International Journal of Mathematics and
Computer Science, **16**(2021), no. 1, 357–377

A novel multi-population passing vehicle search algorithm based co-evolutionary cultural algorithm

Maryam Alami Chentoufi, Rachid Ellaia

Laboratory of study and research in applied mathematics (LERMA)
Engineering for Smart and Sustainable Systems Research Center (E3S)
Mohammed V University of Rabat
Agdal, Rabat, Morocco

email: maryam.lami12@gmail.com

(Received November 29, 2019, Accepted July 20, 2020)

Abstract

In order to fully use the advantages of the PVS algorithm and CA, a hybrid co-evolutionary cultural algorithm based on PVS is proposed. In this algorithm, a co-evolutionary mechanism between two cultural algorithms sub-populations is established to take full advantage of CA. The objective, like other meta-heuristics, is to find global optimum or quasi-optimal solutions for a given function. To verify the performance of the proposed algorithm, it will be tested on a set of benchmark functions. The results demonstrated the superior effectiveness of CA-PVS compared with other meta-heuristic optimization algorithms.

1 Introduction

Evolutionary Computation (EA) started in the 1960s and it has been used in too many hard and complex optimization as engineering problems, combinatorial optimization, and function optimization with and without constraints. Nowadays, several meta-heuristic optimization algorithms have been developed and applied to various real-life problems such as population-based meth-

Key words and phrases: Global optimization, hybrid co-evolutionary, Passing vehicle search, Hybridization, Cultural algorithm, Engineering design optimization.
AMS (MOS) Subject Classifications: 65K10.
ISSN 1814-0432, 2021, http://ijmcs.future-in-tech.net

ods, heuristics, meta-heuristics, advanced optimization techniques, computational intelligence algorithms, non-traditional optimization techniques and intelligent algorithms. EA was inspired by biological evolution. Consequently, it explores the solution space by selection of the fittest candidate solutions [1]. Hybridization of techniques from various domains is an active area of research in computer science. CA, co-evolutionary algorithm (CEA) and PVS are all promising methods in the field of intelligent computation. In this paper, co-evolutionary mechanism between two cultural algorithms is built and PVS algorithm is used in population space of each CA.
PVS considers the mathematics of the passage of a vehicle on a two-lane highway [2]. Similar to other meta-heuristic methods, PVS is a population-based method that requires starting an initial set of solutions and then seeking the optimal solution by following the mathematical characteristics of vehicles that pass on a two-lane highway. CA was first proposed by [3] in 1994 in order to model the evolution of cultural systems based upon principles of human social evolution taken from the social science literature. Indeed, in human society, culture can be described as a complex whole which includes knowledge, beliefs, arts, morals, customs and any other capabilities and habits acquired by man as a member of society [4]. Cultures are sometimes interchanged through the interaction among generations and bring about new cultures [5]. The proposed algorithm implements the concept of cultural exchanges between sub-populations which provides opportunities to create cultural diversity and then explore new solutions in the search space. During our work, two types of knowledge have been added to the belief space of cultural algorithm. The situational knowledge to prevent the algorithm into local optimization by saving the information of best individuals and ensuring the diversity of individuals and the normative knowledge to increase the speed of convergence by guiding the evolution in Passing vehicle search with different selection.
The rest of this paper is organized as follows: in section 2 an overview of PVS, CEA and CA is presented. The proposed algorithm is introduced in section 3. Experiments and results are discussed in section 4. We conclude our paper, in section 5.

2 Background

A Cultural algorithm (CA) with single population ($CA-S$) and multi population ($CA-IM$) was implemented with island model ($CA-IM_1$ and

$CA - IM_2$) in [6], the proposed framework allowed the migration among islands sub-populations and the main population through belief space structures. Results showed that $CA - IM_1$ and $CA - IM_2$ perform the optimum search and reach optimum values equally or above the ones reached by algorithms DGA and DGA-SRM.
In addition to that, a hybrid co-evolutionary cultural algorithm based on particle swarm optimization (CECBPSO) was proposed in [7], CECBPSO is built through introducing a novel space called shared global belief space (SGBS) into the co-evolutionary mechanism where operations of sharing knowledge and experience can help the algorithm improving performance. This algorithm showed a superior performance with respect to other algorithms in terms of accuracy and convergence speed, especially on high-dimensional problems. Many approaches have been proposed, in the literature using particle swarm optimization based on cultural algorithm for different problems such as Short-term Optimal Operation of Cascade Hydropower Stations [8] and clearance of Flight control law [9].
To our best knowledge, PVS has not been used with cultural algorithm in the literature which give originality to this research. Passing vehicle search is a new algorithm developed by [2].

2.1 Passing Vehicle Search algorithm

2.1.1 Introduction

PVS is a meta-heuristic optimization algorithm based on the passing behavior of a vehicle moving on a two-lane highway. This algorithm is based on a human activity. Like other meta-heuristics, the objective of the algorithm is to find the global solution or near-optimal solutions for a given function.
In the PVS modeling, we consider three vehicles that are involved in the passing mechanism on a two-lane highway; back vehicle (BV), front vehicle (FV), and oncoming vehicle (OV).

2.1.2 Mathematical modeling of passing vehicles on a two-lane highway

We assume that there are three different vehicles (BV, FV and OV) on a two-lane highway, which have different velocities (V_1, V_2 and V_3) at any particular time instance. Passing only occurs if the passing supply is more than the passing demand. BV wants to pass FV, which is only possible if FV is slower

than BV. If FV is faster than BV, then passing is not possible by BV. Passing depends on the position and the speed of OV as well as the distance between BV, FV, and OV, and their velocities, which generate different conditions, (figure 2).
We consider:

- x : Distance between BV and FV.
- y : Distance between FV and OV.
- X_1, X_2, X_3 : Distance from reference line of BV, OV and FV respectively.
- V_1, V_2, V_3 : Velocities of BV, OV and FV respectively.

