using metaheuristics methods. Our platform achieves the following objectives:

• Efficiency - the software must use efficiently the machine computing resources.

• Flexibility - there are many algorithms and running configurations.

• Extensibility - new problems and algorithms may be included for testing with and this requires minimal effort from the user. This is possible due to the object-oriented programming concepts on which our application is based.

• Scalability - the software application is able to tailor itself at a growing volume of work.

In order to reduce the simulation time we implemented such as effectively exploits the computer resources by taking advantage of multicore architectures. In addition, it offers the possibility to distribute tasks across multiple workstations and is very useful for complex problems that require additional processing time. For solving the system of differential equations we used the *.DotNumerics* library [27] and, for implementing the optimization algorithms, we extended the algorithms from jMetal.NET library [28].

The optimization was done by simulating a QCM traversing two road profiles with different degree of roughness. The parameters of the model are presented in Table 2 and Table 3.

Quarter-car model	
Sprung mass (m_s)	1500 kg
Unsprang mass (m_u)	50 kg
Tire stiffness (K_s)	200000 N/m
Tire damping (K_u)	850 N×sec/m

Table 2. Quarter-car parameters

For the first experiment, we generated a road profile called AB (Fig. 3) considered a very good one according to the formulas presented in section 3. The road parameters are presented in Table 4. The profile has a maximum elevation of 15mm.

	Lower bound	Upper bound
Ks	30000	170000
Cs	10000	140000

Table 3. Design variables limits

Roughness value	$2 * 10^{-6}$ m³
Road length	100 m
Sampling interval	0.1 m
Wavelength interval	0.3 – 90 m

Table 4. AB road profile parameters

Using this profile we tested a vehicle suspension system travelling at two different speeds, 20 and 80 km/h. In this way, we can see the suspension's behaviour at low and high speeds. The settings used for the optimization algorithms are shown in Table 5. The population was set to 100 individuals and the number of evaluations was set to 5000.

Fig.6 shows the solutions obtained by each of the algorithm tested. We can see that after 5000 evaluations almost all algorithms found solutions on approximately the same front. As expected, VEGA and classic GA did not perform as good as the other algorithms.

In order to compare the performance of the algorithms, we used the Hypervolume quality indicator, which measures the area of coverage made by a Pareto front and a reference point. From Fig.7 we can see that after about 1000 evaluations all algorithms start to converge. Of course, the small number of evaluations needed to find an "optimal" solution is due the simplistic model used in this paper. More complex models with higher degrees of

	NSGAII	SMPSO	SPEA2	FPGA	VEGA	GA
Population	100	100	100	100	100	100
Archive	-	100	100	-	-	-
Mutation probability	1/number_of_variables					
Crossover probability	0.9	-	0.9	0.9	0.9	0.9
Evaluation				5000		

