Oper Res Int J
DOI 10.1007/s12351-017-0320-y

ORIGINAL PAPER

Most Valuable Player Algorithm: a novel optimization algorithm inspired from sport

H. R. E. H. Bouchekara[1]

Received: 2 August 2016 / Revised: 9 February 2017 / Accepted: 27 April 2017

Abstract In this paper a new metaheuristic called the Most Valuable Player Algorithm (MVPA) is proposed for solving optimization problems. The developed algorithm is inspired from sport where players form teams, then these players compete collectively (in teams) in order to win the championship and they compete also individually in order to win the MVP trophy. The performances of MVPA are evaluated on a set of 100 mathematical test functions. The obtained results are compared with the ones obtained using 13 well-known optimization algorithms. These results demonstrate that, the MVPA is a very competitive optimization algorithm, it converges rapidly (with smaller number of functions evaluations) and more successfully (with higher overall success percentage) than the compared algorithms. Therefore, further developments and applications of MVPA would be worth investigating in future studies.

Keywords Optimization · Metaheuristic · Most Valuable Player Algorithm · Sport

Abbreviations

ABC	Artificial Bee Colony
BH	Black Hole
BMO	Bird Mating Optimizer
CBO	Colliding Bodies Optimization
DE	Differential Evolution
DSA	Differential Search Algorithm
E	Experiment

✉ H. R. E. H. Bouchekara
bouchekara.houssem@gmail.com

1 Laboratory of Electrical Engineering of Constantine, LGEC, Department of Electrical Engineering, University of Freres Mentouri Constantine, 25000 Constantine, Algeria

EM	Electromagnetism-like mechanism
FA	Firefly Algorithm
fitness	Fitness (strength or efficiency or rating) of a team
fitnessN	Normalized fitness
FranchisePlayer	Franchise player of a team
$f(\mathbf{x})$	Objective function
$g_i(\mathbf{x})$	Set of equality constraints
GA	Genetic Algorithm
GSA	Gravitational Search Algorithm
$h_j(\mathbf{x})$	Set of inequality constraints
HS	Harmony Search
LCA	League Championship Algorithm
MaxNFix	Maximum number of fixtures
MVP	Most valuable player
MVPA	Most Valuable Player Algorithm
nP	Number of players of one team
$\mathrm{nT_i}$	Number of teams with the same number of players
ObjFunction	The name of the objective function
$\mathrm{Player_i}$	A player in the population
PlayersSize	Number of players in the league (population size)
Pr	Probability
ProblemSize	Problem dimension
PSO	Particle Swarm Optimization
$\mathrm{S_{1,1}}$	Skill
SA	Simulated Annealing
$\mathrm{TEAM_i}$	Groupe of players
TeamsSize	Number of teams in the league
TLBO	Teaching–Learning-Based Optimization
x_k^{min} and x_k^{max}	Domain constraints
$\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$	Vector of design variables

1 Introduction

Optimization is the keystone of modern civilization development; it is everywhere, from engineering design to financial markets, from biology to medicine, from computer sciences to industrial applications and even in our daily activities for example for planning our holidays. It is always a matter of maximizing or minimizing something. In other words, every time we meet a problem we search automatically for the optimal solution of this problem, even though solutions are not always found (Yang 2010b).

Mathematically, an optimization problem can be formulated as follows:

$$\begin{array}{cl} \min\limits_{x \in \mathrm{R}^n} f(\mathbf{x}) & \\ g_i(\mathbf{x}) = 0 & i = 1, \ldots, m \\ h_j(\mathbf{x}) \le 0 & j = 1, \ldots, l \\ x_k^{min} \le x_k \le x_k^{max} & k = 1, \ldots, n \end{array} \tag{1}$$

where: $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ is the vector of design variables, $f(\mathbf{x})$ is the objective function, $g_i(\mathbf{x})$ are the equality constraints, $h_j(\mathbf{x})$ are the inequality constraints and x_k^{min}andx_k^{max} are the domain constraints.

