View metadata, citation and similar papers at core.ac.uk brought to you by CORE

ISSN 1330-3651(Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20151021202802

# A MODIFIED PARTICLE SWARM OPTIMIZATION ALGORITHM FOR THE OPTIMIZATION OF A FUZZY CLASSIFICATION SUBSYSTEM IN A SERIES HYBRID ELECTRIC VEHICLE

*Zsolt Csaba Johanyák*

Original scientific paper

Particle swarm optimization (PSO) based optimization algorithms are simple and easily implementable techniques with low computational complexity, which makes them good tools for solving large-scale nonlinear optimization problems. This paper presents a modified version of the original method by combining PSO with a local search technique at the end of each iteration cycle. The new algorithm is applied for the task of parameter optimization of a fuzzy classification subsystem in a series hybrid electric vehicle (SHEV) aiming at the reduction of the harmful pollutant emission. The new method ensured a better fitness value than either the original PSO algorithm or the clonal selection based artificial immune system algorithm (CLONALG) by using similar parameters.

***Keywords:*** *classification; fuzzy logic; hybrid electric vehicle; particle swarm optimization*

## Modificirani algoritam optimizacije roja čestica za optimizaciju fuzzy podsustava klasifikacije u serijskom hibridnom električnom vozilu

Izvorni znanstveni članak

Algoritmi optimizacije temeljeni na optimizaciji roja čestica (PSO - particle swarm optimization) su jednostavne i lako primjenljive metode male računske složenosti što ih čini podesnim alatom za rješavanje opsežnih nelinearnih problema optimizacije. U radu je prikazana modificirana verzija originalne metode kombiniranjem PSO i lokalne metode pretraživanja na kraju svakog ciklusa ponavljanja. Novi se algoritam primjenjuje u zadatku optimizacije parametara podsustava fuzzy klasifikacije u serijskom hibridnom električnom vozilu u cilju smanjenja ispuštanja štetnih zagađivača. Novom se metodom uz primjenu sličnih parametara osigurava vrijednost veće prikladnosti nego bilo s originalnim PSO algoritmom ili algoritmom umjetnog imunološkog sustava zasnovanog na klonskoj selekciji (CLONALG).

***Ključne riječi:*** *fuzzy logika; hibridno električno vozilo; klasifikacija; optimizacija roja čestica*

## 1 Introduction

Nature inspired algorithms have been widely successfully applied for optimization problems in the last three centuries. Particle Swarm Optimization (PSO) is one of the most popular of them because it is a simple and easily implementable method with low computational complexity. Besides, it also maintains the "memory" of the best solutions [1]. Its original version was developed by Kennedy and Ebenhart [2] in 1995. PSO was inspired by the social behaviour of bird flocking and fish schooling.

Although in the course of years PSO has been applied successfully for many problems [3], several modifications and enhancements have been proposed so far. Richards and Ventura [4] introduced a new method for the initial distribution of the particles. Campana et al. suggested a new mode for the definition of the particle’s trajectories to get a better exploration of the search space. Kennedy and Mendes [5] proposed the usage of a fully informed particle swarm where the velocity vector is determined based on the performance of each neighbour of a particle. Fan and Shi [6] demonstrated that the performance of the original method can be improved by proper dynamic change of $v_{\max}$. Clerc and Kennedy [7] proposed a constriction coefficient to improve the convergence of the particles. Shi et al. [8] proposed the use of an extra parameter called inertia weight for the velocity calculation. Chen et al. [9] suggested the application of an adaptive inertia weight. Pehlivanoglu [10] introduced a new mutation strategy to avoid the convergence of the particles around a local optimum solution. Campana et al. [11] reformulated the standard PSO iteration into a linear dynamic system in order to provide a better positioning of the initial particles. Precup et al. [12] hybridized PSO with the Gravitational Search Algorithm for the optimal tuning of Takagi-Sugeno-Kang PI-fuzzy controllers.

In this paper, we introduce a modified (hybridized) version of PSO where we combine the original method with a simple local search technique. The modified algorithm is applied for the task of parameter optimization of a fuzzy classification subsystem in a series hybrid electric vehicle (SHEV) aiming at the reduction of the harmful pollutant emission. The SHEV system is modelled in Simulink. The rest of this paper is organized as follows. The new hybrid algorithm is presented in Section 2. The main components of the used SHEV and the fuzzy classification subsystem are overviewed briefly in Section 3. Section 4 presents the applied objective (cost) function and the simulation results. The conclusions are drawn in section 5.

## 2 The modified PSO algorithm

In this section we are going to present a modified PSO algorithm, which was used for the optimization of some parameters of the fuzzy system presented in Section 3. The novelty of the proposed method compared to the original PSO is that it applies a local search technique at the end of each iteration cycle.