Table 5. Optimization algorithm's parameters

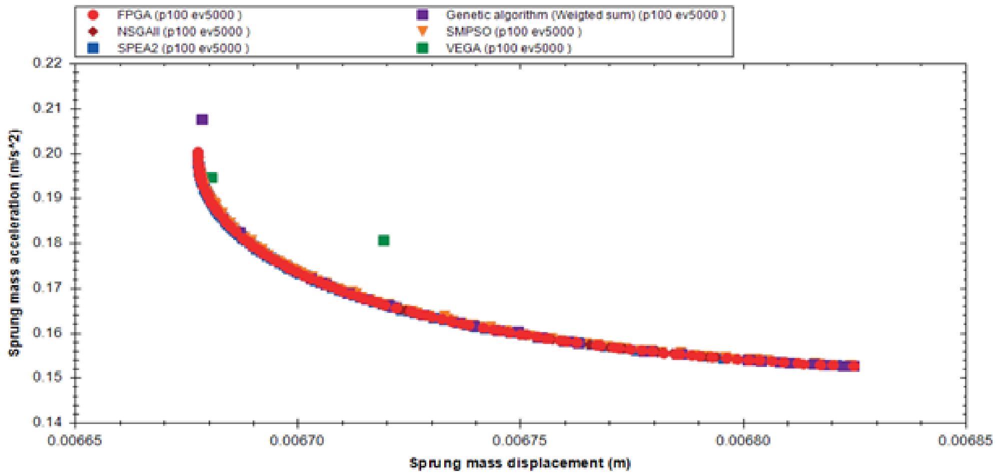

Figure 6. Pareto fronts for AB profile

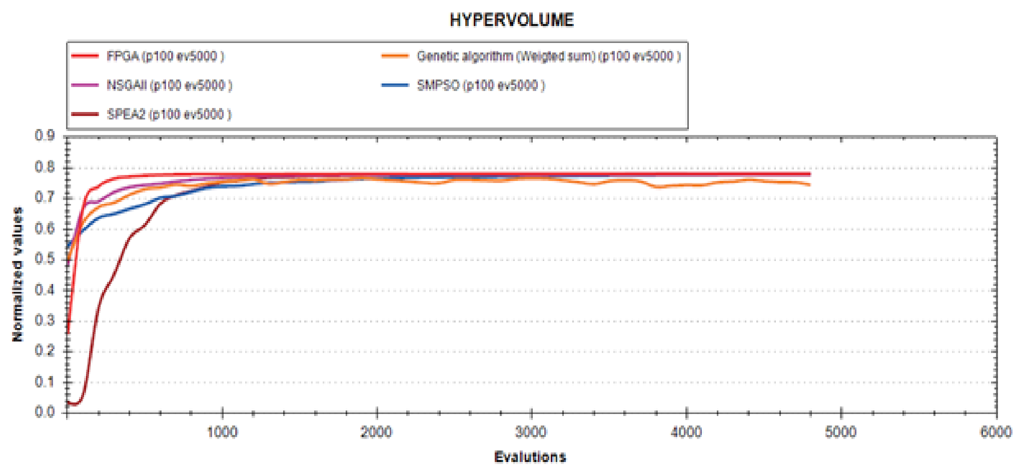

Figure 7. Hypervolume (AB road profile, 20 km/h)

freedom will certainly require more generations to run. Another important observation is that FPGA algorithm has a faster convergence rate and slightly rises above the others.

Because in multi-objective optimization we obtain a set of possible solutions and not a unique solution for every scenario tested, we chose randomly one solution from each algorithm's output and simulate the QCM using that solution Table 6. A graphical representation for the sprung mass displacement and acceleration for one of this solution are shown in Figs.8(a) and 8(b).

Algorithm	Ks	Cs	Peak Acc. (M/m^2)	Peak Disp.(m)	RMS Acc. (N/m^2)	RMS Disp.(m)	Natural freqeumcy
NSGA-II	84196.3	14460.8	0.45	0.013	0.156	0.006	1.00
FPGA	84132.5	13392.9	0.41	0.015	0.137	0.005	1.00
SPEA2	84145.3	13565.8	0.41	0.015	0.137	0.005	1.00
SMPSO	84238.6	14303.8	0.455	0.015	0.155	0.006	1.00
GA(ws)	84148.2	14809.4	0.464	0.015	0.157	0.006	1.00
VEGA	86219.9	24798.2	0.576	0.016	0.169	0.005	1.00

Table 6. Solutions for AB road profile at 20 km/h

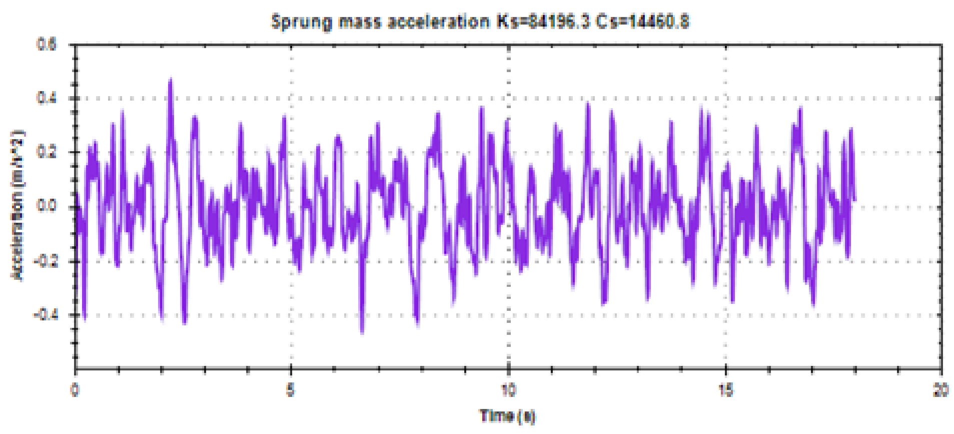

Figure 8(a). Sprung mass acceleration (AB profile – 20 km/h)