Moreover, real-world optimization problems are often very hard to solve and they become more and more complex. There are two classes of optimization algorithms that can be used to solve optimization problems: classic or conventional algorithms and modern metaheuristics (Yang 2011). The classic algorithms, like the gradient-based algorithms, are deterministic however, if there are some discontinuities in the objective function they do not work well (Yang 2011). Metaheuristics are stochastic and they do not guarantee that global minima are found. However, they can cope with challenging optimization problems (Fister et al. 2013; Ali et al. 2005).

Nature is the most important source of inspiration of metaheuristics (Fister et al. 2013). Some of these algorithms have already gained popularity due to their high efficiency like: Genetic Algorithm (GA) (Holland 1975), Simulated Annealing (SA) (Kirkpatrick et al. 1983), Differential Evolution (DE) (Storn and Price 1997), Particle Swarm Optimization (PSO) (Eberhart and Kennedy 1995; Kennedy and Eberhart 1995), Harmony Search (HS) (Geem et al. 2001) and Artificial Bee Colony (ABC) (Karaboga 2005).

More recently, many other metaheuristics have been developed and applied to different optimization problems. From these algorithms, we can find: Gravitational Search Algorithm (GSA) (Rashedi et al. 2009), Firefly Algorithm (FA) (Yang 2009), Teaching–Learning-Based Optimization (TLBO) (Rao et al. 2011), League Championship Algorithm (LCA) (Husseinzadeh Kashan 2011), Differential Search Algorithm (DSA) (Civicioglu 2012), Black Hole (BH) (Hatamlou 2013), Colliding Bodies Optimization (CBO) (Kaveh and Mahdavi 2014) and Bird Mating Optimizer (BMO) (Askarzadeh 2014).

In addition to the aforementioned and non-exhaustive list of algorithms, a brief review of some nature-inspired algorithms for optimization is given in (Fister et al. 2013). Moreover, 134 innovative clever computational methods are exposed in (Xing and Gao 2014).

As the author of this paper comes from an engineering background, where optimization problems are mostly time-consuming, hence, the aim of this paper is to develop a new optimization algorithm called the Most Valuable Player Algorithm (MVPA) that has the following features:

1. fast; i.e. it converges quickly,
2. efficient and reliable,

The MVPA is inspired from the metaphor of sport where a population of players compete collectively in teams in order to win the leagues' championship, and they compete individually in order to win the MVP trophy.

The remainder of the paper is organized as follows. Section 2 presents in detail the MVPA. In Sect. 3, the experimental study is presented and the results exposed. Finally, the conclusion is drawn in Sect. 4.

2 Most Valuable Player Algorithm

2.1 Sport terminology

Before explaining the MVPA, some sport related terms should be defined.

It is worth pointing that all the definitions of this section are quoted from the Oxford learner's dictionaries (Oxford 2015).

Player: 'a person who takes part in a game or sport'.

Team: 'a group of people who play a particular game or sport against another group of people'.

Franchise player: 'the best or most valuable player on a professional sports team'.

Most valuable player: 'in some US sports, the award and name given to the best player in a game or series of games or during a particular season. The best known are in football, baseball and basketball. The players given the award are usually chosen by sports journalists'.

League: 'a group of sports teams who all play each other to earn points and find which team is best'.

Championship: 'a competition to find the best player or team in a particular sport'.

Fixture: 'a sports event that has been arranged to take place on a particular date and at a particular place'.

2.2 Overview

Like other metaheuristics, MVPA exploits a population to evolve to optimality. The population is considered as a group of players that have skills, which is analogous to the design variables where the number of players' skills corresponds to the dimension of the problem. It is noteworthy to mention that, strictly speaking, from the player point of view, higher skills are desired, but higher design variables are not always condition for optimizing a given problem. However, for simplicity we assume that the design variables are analogous to the skills of players. In Fig. 1, we show an example of two players with the corresponding level of skills for each. An example of skills in sport is endurance which is the ability to continue to perform an action for long periods of time like Haile Gebreselassie who won two Olympic gold medals over 10,000 m.Therefore, each player is represented as follows:

$$\mathrm{Player_k} = [\, \mathrm{S_{k,1}} \quad \mathrm{S_{k,2}} \quad \ldots \quad \mathrm{S_{k,ProblemSize}} \,] \tag{2}$$