The conditions we are going to deal with are:

1. FV is slower than BV ($V_3 < V_1$) (Primary condition-1)
   (a) $(y - y_1) > x_1$ (Secondary condition-1)
   (b) $(y - y_1) < x_1$ (Secondary condition-2)
2. FV is faster than BV ($V_3 > V_1$) (Primary condition-2)

2.1.3 Primary Condition 1

In this condition, we suppose that FV is slower than BV which means that the velocity of BV is more than the velocity of FV ($V_F < V_B$). In this case, there is a possibility for BV to overtake FV. We can divide this condition into two sub-conditions:

1. Secondary condition 1.
2. Secondary condition 2.

Secondary condition 1 Let's consider x_1 the distance traveled by FV when BV catches FV and overcomes it. Let, t be the time needed for BV to catch FV and thus passes it. The distance traveled by FV is equal to:

$$x_1 = V_3 t \tag{2.1}$$

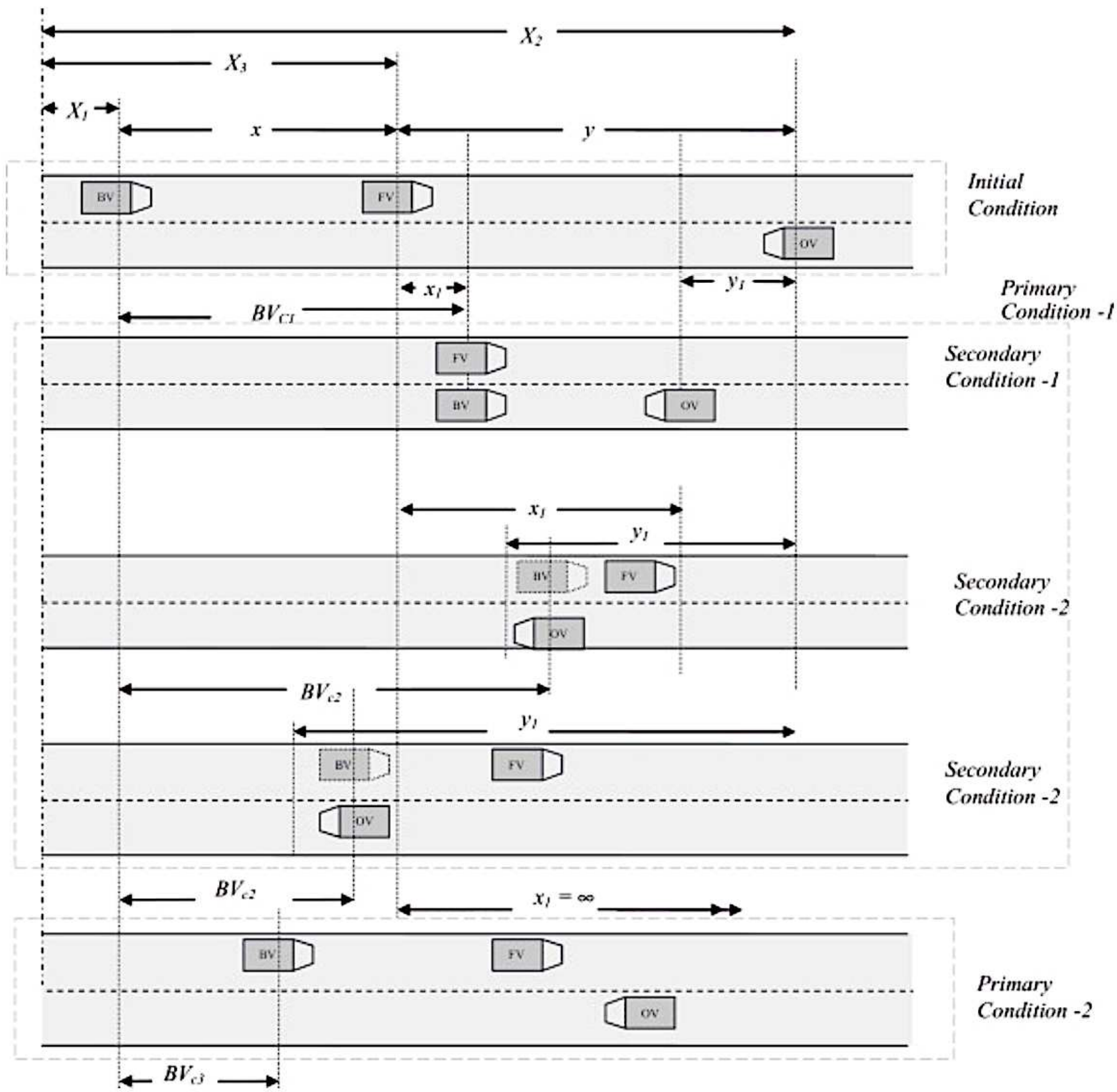


Figure 1: Overtaking rules [2]