PSO is a population based heuristic optimization algorithm. Its goal is to find the optimal values for the elements of a feature vector

$$P = (p_k),\ k = \overline{1,n}, \tag{1}$$

where $n$ is the number of the parameters/properties to be optimized. The optimization problem can be defined as

$$P_{\text{opt}} = \underset{P \geq LB,\, P \leq UB}{\text{argmin}} \left(F\right), \tag{2}$$

where $LB$ and $UB$ are vectors with $n$ elements containing the lower and upper bounds for each parameter, and $F$(.) is the fitness (objective) function, which will be presented in details in Section 4.1.

The particles (members of the swarm) can be characterized by their position in the search space and the corresponding fitness value. Therefore, in case of PSO based methods the feature vector $\boldsymbol{P}$ is called position vector. The initial distribution of the particles usually can be defined randomly. However, a guided initialization is also possible when some a-priory information is available. The particles change their position in each iteration cycle, by exploring new areas of the search space. The distance and the exact direction of movement is defined by a so called velocity vector, which is calculated based on the "memory" of the swarm, i.e. the previous value of the velocity vector, the so far best position of the given particle (personal best) and the so far best position taken into consideration of the whole swarm (overall best). The velocity vector of the particle $j$ in the iteration cycle $i$ is

$$v_{ij} = c_3 \cdot v_{i-1,j} + c_1 \cdot r_1 \left(PB_j - P_{i-1,j}\right) + c_2 \cdot r_2 \left(OB_{i-1} - P_{i-1,j}\right), \tag{3}$$

where $PB_j$ is the personal best position of the $j^{\text{th}}$ particle, $OB_{i-1}$ is the overall best position in the $i-1^{\text{th}}$ iteration cycle, $c_1$ and $c_2 \in [0,1]$ are constant parameters of the method; $r_1$ and $r_2 \in [0,1]$ are random values; and $P_{i-1,j}$ is the position vector of the $j^{\text{th}}$ particle in the previous iteration cycle. The value of $c_3$ is defined usually as an arbitrary function of $c_1$ and $c_2$. For example

$$c_3 = 1 - c_1 - c_2. \tag{4}$$

The new position of the $j^{\text{th}}$ particle is calculated as

$$P_{ij} = \min\left(\max\left(P_{i-1,j} + v_{ij}, LB\right), UB\right). \tag{5}$$

In some cases, only integer values are acceptable as elements of the position vector. This demand can be fulfilled by applying a rounding operation

$$P_{ij} = round\left(P_{ij}\right). \tag{6}$$

In the new position, each particle is evaluated. Furthermore, the personal best and overall best vectors are also updated if necessary.

In order to improve the positions of the particles an iterative local search technique is employed for the first $n_B$ best particles at the end of each iteration cycle. Local search methods try to find better candidate solutions (particle position in our case) based on iterative exploration of the neighbourhood of an initial solution [13]. They apply local changes in each cycle and stop when either the prescribed maximal number of iteration or a desired performance level is reached.

The key idea of the local search technique applied in our case is that the local optima is approached step by step so that in the course of each step two new candidate positions are calculated for the current particle. After evaluating their performance, the particle will be placed in the position with the best performance, which can be either the old position or any of the new position candidates.

This concept is implemented in the following way. In each iteration cycle for each selected particle one examines and modifies each element of the position vector one-by-one. For the current element two new values are calculated

$$p_k' = p_k + s_k, \tag{7}$$

$$p_k'' = p_k - s_k, \tag{8}$$

$$S = \left(s_k\right),\ k = \overline{1,n} \tag{9}$$

where $s_k$ is a predefined step value for the $k$th element of the position vector. From the three available values that one is kept as the new coordinate value of the particle, which ensures the best fitness value

$$p_k = \text{argmin}_{p \in \{p_k, p_k', p_k''\}}\left(F(p)\right). \tag{10}$$

After each parameter modification the $PB$ and $OB$ vectors are updated if necessary. The number of iteration cycles ($n_l$) of the local search is a parameter of the method. Following the local search, a new iteration cycle of the whole method is started with the recalculation of the velocity vectors.