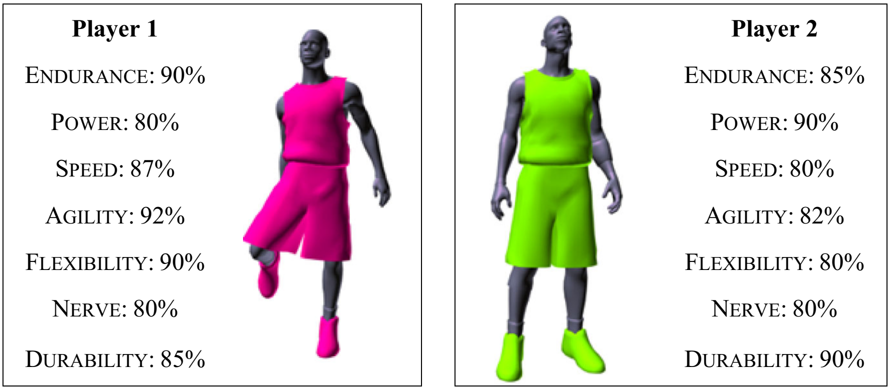

Fig. 1 Example of two players with their skills

and a team, which is composed of a group of players, is represented as follows:

$$\mathrm{TEAM_i} = \begin{bmatrix} \mathrm{Player}_1 \\ \mathrm{Player}_2 \\ \vdots \\ \mathrm{Player}_{\mathrm{PlayersSize}} \end{bmatrix} \tag{3}$$

Or

$$\mathrm{TEAM_i} = \begin{bmatrix} S_{1,1} & S_{1,2} & \cdots & S_{1,\mathrm{ProblemSize}} \\ S_{2,1} & S_{2,2} & \cdots & S_{2,\mathrm{ProblemSize}} \\ \vdots & \vdots & \cdots & \vdots \\ S_{\mathrm{PlayersSize},1} & S_{\mathrm{PlayersSize},2} & \cdots & S_{\mathrm{PlayersSize},\mathrm{ProblemSize}} \end{bmatrix} \tag{4}$$

where: S stands for skill, PlayersSize is the number of players in the league and ProblemSize is the dimension of the problem.

The efficiency (or the rating or the strength) of a player or a team is analogous to fitness in other algorithms. Each team has a franchise player (i.e., its best player so far) and the league's MVP is the best player of the league (the player who has the best solution so far).

2.3 The Algorithm

The following sections explain in detail the main phases or steps of the MVPA.

2.3.1 Initialization

In the initialization step, a population of 'PlayersSize' players is randomly generated in the search space (Fig. 2).

Fig. 2 Generation of a population of players in the initialization stage illustration

Fig. 3 Teams formation illustration

2.3.2 Teams formation

Once the population of players is generated they are randomly distributed to form 'TeamsSize' teams (Fig. 3).

In order to add more flexibility to the MVPA, the number of teams does not need to be necessarily a divisor of the number of players. Therefore, the user has more freedom to select the number of players and teams he wants to. For example, if the MVPA is used for a given problem where in the literature another algorithm has been used to optimize this problem using a specific population size, the user can select the same size (for a fair comparison). Another example is that if a user wants to change the population size during the optimization process he does not have to modify the number of teams' every time since the algorithm can handle this issue. Many other examples can be given in order to show how much flexibility is brought when the number of teams does not need to be necessarily a divisor of the number of players.