And the distance traveled by BV is then:

$$x + x_1 = V_1 t \tag{2.2}$$

Equating (2.1) and (2.2), we get:

$$\frac{x_1}{V_3} = \frac{x + x_1}{V_1} \tag{2.3}$$

Therefore, we obtain,

$$x_1 = \frac{V_3 x}{V_1 - V_3} \tag{2.4}$$

The distance traveled by OV is:

$$y_1 = V_2 t \tag{2.5}$$

By substituting x_1 in (2.1) we get:

$$t = \frac{x}{V_1 - V_3} \tag{2.6}$$

Substituting (2.6) into (2.5), we obtain:

$$y_1 = \frac{V_2 x}{V_1 - V_3} \tag{2.7}$$

Now, the change in position of BV is given by:

$$BV_{c1} = x + x_1 \tag{2.8}$$

Substituting the value of x_1 into (2.4), we get:

$$BV_{c1} = x(\frac{V_1}{V_1 - V_3}) \tag{2.9}$$

Substituting the value of x that corresponds to the reference line, we obtain:

$$BV_{c1} = (X_3 - X_1)\frac{V_1}{V_1 - V_3}. \tag{2.10}$$

Now, the change in the position of BV from the reference line is given by:

$$\boxed{X_1 + BV_{c1} = X_1 + (x_3 - x_1)\frac{V_1}{V_1 - V_3}} \tag{2.11}$$

Secondary condition 2 This is illustrated in Figure 1 for two different values of $y - y_1$, with one positive and the other negative. In either of these situations, BV cannot exceed FV before OV crosses BV. An accident can be avoided if BV does not change lanes until OV crosses BV. The distance between BV and OV is somewhere between the initial positions of BV and OV. Therefore, the change of position of BV is given by:

$$BV_{c2} = R(x + y), \tag{2.12}$$

where R is a random number from (0,1) range.
The change in the position of BV from the reference line is equal to:

$$X_1 + BV_{c2} = X_1 + R(x + y) = X_1 + R(X_2 - X_1) \tag{2.13}$$

2.1.4 Primary condition 2

If FV is faster than BV, then it is not possible for BV to pass FV. Thus, the change in the position of BV is as follows:

$$X_1 + BV_{c3} = X_1 + Rx = X_1 + R(X_3 - X_1). \tag{2.14}$$

2.2 co-Evolutionary algorithm

In spite of the improvement of the evolutionary algorithms by many researchers that proposed encouraging methods, EA doesn't take into consideration the influence of other existing populations or outer environment. Therefore, a large-scale complex real-world problem presents experimental results that are still not up to the standard. The concept **co-evolution** was introduced to resolve this problem by describing the mutual influences in evolving processes of different species. Darwin was the first to propose the original idea of co-evolution [10]. Consequently, co-evolutionary algorithms were EA on the base of co-evolutionary theory, and there are two kinds of CEA: Cooperative co-evolutionary Algorithm (CoopCEA) and Competitive co-evolutionary Algorithm (CompCEA).
In CoopCEA, individuals help each other by cooperative operations and they are rewarded when they work well together. On the other hand, competitive co-evolution means there are several interactional populations. They compete with each other in order to obtain more resources and live space.

2.3 Cultural Algorithm

Cultural Algorithm is a kind of optimal algorithm that can be applied in nearly all fields that optimization algorithms can manage. It was first proposed by [11] and it has been successfully applied in a number of diverse application areas: it includes the experiences of the evolutionary population that are integrated into a belief space made up of various forms of symbolic knowledge. The knowledge sources include normative knowledge, spatial knowledge (topographic), temporal (historic) knowledge, domain knowledge, and exemplar knowledge.
Cultural algorithms operate on two spaces. First, they operate on the population space as any other evolutionary computation technique in which a set of individuals (called population) is adopted. Each individual has a set of features independent from each other which allows us to determine its fitness. Through time, such individuals can be replaced by some of its descendants, obtained after applying a set of operators to the population.
The second space is the belief space where the knowledge acquired by the individuals along the evolutionary process is stored. The information contained in this space must be accessible to any individual so that he/she can use it to modify his/her behavior. To unify both spaces, a communication protocol is established to dictate the rules about individuals that can contribute to knowledge in the belief Space (function of acceptance) and how the belief Space will influence new individuals (Function of Influence). The two most used ways to represent knowledge in the belief space are: Situational Knowledge and Normative Knowledge [12].
Situational Knowledge represents the best individuals found at certain time of evolution. Normative Knowledge represents a set of intervals that characterize the range of values given by the features that make the best solutions. These intervals are used to guide the adjustments (mutations) that occur in individuals.
The adjustment of the range of Normative Knowledge varies according to the best individual; that is, if the individual was accepted by the acceptance function and its range is less than the range stored in the belief space, the range is adjusted and vice versa.
The resolution of problems produces experiences from an individual in the population space which are selected to contribute to the acceptance by the belief space, where the knowledge is generalized and stored. In the initial population, the individuals are evaluated by the fitness function. Then the information on the performance of the function is used as a basis for the

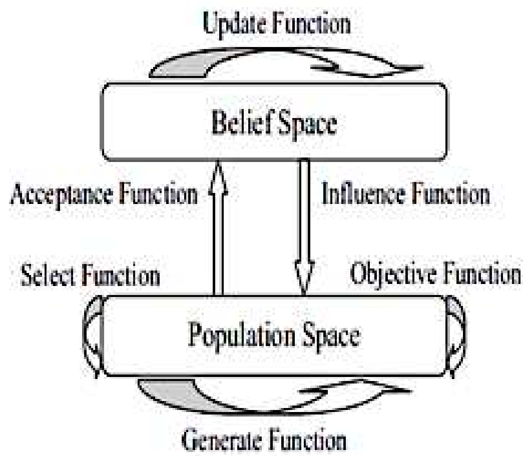


Figure 2: CA framework

production of generalizations for next generations.
The experiences of the individuals selected will be used to make the necessary adjustments on the knowledge of the current belief space [13]. The process of CA can be briefly described as follows:

- Step 1: Build population space and belief space, and initialize all the individuals in the two spaces randomly, and then evaluate the fitness of each individual.
- Step 2: Stop the algorithm if the stopping criterion is satisfied, otherwise go to Step 3.
- Step 3: Update the individuals in the two spaces according to their own rules and evaluate the fitness value of each individual.
- Step 4: Update belief space by accept operation.
- Step 5: Update population space by affect operation.
- Step 6: Return to step 2.