The default stopping criterion is the number of allowed iterations ($n_i$). The algorithm is described below.

```
i=1
REPEAT
  IF i==1 // In the first iteration cycle
    Create first particle randomly or guided
    Create N-1 particles randomly distributed in
the
      search space
    Define initial velocity vector for each particle
  ELSE // In each other iteration cycle
    Update velocity vector for each particle
    Update position of each particle
  END
  Calculate the performance of each particle
  Store personal best position and performance for
    each particle
  Store overall best position and performance
  Select the $n_B$ best particles
  l=1
  REPEAT // Start local search
    For each selected particle
      For each position vector element
        For both directions
          Determine new value (position)
          Calculate particle performance
          Update personal best position and
            performance if the current
            performance is better
          Update overall best position and
            performance if the current
```

```
                    performance is better
        l=l+1
    UNTIL l>n_l
    i=i+1
UNTIL i>n_i
```

Beside the number of allowed iterations other applicable stopping criteria can be considered as well. Such conditions could be:

$n_{fe}$ the number of fitness evaluations exceeds an upper limit – owing to the fact that the calculation of the fitness function is one of the computationally most expensive steps,

$n_i$ the number of iteration cycles exceeds an upper limit,

$F_{tr}$ the fitness of at least one particle becomes better than a threshold value,

$n_{wi}$ the number of consecutive generations without any improvement regarding the overall best fitness value exceeds an upper limit.

The parameters of the algorithm are summarized in Tab. 1.

**Table1** Parameters of the algorithm

| Param. | Explanation |
|---|---|
| $N$ | Number of particles in the swarm |
| $c_1$, $c_2$ | Constant parameters of the algorithm |
| $t_{nr}$ | Type of new value generation (1 – real values, 2 – integer values) |
| $t_s$ | Type of stopping criterion (1 – $n_{fe}$, 2 – $n_i$, 3 – $F_{tr}$, 4 – $n_{wi}$) |
| $v_s$ | Value for the stopping criterion |
| $n$ | Number of parameters to be optimized |
| $P_0$ | Predefined initial particle |
| $S=(s_k)$ | Vector containing the predefined step values |
| $n_B$ | Number of particles selected for local search |
| $n_l$ | Number of local search iteration cycles |
| $n_{fe}$ | Allowed number of fitness evaluations |
| $n_i$ | Allowed number of iteration cycles |
| $F_{tr}$ | Fitness threshold value |
| $n_{wi}$ | Allowed number of generations without improvement |
| $LB$, $UB$ | Lower and upper bounds for the parameters |

## 3 SHEV and fuzzy classification subsystem

The concept of hybrid electric vehicles (HEVs) emerged from the need for finding solution for the task of reducing the fuel consumption of the traditional internal combustion engine (ICE) based vehicles and to extend the drive range of the electric cars. In general, there are four main HEV types [14] depending on the links between the different components and the applied propulsion type. They are the series [15, 16], parallel [17, 18], series-parallel [19] and complex [20] HEVs.

The configuration and components of SHEVs are presented in Fig. 1, where *FT* is the fuel tank, *G* is the generator, *B* is the battery, *M* is the electric motor, *P* is the power converter, and *T* is the transmission system.

The thin line represents the hydraulic link, the double line indicates the mechanical link, and the thick line is used for the electrical link. In a SHEV there is no mechanical link between the ICE and the transmission system.

SHEVs' advantageous features are their simplicity; the freedom they give to the vehicle designer as well as the bigger efficiency compared to other configuration types, which is a result of the lower number of mechanical links [21, 22]. Practically the vehicle functions as an electric car, the ICE is used only for battery charge.

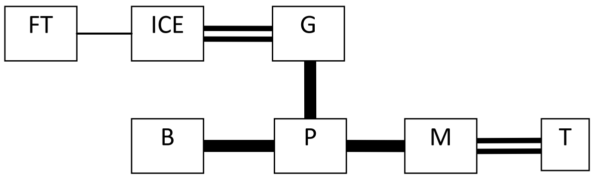

**Figure 1** Configuration and components of SHEVs

An important purpose of this research was to create a control structure that ensures low level harmful pollutant emission and low fuel consumption. To fulfil this task, a SHEV model [23] was created in Simulink.

Fuzzy concept based solutions are widely recognised as suitable tools for control, classification, and other purposes (e.g. [24÷30]). Therefore, the subsystem of the SHEV that aims at the control of the generator and the electric motor was developed with a fuzzy classification system in it. The task of the fuzzy subsystem is the determination of the desired charging state of the battery in function of the acceleration/deceleration needs, the energy stored in the battery, and the altitude change in the next section of the route.

The block diagram of the fuzzy subsystem is given in Fig. 2. The fuzzy subsystem works with the following three input signals.

*acc* a value belonging to the interval [0, 1] indicating the position of the gas pedal.

*brk* a value belonging to the interval [0, 1] indicating the position of the brake pedal.

*X* a vector with multiple components from which we use the indicator of the energy stored in the battery ($E_b$) and the altitude values for the next 500 m of the route ($s_j$).