To form teams, the first 'nT_1' teams have 'nP_1' players, while the remaining 'nT_2' teams have 'nP_2' players where nP_1, nP_2, nT_1 and nT_2 are calculated using the following expressions:

$$nP1 = \text{ceil}\left(\frac{\text{PlayersSize}}{\text{TeamsSize}}\right) \tag{5}$$

$$nP2 = nP1 - 1 \tag{6}$$

$$nT1 = \text{PlayersSize} - nP2 \times \text{TeamsSize} \tag{7}$$

$$nT2 = \text{TeamsSize} - nT1 \tag{8}$$

where: ceil is a function that rounds a real number to the smallest following integer.

To illustrate the concept of how teams are formed let's take some examples.

Example 1 A league with 4 teams and 20 players.

In this first example, the population has the following data: PlayersSize = 20 and TeamsSize = 4. Therefore, all teams must have 5 players randomly distributed among them as follows: players number {3, 12, 17, 18, 19} form TEAM_1, players number {2, 6, 11, 16, 20} form TEAM_2, players number {4, 8, 10, 14, 15} form TEAM_3 and players number {1, 5, 7, 9, 13} form TEAM_4.

Example 2 A league with 3 teams and 20 players.

In this second example, the population has the following data: PlayersSize = 20 and TeamsSize = 3. Hence, the first 2 teams must have 7 players and the last team must have 6 players, randomly distributed among them as follows: players number {2, 3, 6, 12, 17, 18, 19} form TEAM_1, players number {4, 8, 11, 14, 15, 16, 20} form TEAM_2 and players number {1, 5, 7, 9, 10, 13} form TEAM_3.

2.3.3 Competition phase

After the initialization, comes the competition phase. In this phase, players try to improve their skills individually to be better players and then as teams, they play against each other. Therefore, a loop over all teams will select teams one by one (i.e., for i = 1:TeamsSize) and the selected team will follow two steps that are individual competition and team competition.

2.3.3.1 Individual competition It is legitimate that each player aims to be his team's franchise player and the league's MVP. Thus, he tries to improve his skills (in training for example) compared to his team's franchise player and to the league's MVP. Therefore, the players' skills of the selected TEAM_i are updated as follows:

$$\begin{aligned}\text{TEAM}_i = {} & \text{TEAM}_i + \text{rand} \times (\text{FranchisePlayer}_i - \text{TEAM}_i) + 2 \times \text{rand} \\ & \times (\text{MVP} - \text{TEAM}_i)\end{aligned} \tag{9}$$

where rand is a uniformly distributed random number between 0 and 1.

It is worth mentioning that, the constant 2 has been selected after several tests and it gives excellent results for the tested optimization problems. However, for a specific optimization problem the user can change the constant 2 by another constant or even change the term '2 × rand' by another random number with a different distribution like normal distribution for example.

2.3.3.2 Team competition In this phase, for the selected TEAM_i, another team TEAM_j is randomly selected where ($i \neq j$) and they play against each other. The outcome of this fixture or this game is the win of one team over the other (there are no tie games). The mechanism of how a winning team is determined, is described below.

Suppose that two teams referred to as TEAM_r and TEAM_y are playing against each other, and TEAM_r has a winning percentage of r %, while TEAM_y has y % of winning percentage. If TEAM_r plays against TEAM_y, the two questions that should be answered are:

1. What is the probability that a team will win?
2. Which team actually wins?

To answer the first question a simple principle of combining probabilities is used (Brown 2015). If we need to assign a probability that a team will win and the only information available is r and y, it is clear that the answer must assume the probability Pr of TEAM_r beating TEAM_y is some function of r and y. Thus, we need a function F(r, y) such that:

$$\Pr\left\{\text{TEAM}_\text{r} \text{ beats } \text{TEAM}_\text{y}\right\} = F(r, y) \tag{10}$$

It follows that:

$$F(r, y) + F(y, r) = 1 \tag{11}$$

and $0 \leq F(x,y) \leq 1$ for any x,y in [0,1]. One class of functions that satisfies this requirement is

$$F(r, y) = \frac{f(r)}{f(r) + f(y)} \tag{12}$$

where f is any mapping from [0, 1] to [0, $+\infty$].