2.3.1 Design of the belief space

The acceptance function selects individuals that can directly impact the information of current belief space. The number of the accepted individuals can be represented as follows:

$$\eta_{accept} = (p\% + p\%/t) * N, \tag{2.15}$$

where p is the probability, given according to the needs and is normally set to 20 %; t is the current iteration; N is the size of population.
The normative knowledge contains the intervals for each domain variable in order to move the new individuals towards to the good solutions. For each parameter j, $NK =< Ij, Lj, Uj >$ where I_j denotes the internal layout of the parameter j, which is defined as follows:

$$I_j = [l_j, u_j] = x|l_j \leq x \leq u_j, x \in R, \tag{2.16}$$

where l_j and u_j are the lower and upper bounds of the belief space $B1$ and $B2$. U_j and L_j are the values of the fitness function corresponding to the bound l_j and u_j.
Then the normative knowledge is renewed under the assumption that the lower bound of j_th individual is affected by i_th individual and upper bound is affected by k_th individual. The normative knowledge update rules are as follows:

$$l_j^{t+1} = \begin{cases} x_{ij}^t x_{ij}^t \leq l_j^t \ \ or \ \ f(x_{ij}^t) < L_j^t \\ \qquad\qquad l_j^t \ \ otherwise \end{cases} \tag{2.17}$$

$$u_j^{t+1} = \begin{cases} x_{kj}^t x_{kj}^t \geq u_j^t \ \ or \ \ f(x_{kj}^t) < U_j^t \\ \qquad\qquad u_j^t \ \ otherwise \end{cases} \tag{2.18}$$

$$\begin{cases} L_j^{t+1} = f(x_{ij}^t) x_{ij}^t \leq l_j^t \ \ or \ \ f(x_{ij}^t) < L_j^t \\ \qquad\qquad L_j^t \ \ otherwise \end{cases} \tag{2.19}$$

$$\begin{cases} U_j^{t+1} = f(x_{kj}^t) x_{kj}^t \geq u_j^t \ \ or \ \ f(x_{kj}^t) < U_j^t \\ \qquad\qquad U_j^t \ \ otherwise \end{cases} \tag{2.20}$$

Situational knowledge consists of the best exemplar found along the evolutionary process. It represents a leader for the other individuals to follow. Situational knowledge to be adjusted according to the update function Update():

$$\begin{cases} P_g^{t+1} = x_{best}^t f(x_{best}^t) < f(P_g^t) \\ \qquad\qquad P_g^t \ \ otherwise, \end{cases} \tag{2.21}$$

where x_{best}^t represents the best individual of the t^{th} generation. If the optimal solution x_{best} in current population is better than p_g, and available x_{best}

instead of p_g to complete update the situational knowledge. After the belief space to complete its own update by affecting the function of the population space solutions exert influence. This requires that by the following equation to adjust the position of the solution in the population space.

$$\begin{cases} x_{id}^{t+1} = \mathrm{x}_{id}^{t} + |size(I_d) * N(0.1)|(x_{id}^{t} < P_{gd}) \\ \mathrm{x}_{id}^{t} - |size(I_d) * N(0.1)|(x_{id}^{t} > P_{gd}) \\ \mathrm{x}_{id}^{t} + \eta * size(I_d) * N(0.1)|(x_{id}^{t} = P_{gd}), \end{cases} \tag{2.22}$$

where $N(0,1)$ is the standard normal distribution random number, $size(I_d)$ is the belief space variable d adjustable interval length, η is the step length contraction factor.

3 The proposed method

In order to fully use the advantages of PVS and CA, a hybrid co-evolutionary CA based on PVS is proposed. In this algorithm, a co-evolutionary mechanism between two cultural algorithms is established to take full advantage of CA and CEA. PVS is used to evolve the individuals in the two population spaces, the two belief spaces are directed by the <Situational knowledge, Normative knowledge> structure, and the cultural exchange is ensured between the two belief spaces due to cooperative co evolutionary mechanism.
The next figure shows the framework of the new algorithm. In CECBPVS, there are two cultural-algorithm-populations. PVS is used in all the sub spaces (Belief space 1 and 2, and Population space 1 and 2). In each generation, individuals of the two sub-belief spaces (Belief space 1 and 2) are exchanged. Excellent individuals in each belief space will be reserved and the bad ones will be abandoned and replaced by reinitialized individuals. Then the two belief spaces will implement affect operations to the two subpopulation spaces (Population spaces 1 and 2). The two sub-population spaces also exchange their experience in each generation. Through the synergistic mechanism, the algorithm has higher probability of avoiding local optima.

3.1 Pseudocode

- Step 1: Define the parameters such as population size (PS), stopping criteria (Number Of Generations (NOG), Maximum function evaluations (FE), and error) number of design variables (DV), bounds on

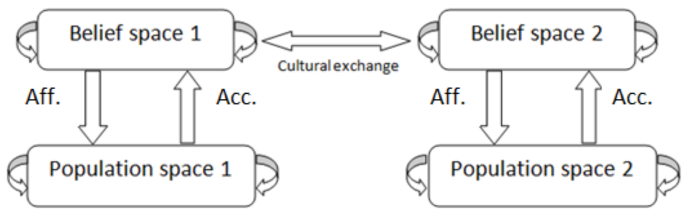


Figure 3: The novel PVS CA framework

design variables ((LB),(UB)).
Define the function for optimization:
Minimize $f(x)$
subject to $X = X_1, X_2, ..., X_{DV}$, where $LB_i \leq X_i \leq UB_i$

- Step 2: Generate an initial set of solutions.