The output of the fuzzy subsystem is the desired charging state. It has three possible values, which are used later by another subsystem. These values are

0 - no charge,
1 - slow charge,
2 - fast charge.

The fuzzy system works as a classifier with three inputs. Supposing that the driver does not press at the same time the gas and the brake pedals the two signals could be aggregated taking the brake signal with negative sign. The second input is the normalized (relative) value of the energy stored in the battery $E_{brel}$. The third input of the fuzzy controller is the relative value of the altitude change in the next 500 m of the route. We supposed that the altitude difference between the current point of the route and the point to which one can "see ahead" (500 m route distance) is not bigger than the mentioned route distance itself. Therefore, we used this distance for the normalization of the altitude difference. This input value indicates whether one can expect a positive (battery charging) or negative (battery depletion) effect of the altitude change.

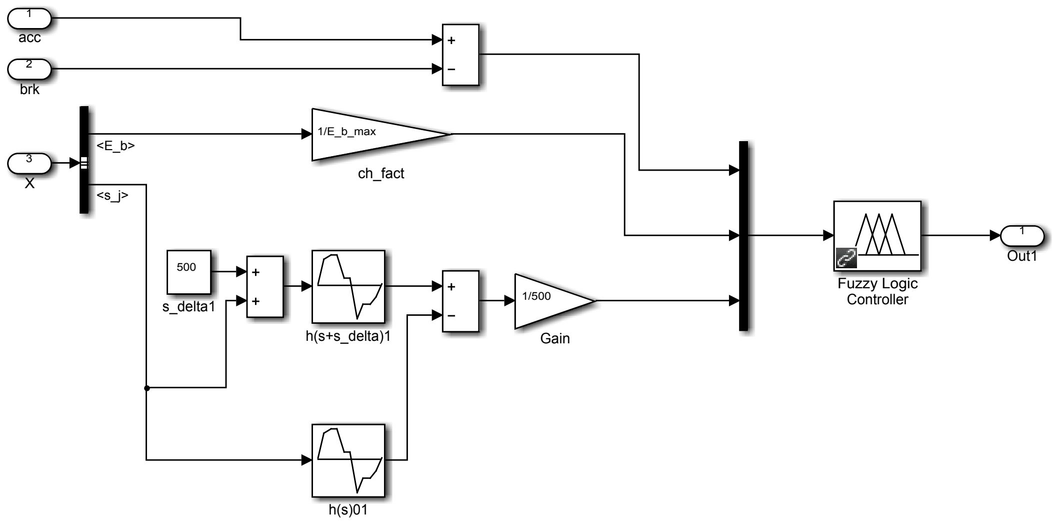

**Figure 2**Structure of the fuzzy subsystem

In order to keep low the complexity of the rule base we defined the input partitions with a relative low number of fuzzy sets. Thus the first input variable (*acc_brk*) has a partition with four sets (*NL*, *NS*, *PS*, *PL*) owing to the fact that it aggregates two signals (acceleration and braking). All membership functions are triangle shaped (see Fig. 3).

In case of the second input ($E_{brel}$) bell shaped membership functions were chosen. The corresponding partition is shown in Fig. 4. It contains three fuzzy sets (*S*, *M*, *L*).

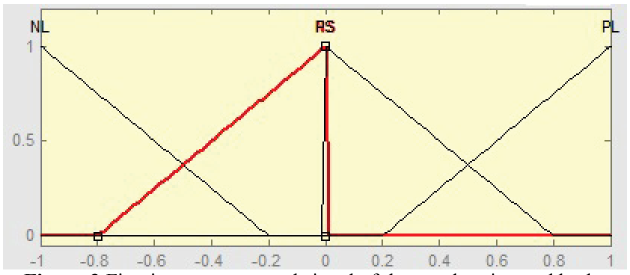

**Figure 3** First input - composed signal of the acceleration and brake signals

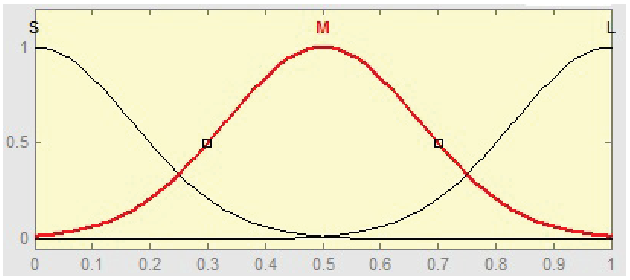

**Figure 4** Second input - relative value of SOC

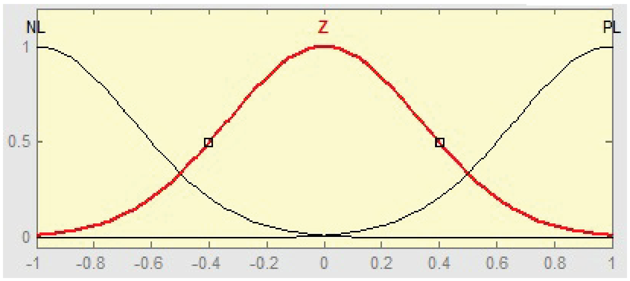

**Figure 5** Third input of the fuzzy controller - relative value of the height level change in the following 500 m

In case of the third input ($s_j$) also bell shaped membership functions were used. The corresponding partition is shown in Fig. 5.