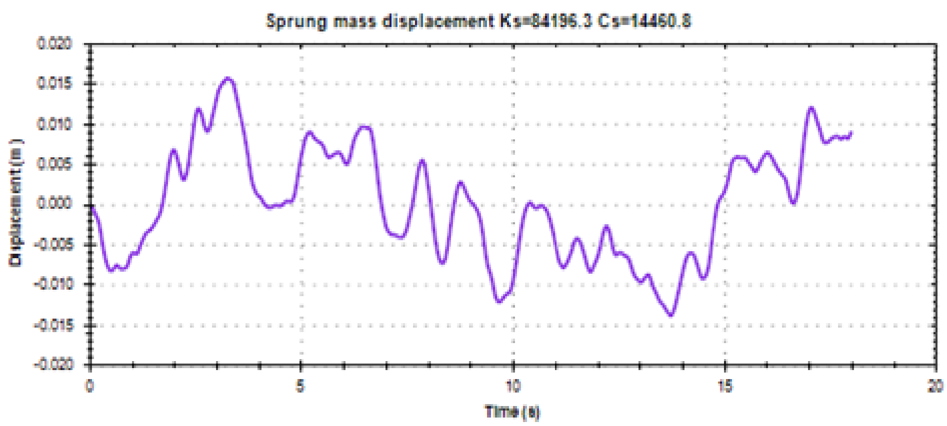

Figure 8(b). Sprung mass displacement (AB profile – 20 km/h)

Now, we change the vehicle's velocity to 80 km/h. The road profile and all parameters remain the same. Below are the obtained results. Fig.9 presents the Pareto fronts of all algorithms after 5000 evaluations. Fig. 10 shows the hypervolume computed for this fronts and Table 7 contains one solution chosen randomly from the set of possible solutions.

For our second experiment, we generated a rougher road profile (CD) as the one from Fig. 12. The only parameter that changed was the value of roughness which was set to $32*10^{-6}$ m. The maximum elevation in this case is 63 mm.Results obtained for this profile are shown below. The algorithms behave in the same way as in the previous experiment, finding similar solutions except VEGA which is way off. The hypervolume (Fig. 14) shows the convergence of the algorithms after 1000 evaluations, with FPGA having better results.