- Step 3: Set generation counter t = 1

- Step 4: Establish four spaces P1, B1, P2 and B2 and initialize all the individuals in the four spaces

- Step 5: Select any three set of parameters from the given solution space and out of which one solution represents the Current/Back Vehicle (BV) and the other two solutions are selected randomly and represent the Front Vehicle (FV) and the oncoming Vehicle (OV).

- Step 6: Evaluate the Distance (D) and Velocities (V) for the three set of parameters (vehicles).
  The distances between vehicles are calculated based on the fitness value. Arrange the population in ascending order and calculate the distance using eq. 24. From the expressions, it can be observed that the value of the distance is normalized to 1. The velocity corresponding to the vehicle is calculated using eq. 25.

$$D_k = \frac{n_k}{PS} \tag{3.23}$$

$$V_k = R_k(1 - D_k), \tag{3.24}$$

where $k = (1, 2, 3)$ and R_k is a random number $\in (0, 1)$

- Step 7: Depending upon the velocity and distance of the set of parameters, update the solution space and if the solution is better than previous solution, accept the new solution (new parameters).
- Step 8: Evaluate the elite solution (Acceptance function from p1 to b1 and from p2 to b2) (eq. 16)
- Step 9: Update normative knowledge eqs. (17, 18, 19)
- Step 10: Update situational knowledge eq.(20)
- Step 11: Implement cultural exchange between two belief spaces (situational knowledge, Normative Knowledge) once per period.
- Step 12: The location of each solution is updated in the population space by the influence function (eq. 21)
- Step 13: Repeat the procedure until the termination criteria is satisfied, else go to Step 2

3.2 Cultural exchange between B1 and B2

In our algorithm we use cultural exchange between the two belief subspaces in the situational knowledge and the normative knowledge simultaneously, it is important to mention that the exchange in this study is based on the position of individuals in the normative knowledge, and the frequency of the exchanging cultures is implemented once per period.

4 Numerical results

In order to test the performance of the proposed hybrid passing vehicle search algorithm, we first introduce the set of parameters used by the proposed algorithm (Table 1).

Table.1:Algorithm Parameters

Parameter	Definition
nPop	population size
nVar	Number of Decision Variables f
varsize	Decision Variables Matrix Size
var_{min}	Decision Variables Lower Bound
var_{max}	Decision Variables Upper Bound
Maxit	Maximum Number of Iterations
pAccept	Acceptance Ratio
nAccept	Number of Accepted Individuals
M_f	Maximum number of Function Evaluation
α	
β	

Table 1: Algorithm Parameters

4.1 Test Functions

An efficient global optimization algorithm must avoid being trapped to local optima in order to converge to the global optimal. The most complex case appears when the local optima are randomly distributed in the search space. Another challenge is the dimensional of the search space which is an important factor in the complexity of the problem. In general, the complexity increases with the dimensions. Therefore, in order to investigate the hybrid CA-PVS performances, a wide set of 50 classical benchmark problems are selected with diverse properties and different levels of complexity to evaluate our proposed algorithm, as illustrated in Tables 1 and 2.

4.2 Simulation Parameters

In the current study, the following simulation parameters are adopted for Hybrid CA-PVS. Results are shown in table 4 (Dimension 3) and table 5 (Dimension 4). Hybrid CA with PVS specific parameters:

- Dimension : 3 and 4.
- Lower Bound and Upper Bound : depends on the benchmark function.
- Population Size= 20.
- Maximum number of Iterations/Generation : 200.
- pAccept=0.35; Acceptance Ratio.
- nAccept=round(pAccept*nPop) = 7 ; Number of Accepted Individuals
- $\alpha = 0.3$;

No	Fct. Name	Optimal	Mean	STD	Time (s)	SR
f_1	Sphere	0	6,39523E-70	2,9103E-69	0,49	100%
f_3	Ackley	0	1,0066E-15	6,48634E-16	0,55	100%
f_5	Rastrigin	0	0,994958721	1,306444188	0,54	43%
f_7	Quartic	0	0,000833544	0,000534235	0,52	67%
f_8	Zakharov	0	3,06867E-61	1,61502E-60	0,55	100%
f_9	Step	0	0	0	0,47	100%
f_{11}	Penalized 1	0	1,57054E-31	6,68088E-47	0,68	100%
f_{12}	Penalized 2	0	0,000366246	0,002006009	0,63	97%
f_{13}	Himmelblau	-78,3323	-73,9342	4,78213	0,48	53%
f_{14}	Qing	0	3,9443E-31	0	0,47	100%
f_{15}	Mishra 11	0	0	0	0,48	100%
f_{16}	Alpine	0	6,29126E-17	1,89924E-16	0,51	100%
f_{19}	Schwefel 2.21	0	6,18712E-34	1,89921E-33	0,49	100%
f_{20}	Schwefel 2.22	0	1,40554E-36	2,36531E-36	0,50	100%
f_{21}	Powell sum	0	4,53789E-84	1,37E-83	0,55	100%
f_{24}	Dixon Price	0	4,02648E-32	1,25088E-32	0,47	100%
f_{27}	Multimod	0	-1	0	0,52	100%
f_{28}	Exponential	-1	-1	0	0,47	100%
f_{29}	Sumsquares	0	1,89682E-68	9,98428E-68	0,48	100%
f_{30}	Brown	0	2,31237E-72	1,1603E-71	0,49	100%

Table 2: Simulation results Hybrid CA-PVS without cultural exchange for $f_1 - f_{30}$ on dimension $D = 3$ over 30 runs

- $\beta = 0.5$;

The experiments are performed on the set of Benchmark functions among (f_1-f_{30})

4.3 Hybrid CA-PVS Simulation Results

Table 4 shows us the optimization results of some of the multidimensional benchmark functions in terms of the mean best cost (Mean), the standard deviation (STD), the run time (Time), and success rate (SR). These simulation results show the high accuracy of the Hybrid CA-PVS algorithm for almost all the considered problems: a success rate of 100% is obtained for all the functions except for Penalized 2 (97%), Quartic (67%), Himmelblau (53%) and Rastrigin (43%).