In case of the output we need only three distinct values (0, 1, and 2) referring to the three possible charge states. The corresponding partition is created using singleton type membership functions. Its graphical representation is shown in Fig. 6.

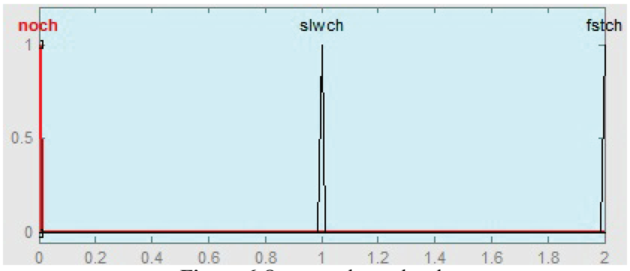

**Figure 6** Output - charge levels

Mamdani-type inference technique was chosen owing to the following two aspects: (1) it applies easily interpretable IF-THEN rules that contain both the antecedent and consequent sides fuzzy sets, and (2) it is implemented in Matlab's Fuzzy Logic Toolbox.

Mamdani type fuzzy reasoning activates and uses all the rules whose antecedent sets overlap the observation with a membership value greater than zero and generates the conclusion fuzzy set applying an *s*-norm to the results obtained firing the individual rules.

In our case there are only three acceptable output values, which situation demands the use of a special defuzzification method that yields the abscissa value where the conclusion set takes its maximum. The application of singleton consequent fuzzy sets ensure that this maximum can be only one of the desired output values.

Having the antecedent and consequent partitions defined an initial rule base was created with 36 rules based on assumptions of human experts.

## 4 Experimental results

### 4.1 Fitness function

In the course of the optimization process one is looking for that solution candidate, which proves to be better than all the others by means of an objective (fitness) function. In most of the cases the type of this function is "the smaller the better". It means that the lower the values of the objective function are the better the given parameter set will be considered. In our case the goal of the optimization was to reduce the harmful pollutant emission by taking into consideration the battery energy level and fuel consumption as well. The available SHEV model [23] made possible the measurement of five characteristics that were the fuel consumption of the ICE ($B$), the CO emission of the ICE ($e_{CO}$), the HC emission of the ICE ($e_{HC}$), the NOx emission of the ICE ($e_{NOx}$), and the battery energy level ($E_{br}$). Therefore, the cost function ($F$) used in [23] and [31] was applied as objective function.

$$D=\left(\frac{B}{B^{\max}}\right)^2+\left(\frac{e_{CO}}{e_{CO}^{\max}}\right)^2+\left(\frac{e_{HC}}{e_{HC}^{\max}}\right)^2+\left(\frac{e_{NOx}}{e_{NOx}^{\max}}\right)^2+\left(\frac{E_{br}}{E_{br}^{\max}}\right)^2,\quad(11)$$

$$F=\frac{\int_0^T\sqrt{D}dt}{5T},\quad(12)$$

where $T$ is the simulation time.

All components of the objective function were taken into consideration with equal weights. Thus lower cost function values represent lower fuel consumption and harmful pollutant emission and therefore correspond to better solutions. All components are relative values compared to predefined constant maximum values, which were taken from [31] (see Tab. 1).

**Table1** Constants used for normalization [30]

| Constant | Value | Unit |
|---|---|---|
| $B^{\max}$ | 0.0014 | kg/s |
| $e_{CO}^{\max}$ | $4.5747\times10^{-4}$ | kg/s |
| $e_{HC}^{\max}$ | $6.8620\times10^{-4}$ | kg/s |
| $e_{NOx}^{\max}$ | $5.7907\times10^{-4}$ | kg/s |
| $E_b^{\max}$ | $8.5\times10^{6}$ | J |

## 4.2 Parameter optimization

In case of this research the parameter optimization aimed at the improvement of the rule base by means of changing (if necessary) the categories (output values) associated to the individual rules. In the course of the optimization process the SHEV had to track the HTDC (Heavy Traffic Driving Cycle) [21] speed profile.