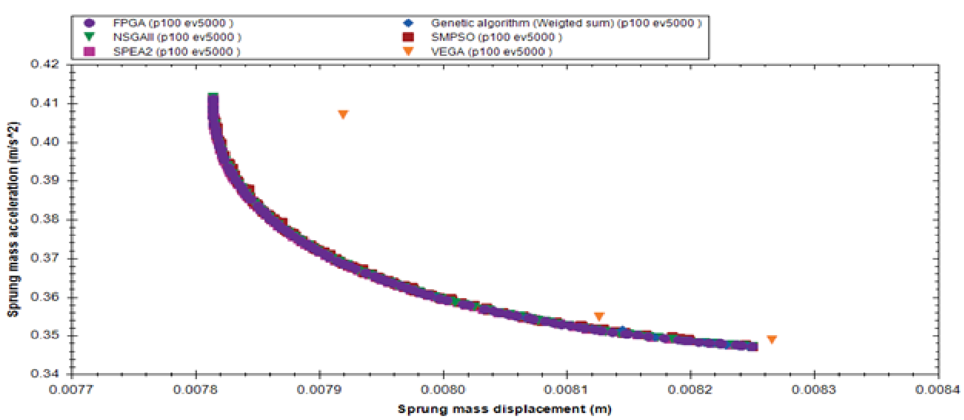

Figure 9. Pareto Fronts for AB road profile at 80 km/h

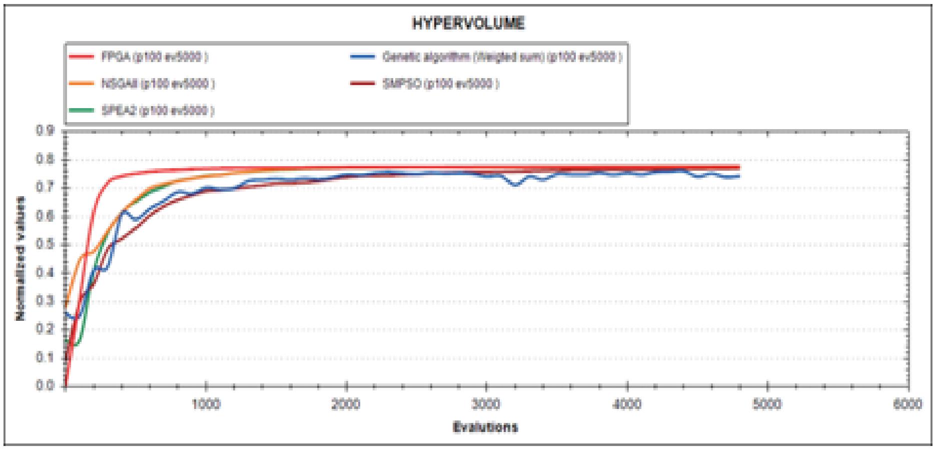

Figure 10. Hypervolume for AB road profile at 80km/h

	Ks	C	Peak Acc.(N/m^2)	Peak Disp.(m)	RMS Acc. (N/m^2)	RMS Disp.(m)	Natural frequency
NSGA-II	84135.5	11776.7	1.002	0.020	0.368	0.008	1.00
FPGA	84139.9	11956.7	0.94	0.017	0.325	0.009	1.00
SPEA2	84129	13348.3	0.98	0.017	0.34	0.009	1.00
SMPSO	84239.4	11733.5	1.002	0.020	0.368	0.008	1.00
GA(ws)	84127.9	11632.3	1.002	1.002	0.368	0.008	1.00
VEGA	85266.0	11619.2	0.94	0.017	0.323	0.009	1.00

Table 7. Solutions for AB road profile at 80km/h

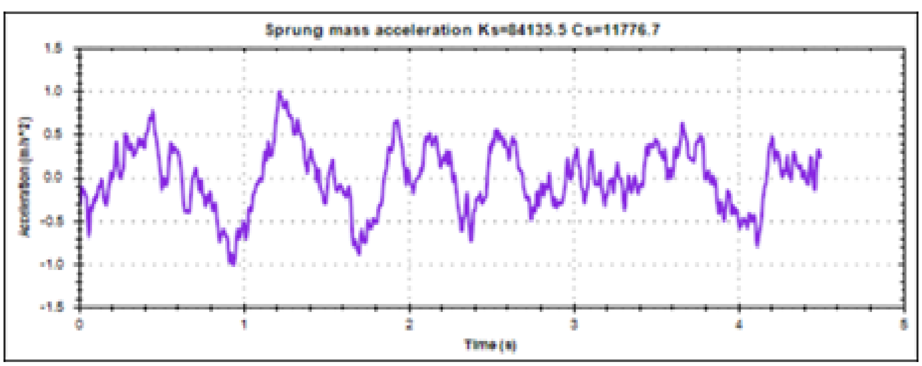

Figure 11(a). Sprung mass acceleration AB profile at 80km/h

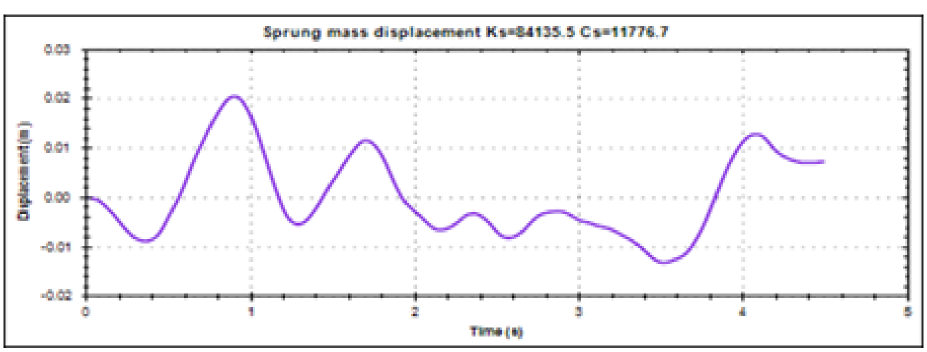

Figure 11(b). Sprung mass displacement AB profile at 80km/h

For our second experiment, we generated a rougher road profile (CD) as the one from Fig. 12. The only parameter that changed was the value of roughness which was set to $32*10^{-6}$ m. The maximum elevation in this case is 63 mm.Results obtained for this profile are shown below. The algorithms behave in the same way as in the previous experiment, finding similar solutions except VEGA which is way off. The hypervolume (Fig. 14) shows the convergence of the algorithms after 1000 evaluations, with FPGA having better results.