4.4 Comparison with other algorithms

To investigate the performance of the proposed algorithm, the results obtained by hybrid CA-PVS algorithm are compared with the results obtained by other optimization algorithms such as Particle Swarm Optimization (PSO)

No	Fct. Name	Optimal	Mean	STD	Time (s)	SR
f_1	Sphere	0	2,1293E-53	7,05002E-53	0,56	100%
f_3	Ackley	0	0,1226	0,4668	0,67	93%
f_7	Quartic	0	0,0014	0,0013	0,63	47%
f_8	Zakharov	0	1,5411E-34	8,3238E-34	0,59	100%
f_9	Step	0	0	0	0,63	100%
f_{11}	Penalized 1	0	1,1795E-31	6,1402E-34	0,75	100%
f_{12}	Penalized 2	0	0,0014	0,0037	0,70	87%
f_{13}	Himmelblau	-78,3323	-74,5625	4,4446	0,54	53%
f_{14}	Qing	0	3,9443E-31	0	0,55	100%
f_{15}	Mishra 11	0	0	0	0,52	100%
f_{16}	Alpine	0	6,6428E-16	1,5846E-15	0,52	100%
f_{17}	Cosine Mixture	-0,4	-0,3950	0,0269	0,53	97%
f_{19}	Schwefel 2.21	0	2,5566E-19	9,3267E-19	0,56	100%
f_{20}	Schwefel 2.22	0	7,0106E-28	3,3022E-27	0,55	100%
f_{21}	Powell sum	0	1,3493E-61	6,3418E-61	0,59	100%
f_{24}	Dixon Price	0	0,02699	0,1236	0,57	93%
f_{27}	Multimod	0	7,1367E-56	3,8761E-55	0,52	100%
f_{28}	Exponential	-1	-1	0	0,53	100%
f_{29}	Sumsquares	0	2,2718E-50	1,2440E-49	0,55	100%
f_{30}	Brown	0	1,6776E-54	4,5997E-54	0,54	100%

Table 3: Simulation results for Hybrid CA-PVS without cultural exchange for 21 functions on dimension $D = 4$ over 30 runs

[14], Firefly Algorithm(FA) and BAT algorithm (BAT) [15]. Since the algorithms are stochastic in nature, the results of two successive runs usually do not match. Hence, we have taken 30 different independent runs of each algorithm. To compare the performance of the algorithms and to be fair, we also choose to set, in addition to the optima comparison table, a comparative table of computation time in purpose to determine and compare the mean elapsed time by each algorithm while processing a certain function.
In this section, simulations are realized with the functions among (f_1 to f_{30}). The algorithms used in this comparative study do have common inputs such as dimension, population, and the maximum number of iterations. The values of the common parameters are as following:

- General (Common) Settings:
  - Dimension : 3 and 4
  - Population : 20
  - Maximum number of iterations: 200
- BAT Algorithm specific parameters:
  - $F_{max} = 2$; maximum frequency

  - $F_{min} = 0$; minimum frequency
  - A=random(N,1); loudness for each BAT
  - r=random(N,1); pulse emission rate for each BAT
  - $\alpha = 0.5$; constant for loudness update
  - $\gamma = 0.5$; constant for emission rate update
  - ro=0.001; initial pulse emission rate

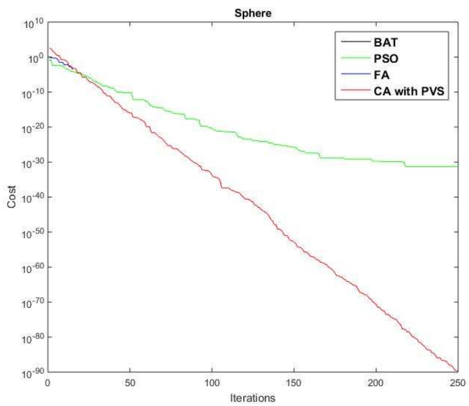
- Firefly Algorithm specific parameters:
  - $\alpha = 0.5$; constant for loudness update
  - $\beta_{min} = 0.5$; constant for loudness update
  - $\gamma = 0.5$; constant for emission rate update

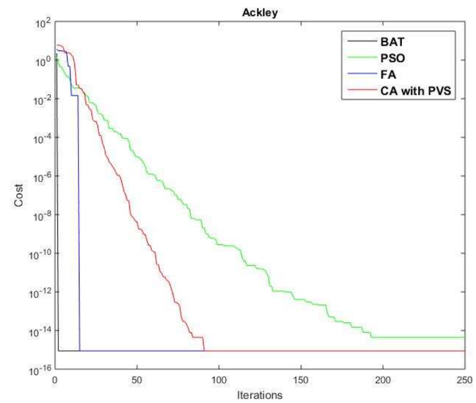
- Particle Swarm Optimization specific parameters:
  - $\alpha = 0.5$; constant for loudness update
  - $\beta_{min} = 0.5$; constant for loudness update
  - $\gamma = 0.5$; constant for emission rate update

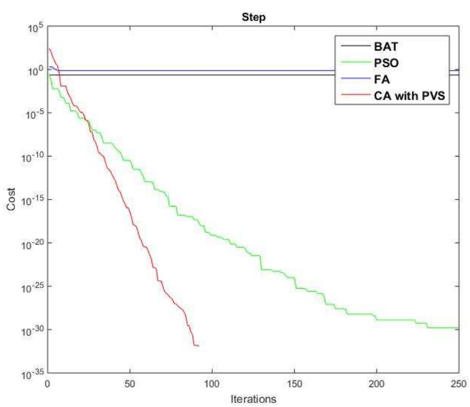
Concerning the success rate (SR), all results coming at most of 10^{-4} from the global optima are considered as a success.