In case of all parameters of the fuzzy inference except the defuzzification the default values were used, i.e. the min operator as $t$-norm (*and* method), and the max operator as $s$-norm (*or* method). These are the most commonly used parameter values in case of Mamdani-type systems.

The algorithm has been implemented in Matlab. The position vector of a particle contained the consequents of the rules using a numerical encoding, i.e. 1 for no charge, 2 for slow charge, and 3 for fast charge. Due to their meaning the elements of the position vector can be only integer values. Thus the rounding operation (6) was also included. Its initial version was

$$P_0=[\,3\,3\,3\;\;3\,3\,3\;\;3\,3\,3\;\;2\,2\,2\;\;2\,2\,2\;\;2\,2\,2\;\;2\,2\,2\;\;2\,2\,2\;\;2\,2\,2\;\;1\,1\,1\;\;1\,1\,1\;\;1\,1\,1\,].\quad(13)$$

The simulations were run in Simulink with $T_s$=1500 s simulation time. The parameters of the algorithm were $n$=36, $N$=25, $c_1$=0.2, $c_2$=0.2, $t_{nr}$=2, $t_s$=1, $v_s$=150, $n_B$=1, $n_l$=1, $s_k$=$1_{1\times36}$, $LB$=$1_{1\times36}$, $UB$=$3_{1\times36}$. The resulting initial value of the fitness function was $F$=0.0342. At the end of the optimization the fitness value became $F$=0.0141.

The same optimization task was also carried out using the original PSO algorithm with similar parameters and a clonal selection based artificial immune system algorithm (CLONALG) [32] with the same population size and allowed system evaluation number. The final fitness values are presented in Tab. 2. Thus the proposed algorithm proved to provide the best results.

**Table 2** Final fitness values

| Method | Fitness |
|---|---|
| PSO | 0.0229 |
| Modified PSO | 0.0141 |
| CLONALG | 0.0233 |

The rule base corresponding to the best particle of the final swarm is presented in Tab. 3.

**Table 3** Final rule base

| | | $E_{brel}$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | S | | | M | | | L | | |
| | | Height level change | | | | | | | | |
| | | NL | Z | PL | NL | Z | PL | NL | Z | PL |
| acc_brk | NL | 3 | 3 | 2 | 1 | 2 | 3 | 3 | 2 | 1 |
| | NS | 2 | 1 | 1 | 3 | 3 | 2 | 3 | 1 | 2 |
| | PS | 2 | 3 | 2 | 2 | 2 | 1 | 1 | 2 | 3 |
| | PL | 3 | 3 | 1 | 3 | 3 | 2 | 1 | 2 | 3 |

## 5 Conclusions

In this paper, a modified version of the particle swarm optimization was proposed, which was applied for the optimization of a fuzzy classification subsystem used in the model of a series hybrid electric vehicle. Here the goal was to ensure a low level of harmful pollutant emission and low fuel consumption. The fuzzy classifier controls the internal combustion engine by switching between three possible charging states. It applies Mamdani type fuzzy reasoning and a special defuzzification technique. The proposed optimization algorithm was used for the determination of the consequent sets of the rules.

The new technique combines the original PSO with a local search technique at the end of each iteration cycle. The local optimum of the objective function is approached so that in the course of each step two new candidate positions are calculated in the neighbourhood of the current particle and the particle will be moved in that position, which provides the best objective function value.

The results obtained by the proposed method were compared to the ones achieved with the original PSO and the artificial immune system algorithm CLONALG. The new method outperformed both of them. Further research will consider the application of a fuzzy rule interpolation based interpolation technique (e.g. [33÷36]) in order to reduce the complexity of the rule base and the time necessary for fitness evaluation.

## Acknowledgements

This research is supported by EFOP-3.6.1-16-2016-00006 "The development and enhancement of the research potential at Pallas Athena University" project. The Project is supported by the Hungarian Government and co-financed by the European Social Fund.

## 6 References

[1] Valle, Y. D.; Venayagamoorthy, G. K.; Mohagheghi, S.; Hernandez, J. C.; Harley, R. G. Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power System. // IEEE Transactions on Evolutionary Computation, 12, 2(2008), pp. 171-195. https://doi.org/10.1109/TEVC.2007.896686

[2] Kennedy, J.; Eberhart, R. Particle Swarm Optimization. // Proceedings of IEEE International Conference on Neural Networks IV. / Perth, 1995, pp. 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

[3] Tang, M.; Gong, D.; Liu, S.; Zhang, H. Applying multi-phase particle swarm optimization to solve bulk cargo port scheduling problem. // Advances in Production Engineering & Management. 11, 4(2016), pp. 299-310. http://dx.doi.org/10.14743/apem2016.4.228

[4] Richards, M.; Ventura, D. Choosing a starting configuration for particle swarm optimization. // Proceedings of the IEEE International Joint Conference on Neural Networks / Budapest, 2004, vol. 3, pp. 2309-2312.