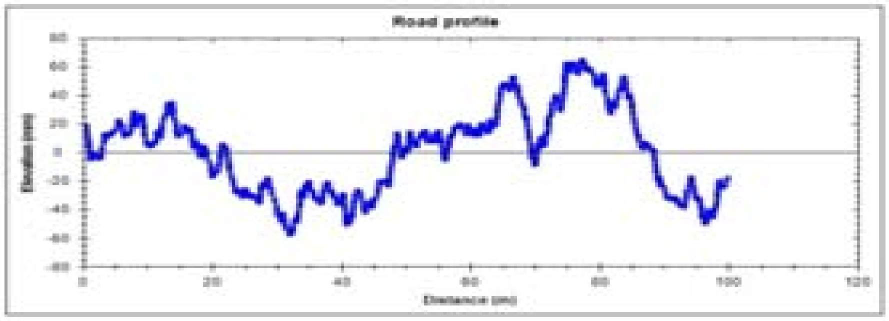

Figure 12. Road profile of type CD

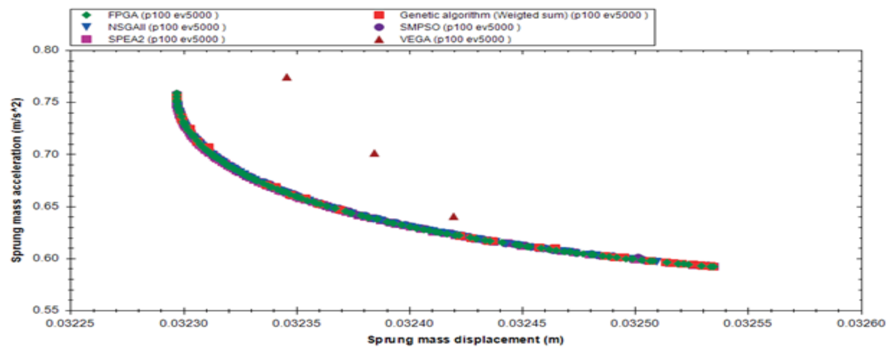

Figure 13. Pareto Fronts for CD road profile at 20 km/h

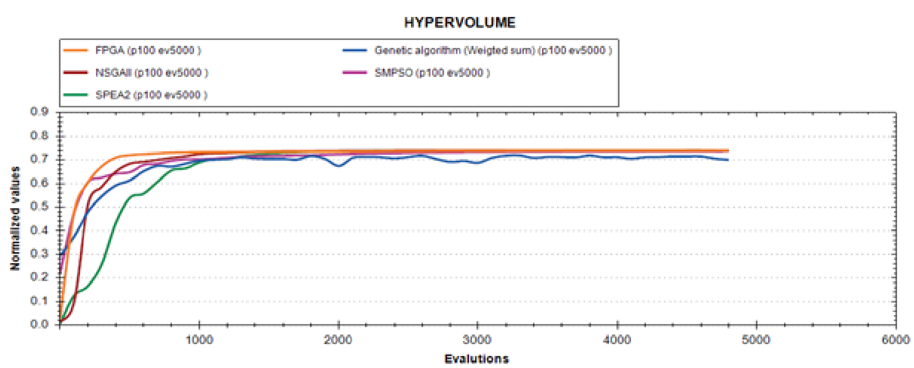

Figure 14. Hypervolume for CD road profile at 20 km/h

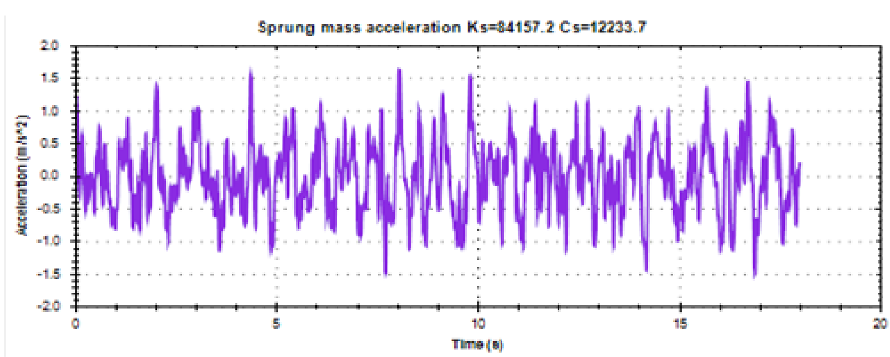

Figure 15. Sprung mass acceleration CD profile at 20km/h

	Ks	Cs	Peak Acc.	Peak Disp.(m) (N/m²)	RMS Acc. (N/m²)	RMS Disp.(m)	Natural frequency
NSGA-II	84157.2	12233.6	1.621	0.054	0.545	0.026	1.00
FPGA	84141.1	12016.1	2.037	0.062	0.631	0.024	1.00
SPEA2	84126.5	11359.8	2.026	0.062	0.622	0.024	1.00
SMPSO	84244.6	11300.1	1.560	0.055	0.532	0.026	1.00
GA(ws)	84401.3	10679.6	1.51	0.055	0.524	0.026	1.001