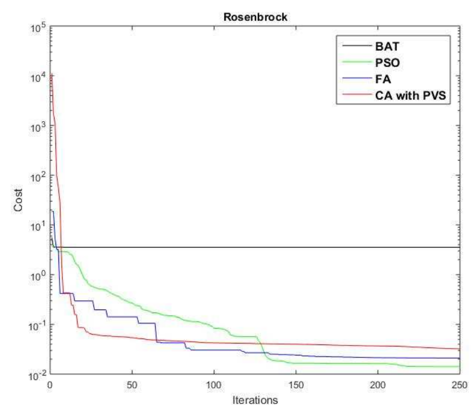
4.4.1 Performance Results

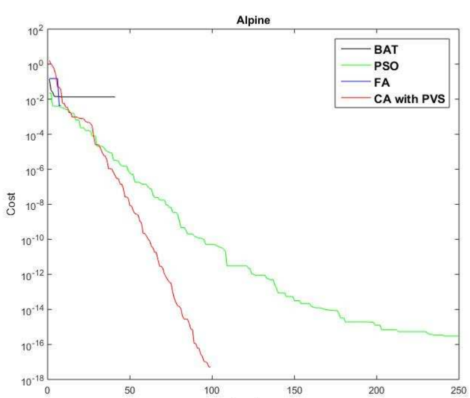
The comparison results for some functions between $(f_1 - f_{30})$ on dimension $D = 3$ are shown in Table **??** and 7.
We see that Hybrid CA-PVS has the highest rate of success at almost all the functions. Although PSO has shown a good performance, our new algorithm is powerful in a way that makes the optimization process avoid being trapped in local optimums. The plots below show that the best cost evolves in most cases in a linear way, and it keeps getting closer to the optima. Although it could take more iterations, but the convergence at the global optimum is guaranteed.