[5] Kennedy, J.; Mendes, R. Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms. // Proceedings of the IEEE International Workshop Soft Computing in Industrial Applications / Binghamton, 2003, pp. 45-50. https://doi.org/10.1109/SMCIA.2003.1231342

[6] Fan, H. Y.; Shi, Y. Study on Vmax of particle swarm optimization. // Proceedings of the Workshop on Particle Swarm Optimization / Purdue School of Engineering and Technology, Indianapolis, 2001.

[7] Clerc, M.; Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. // IEEE Transactions on Evolutionary Computation, 6, 1(2002), pp. 58-73. https://doi.org/10.1109/4235.985692

[8] Shi, Y.; Eberhart, R. Parameter selection in particle swarm optimization. // Proceedings of the 7th Annual Conference on Evolutionary Programming / San Diego, 1998, pp. 591-601. https://doi.org/10.1007/BFb0040810

[9] Chen, Q.; Guo, G.; Li, C. An Improved PSO Algorithm to Optimize BP Neural Network. // Proceedings of the 5th International Conference on Natural Computation / Tianjian, 2009, pp. 357-360. https://doi.org/10.1109/ICNC.2009.436

[10] Pehlivanoglu, Y. V. A New Particle Swarm Optimization Method Enhanced with a Periodic Mutation Strategy and Neural Networks. // IEEE Transactions on Evolutionary Computation, 17, 3(2012), pp. 436-452. https://doi.org/10.1109/TEVC.2012.2196047

[11] Campana, E. F.; Fasano, G.; Pinto, A. Dynamic system analysis and initial particles position in particle swarm optimization. // Proceedings of the IEEE Swarm Intelligence Symposium /Indianapolis, 2006, pp. 202-209.

[12] Precup, R. E.; David, R. C.; Stinean, A. I.; Radac, M. B.; Petriu, E. M. Adaptive Hybrid Particle Swarm Optimization-Gravitational Search Algorithm for Fuzzy Controller Tuning. / Proceedings of the 2014 IEEE International Symposium on Innovations in Intelligent Systems and Applications INISTA 2014 /Alberobello, 2014, pp. 14-20. https://doi.org/10.1109/INISTA.2014.6873591

[13] Aarts, E. H. L.; Lenstra, J. K. Local search in combinatorial optimization. Princeton University Press, 2003.

[14] Poursamad, A.; Montazeri, M. Design of genetic-fuzzy control strategy for parallel hybrid electric vehicles. // Control Engineering Practice, 16, (2008), pp. 861-873. https://doi.org/10.1016/j.conengprac.2007.10.003

[15] Barsali, S.; Ceraolo, M.; Possenti, A. Techniques to control the electricity generation in a series hybrid electrical vehicle. // IEEE Transactions on Energy Conversion, 17, 2 (2002), pp. 260-266. https://doi.org/10.1109/TEC.2002.1009478

[16] Brahma, A.; Guezennec, Y.; Rizzoni, G. Optimal energy management in series hybrid vehicles. // Proceedings of the American Control Conference / Chicago, 2000, pp. 60-64. https://doi.org/10.1109/ACC.2000.878772

[17] Kheir, N. A.; Salman, M. A.; Schouten, N. J. Emissions and fuel economy trade-off for hybrid vehicles using fuzzy logic. // Mathematics and Computers in Simulation, 66, (2004), pp. 155-172. https://doi.org/10.1016/j.matcom.2003.11.007

[18] Weimin, L.; Guoqing, X.; Yangsheng, X. Online Learning Control for Hybrid Electric Vehicle. // Chinese Journal of Mechanical Engineering, 25, 1(2012), pp. 98-106. https://doi.org/10.3901/CJME.2012.01.098

[19] Miller, J. M.; Ehsani, M.; Gao, Y. Understanding power flows in HEV eCVT's with ultracapacitor boosting using simplorer. // Proceedings of the IEEE Power Propulsion Conference/ Chicago, 2005, pp. 742-746.

[20] Chau, K. T.; Wong, Y. S. Overview of power management in hybrid electric vehicles. // Energy Conversion and Management, 43, (2002), pp. 1953-1968. https://doi.org/10.1016/S0196-8904(01)00148-0

[21] Bári, G.; Varga, D.; Kocsis, B.; Ailer, P. Subjective features of hybrid electrical drive train conception (in Hungarian). // A jövőjárműve, 03-04, (2013), pp. 46-51.