VEGA	96727.2	16228.5	2.252	0.063	0.707	0.024	1.049

Table 8. Solutions for CD road profile at 20km/h

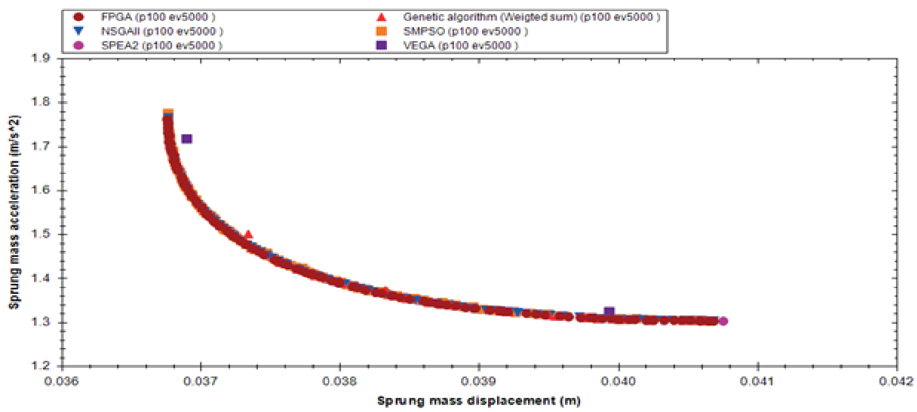

Figure 16. Pareto Fronts for CD road profile at 80 km/h

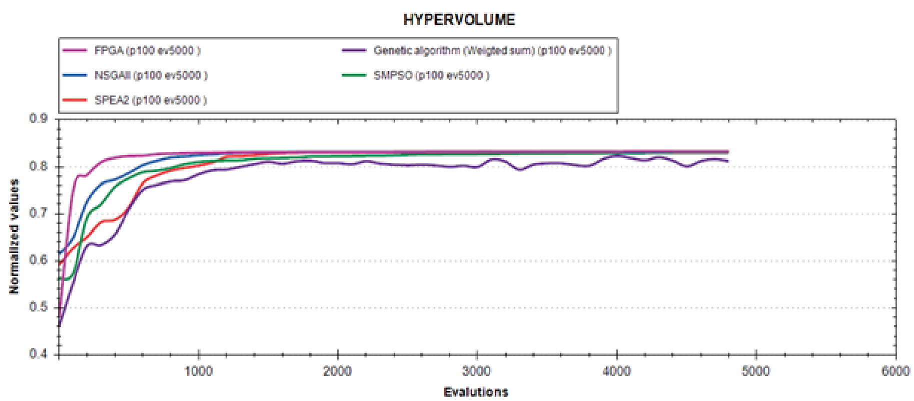

Figure 17. Hypervolume for CD road profile at 80km/h

	Ks	Cs	Peak Acc.	Peak Disp.(m) (N/m²)	RMS Acc. (N/m²)	RMS Disp.(m)	Natural frequency
NSGA-II	84157.1	13429.8	3.418	0.062	1.291	0.032	1.00
FPGA	84237.5	13225.1	3.313	0.061	1.303	0.031	1.00
SPEA2	84221.2	14343.3	3.521	0.062	1.315	0.031	1.00
SMPSO	84652.1	14628.3	3.565	0.061	1.319	0.031	1.002
GA(ws)	84184.3	14628.8	3.501	0.061	1.310	0.031	1.00
VEGA	86684	14523.5	3.666	0.064	1.351	0.032	1.04