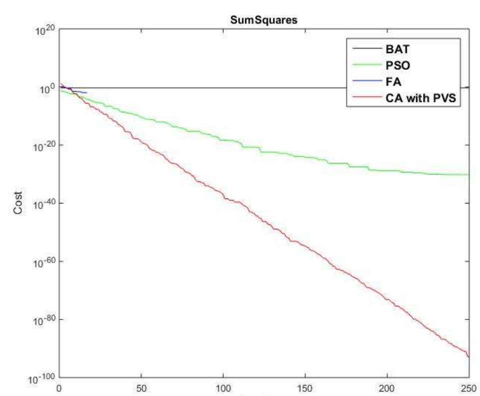

Sphere
BAT
PSO
FA
CA with PVS
Cost
Iterations

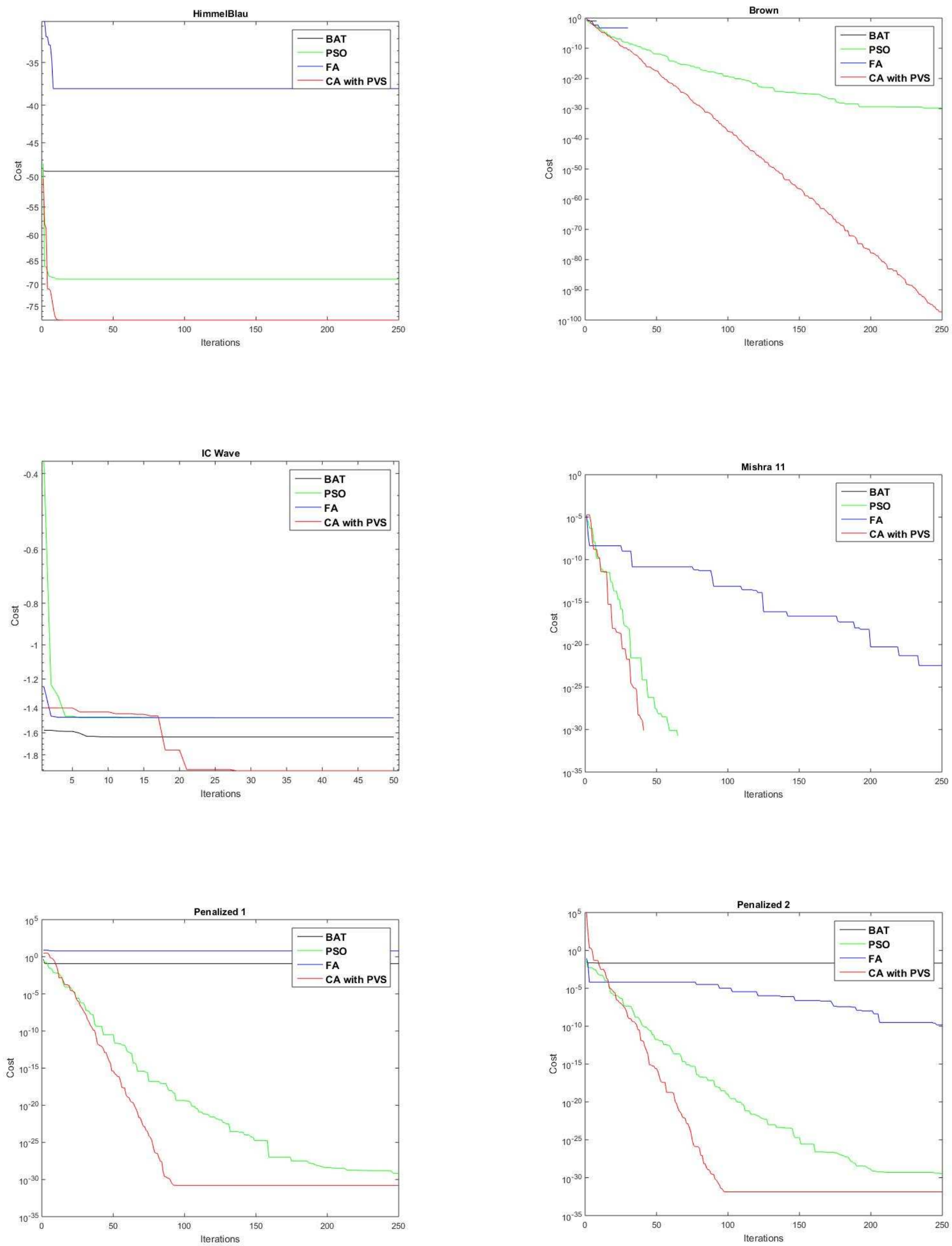

Ackley
BAT
PSO
FA
CA with PVS
Cost
Iterations


Step
BAT
PSO
FA
CA with PVS
Cost
Iterations


Rosenbrock
BAT
PSO
FA
CA with PVS
Cost
Iterations


Alpine
BAT
PSO
FA
CA with PVS
Cost
Iterations


SumSquares
BAT
PSO
FA
CA with PVS
Cost
Iterations


HimmelBlau
BAT
PSO
FA
CA with PVS
Cost
Iterations
Brown
BAT
PSO
FA
CA with PVS
Cost
Iterations
IC Wave
BAT
PSO
FA
CA with PVS
Cost
Iterations
Mishra 11
BAT
PSO
FA
CA with PVS
Cost
Iterations
Penalized 1
BAT
PSO
FA
CA with PVS
Cost
Iterations
Penalized 2
BAT
PSO
FA
CA with PVS
Cost
Iterations

5 Conclusion

Due to the countless problems encountered nowadays, among which some can be extremely difficult to solve, we often find the need to resort to algorithms, and more specifically to meta-heuristics when it comes to optimization problems.
Therefore, this study has consisted on presenting the passing vehicle search algorithm, the cultural algorithm and its performances in comparison with other algorithms in literature such as PSO, BAT, FIREFLY,... Then, representing the Hybridization of the cultural algorithm based on Passing vehicle search algorithm. The Hybrid passing vehicle search algorithm was tested on a set of well known Benchmark functions in order to demonstrate its performance. The proposed algorithm has shown the ability to avoid being trapped in local optima and the stability in all functions that were studied in this paper and therefore can be efficiently used for multivariable function optimization. Furthermore, simulation results show that the proposed algorithm outperforms the other algorithms in terms of success rate of finding global optimum.

References

[1] J. A. Adeyemo, "Reservoir Operation using Multi-objective Evolutionary Algorithms- A Review," In: Asian Journal of Scientific Research, **4,** no. 1, (2011), 16–27.

[2] Poonam Savsani, Vimal Savsani, "PVS: A novel meta-heuristic algorithm", Applied Mathematical Modeling **40,** (2016), 3951–3978.

[3] V. Reynolds, "An introduction to culture algorithms", In: Proceedings of the third annual conference on evolutionary programming, San Diego, (1994), 131–139.

[4] E. B. Tylor, "Primitive Culture: Researches into the Development of Mythology, Philosophy, Religion, Language, Art and Custom", London, J. Murray, (1871).

[5] J. Alami, A El Imrani, A. Bouroumi, "A multi-population cultural algorithm using fuzzy clustering", Applied Soft Computing, (2007).

[6] Deam James Azevedo da Silva, Otavio Noura Teixeira, Roberto Celio Limao de Oliveira, "Performance Study of Cultural Algorithms Based on

Genetic Algorithm with Single and Multi Population for the MKP", Bio-Inspired Computational Algorithms and Their Applications, Shangce Gao (Ed.), (2012).

[7] Yang Sun, Lingbo Zhang, Xingsheng Gu, "A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems", Elsevier, 2012.

[8] Leandro dos Santos Coelho, Viviana Cocco Mariani, "An Efficient Particle Swarm Optimization Approach Based on Cultural Algorithm Applied to Mechanical Design", IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, (2006).

[9] Wei Xie, Chang-ming Ji, Xin-wu Li, "Particle Swarm Optimization Based on Cultural Algorithm for Short-term Optimal Operation of Cascade Hydro power Stations", Fifth International Conference on Natural Computation, (2009).

[10] C. Darwin, "The Origin of Species by Means of Natural Selection; or the Preservation of Favored Races in the Struggle for Life, John Murray, London, 1929.

[11] Robert G. Reynolds, "An introduction to cultural algorithms," Proceeding of the Third Annual Conference on Evolutionary Programming, World Scientific, New Jersey, USA, (1994).

[12] Carlos A. Coello, Ricardo Landa Becerra,"Evolutionary Multiobjective Optimization using a Cultural Algorithm", Proceedings of the 2003 IEEE Swarm Intelligence Symposium, (2003).

[13] Deam James Azevedo da Silva, Roberto Celio Limao de Oliveira, "A Modified Cultural Algorithm based on Genetic Algorithm for Solving Global Optimization Problems", Proceedings of the International Conference on Genetic and Evolutionary Methods, Las Vegas, Nevada, USA, July 13-16, 2009.

[14] J. Kennedy, R. Eberhart, "Particle Swarm Optimization," Proceedings of IEEE International Conference on Neural Networks, (1995), 1942–1948.

[15] Xin-She Yang, "A New meta-heuristic Bat-Inspired Algorithm", J. R. Gonzàlez et al.,(Eds.), Nature Inspired Cooperative Strategies for Optimization, 284, (2010), 65–74.