[22] Bári, G.; Varga, D.; Kocsis, B.; Trencséni, B.; Ailer, P. Comparison of hybrid electrical drive train conceptions based on objective features (in Hungarian). // A jövőjárműve, 03-04, (2013), pp. 52-58.

[23] Research for hybrid and electronic vehicle design – Vehicle dynamics and control. Vehicle concept and energy management of hybrid drive train, Technical Report TÁMOP-4.2.2.A-11/1/KONV-2012-0012, 2014.

[24] Devasenapati, S. B.; Ramachandran, K. I. Hybrid Fuzzy Model Based Expert System for Misfire Detection in Automobile Engines. // International Journal of Artificial Intelligence, 7, A11(2011), pp. 47-62.

[25] Portik, T.; Pokorádi, L. Fuzzy rule based risk assessment with summarized defuzzyfication. // Proceedings of the XIIIth Conference on Mathematics and its Applications / Timisoara, 2013, pp. 277-282.

[26] Precup, R. E.; Preitl, S.; Rădac, M. B.; Petriu, E. M.; Dragoş, C. A.; Tar, J. K. Experiment-based teaching in

advanced control engineering. // IEEE Transactions on Education, 54, 3(2011), pp. 345-355. https://doi.org/10.1109/TE.2010.2058575
[27] Schipor, O. A.; Pentiuc, S. G.; Schipor, M. D. Improving Computer Based Speech Therapy Using A Fuzzy Expert System. // Computing and Informatics, 29, (2010), pp. 303-318.
[28] Škrjanc, I.; Blažič, S.; Matko, D. Direct fuzzy model-reference adaptive control. // International Journal of Intelligent Systems, 17, 10(2002), pp. 943-963. https://doi.org/10.1002/int.10054
[29] Vaščák, J. Approaches in adaptation of fuzzy cognitive maps for navigation purposes. // Proceedings of the 8th International Symposium on Applied Machine Intelligence and Informatics – SAMI 2010 / Herľany, 2010 pp. 31-36.
[30] Yorita, A.; Botzheim, J.; Kubota, N. Self-efficacy using fuzzy control for long-term communication in robot-assisted language learning. // Proceedings of the International Conference on Intelligent Robots and Systems (IROS) / Tokyo, 2013, pp. 5708-5715. https://doi.org/10.1109/IROS.2013.6697183
[31] Johanyák, Z. C. A Simple Fuzzy Logic Based Power Control for a Series Hybrid Electric Vehicle. // Proceedings of the 9th IEEE European Modelling Symposium on Mathematical Modelling and Computer Simulation (EMS 2015), October 6-8 Madrid, 2015, Spain Ed.: David Al-Dabass, Gregorio Romero, Alessandra Orsoni, and Athanasios Pantelous, IEEE Computer Society-Conference Publishing Services, ISBN: 978-1-5090-0206-1, pp. 207-212. https://doi.org/10.1109/EMS.2015.40
[32] Johanyák, Z. C. Clonal Selection Based Parameter Optimization for Sparse Fuzzy Systems. // Proceedings of IEEE 16th International Conference on Intelligent Engineering Systems (INES 2012), June 13–15, 2012, Lisbon, Portugal, pp. 369-373 https://doi.org/10.1109/INES.2012.6249861
[33] Johanyák, Z. C. Performance Improvement of the Fuzzy Rule Interpolation Method LESFRI. // Proceedings of the 12th IEEE International Symposium on Computational Intelligence and Informatics / Budapest, 2011, pp. 271-276. https://doi.org/10.1109/CINTI.2011.6108512
[34] Kovács, L.; Ratsaby, J. Analysis of Linear Interpolation of Fuzzy Sets with Entropy-based Distances. // Acta Polytechnica Hungarica, 10, 3(2013), pp. 51-64.
[35] Perfilieva, I.; Wrublova, M.; Hodakova,P. Fuzzy Interpolation According to Fuzzy and Classical Conditions. // Acta Polytechnica Hungarica, 7, 4(2010), pp. 39-55.
[36] Vincze, D.; Kovács, S. Performance Optimization of the Fuzzy Rule Interpolation Method "FIVE". // Journal of Advanced Computational Intelligence and Intelligent Informatics, 15, 3(2011), pp. 313-320. https://doi.org/10.20965/jaciii.2011.p0313

**Authors' addresses**

***Zsolt Csaba Johanyák, PhD, Professor***
Department of Information Technology,
Pallasz Athéné University
Izsáki út 10., H-6000, Kecskemét, Hungary
johanyak.csaba@gamf.kefo.hu