Table 9. Solutions for CD road profile at 80km/h

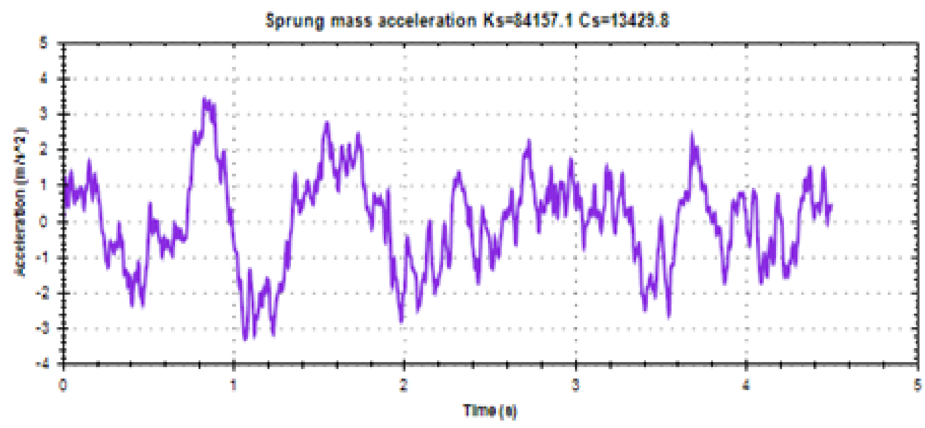

Figure 18. Sprung mass acceleration CD profile at 80km/h

The vehicle's speed is set to 80 km/h and road profile is CD. All the other parameters remain the same. As in previous experiments, the Pareto fronts (Figure 16),hypervolume (Figure 17) and one solution from each algorithm (Table 9) are presented.

Comparing Table 6 with Table 7, Table 8 and Table 9 results that as the speed increase and as the road became worse, the Peak and RMS acceleration and displacements will increase decreasing the ride comfort.

9. Conclusions

Stochastic methods such as evolutionary algorithms proved to be useful in the problem of suspension optimization. A QCM with two degree-of-freedom was used and its performance was evaluated in time domain. The model was excited by random road profiles with different degree of roughness. The first profile generated was considered to be a very good one and the second a more difficult one. For each type of road, the suspension model was tested using a constant speed of 20 km/h and 80 km/h. As expected, travelling at high speeds showed a significant increase of the sprung mass acceleration, while a road with high degree of roughness making it much worse.

With the help of this software application, we will improve the quality of spare parts production that are required by the suspensions mechanism through focusing on specific directions which depends on the geographic area, the infrastructure of the region, environmental conditions, characteristics of fuels, etc. Due to vehicle's operating conditions, the manufacturers should tailor the components of the suspensions mechanism and differentiate the period in which they perform maintenance revision, depending by country, geographic area, etc.

The optimization algorithms NSGA-II, FPGA, SPEA2 and SMPSO managed to obtain better results in terms of fast convergence and solution diversity than the classical methods such as VEGA and GA with objective aggregation, which do not truly benefit from the concept of Pareto optimality. Performance comparison revealed that FPGA algorithm obtained slightly better results than NSGA-II, being a good choice when only a small number of iterations can be executed. For further work, we are concerned to improve the suspensions system stiffness and damping by optimizing the coefficients with the help of fuzzy logic techniques and expand research on semi-active and active suspension systems. In addition, we will extend our study on models of suspensions having more degrees of freedom (4-DOF, etc).

References

[1] Deb, Kalyanmoy (2001) Multi-Objective Optimization Using Evolutionary Algorithms (West Sussex: John Wiley & Sons).

[2] Deb, Kalyanmoy, AmritPratap, Sameer Agarwal, and T. Meyariva (2002),A Fast and Elitist Multiobjective GeneticAlgorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation,* 6 (2) 182 – 197.

[3] Zitzler, Eckart, Marco Laumanns and Lothar Thiele (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, International Center for Numerical Methods in Engineering, Athens, Greece, p. 95-100.

[4] Eskandari, Hamidreza., Christopher, D. Geiger (2008) Solving Expensive Multiobjective Optimization Problems: A Fast Pareto Genetic Algorithm Approach, *Journal of Heuristics*, 14 (3) 203-241.

[5] Cagnina, Leticia., Susana Esquivel (2005) A Particle Swarm Optimizer for Multi-Objective Optimization, *Journal of Computer Science and Technology,* 4, p. 204-210.

[6] Haug, Edward and Jasbir Arora (1979) Applied Optimal Design: Mechanical and Structural Systems, (West Sussex: John Wiley & Sons).

[7] Sawant, S. H., Tamboli, J.A. (2013) Experimental Verification of Passive Quarter Car Vehicle Dynamic System Subjected to Harmonic Road Excitation with

Nonlinear Parameters, *Journal of Mechanical and Civil Engineering* 3, p. 39-45.

[8] Yu, Hong Biao and Nan Yu (1998) Application of Genetic Algorithms to Vehicle Suspension Design, Computer Methods in Applied Mechanics and Engineering, 163, p. 87–94.

[9] Anirban, C. M., Gourav. J.D., Saaish. R. P., Parag H. S., Waseem M.H.K., Nilotpal B., Optimization of Passive Vehicle Suspension System by Genetic Algorithm, Procedia Engineering, Volume 144, 2016, p. 1158–1166.

[10] Elsawaf, A., Vampola, T., Passive Suspension System Optimization Using PSO to Enhance Ride Comfort when Crossing Different Types of Speed Control Profiles, *Journal of Traffic and Logistics Engineering* 3 (2) December 2015.

[11] Šagi G., Luliæ Z., Ilinèiæ P. (2015). Multi-objective optimization model in the vehicle suspension system development process, *Tehnièkivjesnik* 22 (4) p. 1021-1028.

[12] Badran, Saeed and Ashraf Salah (2012) Design of Optimal Linear Suspension for Quarter Car with Human Model using Genetic Algorithms, *Research Bulletin of Jordan*, 2, p. 42-51.

[13] Jazar, Reza (2008) Vehicle Dynamics (New York: Springer Science and Business Media).

[14] Wong, Jo Yung (2001) Theory of Ground Vehicles (New York: John Wiley & Sons).

[15] Roman, Lucian, Adrian Florea and Ileana Cofaru (2014) Mathematical Model and Software Simulation of Suspension's System from Opel Cars, *Annals of the Oradea University Fascicle of Management and Technological Engineering* 23 (3) 94-99.

[16] New, Tomas Michael (2008) Random Road Analysis and Improved Gear Ratio Selection of a Front Wheel Drive Drag Racing Car, Master of Science, Clemson University.

[17] Verris, G., Natsiavas, S (2005) Design Optimization of Quarter-car Models with Passive and Semi-active Suspensions under Random Road Excitation, *Journal of Vibration and Control*, p. 581-606.

[18] Melcer, Jozef, and Gabriela Lajèáková (2012) Numerical Simulation of Vehicle Motion along the Road Structure, Rocznikiinzynieriibudowlanej, p. 37-42.

[19] Múèka, Peter (2012) Longitudinal Road Profile Spectrum Approximation by Split Straight Lines, J*ournal of Transportation Engineering*, 138 (2). 243-25.

[20] Agostinacchio, M., D. Ciampa , S. Olita (2013) The Vibrations Induced by Surface Irregularities in Road Pavements - a Matlab approach, *European Transport Research Review*, 6 (3) p. 267-275.

[21] Coello, Carlos A., Gary B. Lamont, and David A. Van Veldhuizen (2007) *Evolutionary Algorithms for Solving Multi-Objective Problems* (New York: Springer Science+Business Media).

[22] Gen, Mitsuo and Runwei Cheng (2000). Genetic Algorithms and Engineering Optimization (West Sussex: John Wiley & Sons).

[23] Schaffer, David (1984) Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, PhD Thesis, Vanderbilt University, Nashville, Tennessee.

[24] Zitzler, Eckart and Lothar Thiele (1999) Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, *IEEE Transactions on Evolutionary Computation,* 3, p. 257 – 271.

[25] Eberhart, Russell and James Kennedy (1995) A new optimizer using particle swarm theory, *In:* Proceedings of the sixth IEEE International Symposium on Micro Machine and Human Science (43) New York, USA, p. 39-43.

[26] Nebro, Antonio, Juan Durillo, Jose Garcia-Nieto, C.A. CoelloCoello, Francisco Luna, and Enrique Alba (2009) SMPSO: A new PSO-based metaheuristic for multi-objective optimization, *In:*IEEE Symposium on Computational Intelligence in Multi-criteria Decision-making, p. 66-73.

[27] Santiago-Castillo Jose Antonio (2015) DotNumerics, http://www.dotnumerics.com/NumericalLibraries/DifferentialEquations/Default.aspx, (accessed 25 March 2015).

[28] Durillo, Juan., Nebro, Antonio ., Alba, Enrique (2010). The Metal Framework for Multi-Objective Optimization: Design and Architecture, *Congress of Evolutionary Computing*, p. 1-8.