For instance, suppose that r = 0.2 and y = 0.7. Taking:

$$f(x) = x \tag{13}$$

Therefore, TEAM_r has 22.22% chance towin while TEAM_y has 77.78% chance to win.More generally, if:

$$f(x) = x^k \tag{14}$$

and the exponent k tends to 0, the probabilities tend to 50/50, whereas with k greater than 1 the probability of TEAM_r winning goes to zero as illustrated in Table 1.

By analogy, in the MVPA the fitness values of all teams are normalized. The normalization of the fitness of a given team can be done as follows:

$$\text{fitnessN}(\text{TEAM}_\text{i}) = \text{fitness}(\text{TEAM}_\text{i}) - \min(\text{fitness}(\text{All Teams})) \tag{15}$$

Table 1 The influence of the exponent k on the probability that a team will win with $r = 0.2$ and $y = 0.7$

k	% of winning	
	$TEAM_r$	$TEAM_y$
0	50.00	50.0
1	22.22	77.8
2	7.55	92.5
3	2.28	97.7
4	0.66	99.3
5	0.19	99.8
6	0.05	99.9
7	0.02	100.0
8	0.00	100.0
9	0.00	100.0
10	0.00	100.0

then, the probability that $TEAM_r$ beats $TEAM_y$ is calculated using the following formula:

$$\Pr\left\{TEAM_r \text{ beats } TEAM_y\right\} = 1 - \frac{(\text{fitnessN}(TEAM_r))^k}{(\text{fitnessN}(TEAM_r))^k + \left(\text{fitnessN}\left(TEAM_y\right)\right)^k} \tag{16}$$

In our implementation of the MVPA, in order to evaluate the percentage of winning of a team, the exponent k is selected as 1. Furthermore, it is worth pointing that, in MVPA the strength or fitness of a team is assumed to be the fitness of the franchise player of that team. One can ask why the team fitness is not selected as the average fitness of the players of this team. The answer to this question is that in the present version of the MVPA the players are randomly selected to form teams and not the players that are near to each other in the search space form a team. Therefore, taking the average fitness of players as the team's fitness is not correct. However, if the teams were formed using players that are neighbors in the search space, taking the fitness of the team as the average fitness would be more appropriate.

Once the probability of winning is determined, the second question must be answered. Because even though $TEAM_r$ has a higher winning probability than $TEAM_y$, this last one still has a chance to win and this is what makes sport amazing and fantastic, it is not an exact science. Thus, in order to determine which team wins between $TEAM_r$ and $TEAM_y$, a random number is generated if this number is superior to the higher probability between r and y, then the team with the lower probability of winning wins, otherwise, the team with the higher probability of winning wins.

Moreover, in the MVPA there are no tie games, as in basketball for example; if the score is tied at the end of regular time, the teams play multiple 5-min overtime periods in order to determine a winner. In the MVPA, if two teams have the same strength or fitness they will have the same winning probability. Hence, a random number is generated; if this number is higher than 0.5 the first team wins, otherwise the second team wins.

Finally, in the team competition phase, if $TEAM_i$ is selected and it plays against $TEAM_j$, if $TEAM_i$ wins the players' skills of $TEAM_i$ are updated using the following expression:

$$\mathrm{TEAM_i} = \mathrm{TEAM_i} + \mathrm{rand} \times \left(\mathrm{TEAM_i} - \mathrm{FranchisePlayer_j}\right) \tag{17}$$

Otherwise, the players' skills of $TEAM_i$ are updated using the following expression:

$$\mathrm{TEAM_i} = \mathrm{TEAM_i} + \mathrm{rand} \times \left(\mathrm{FranchisePlayer_j} - \mathrm{TEAM_i}\right) \tag{18}$$

It is worth to mention that, in the competition phase, the population is checked to see if there are players outside the bounding box of the population. If a skill of a generated player crosses a bound of the search space it takes the value of this bound (i.e. if it crosses the lower bound it takes the value of the lower bound likewise if it crosses the upper bound it takes the value of the upper bound).

Let us take an example to illustrate how the winning team in a duel of two teams is determined.

Example 3 Suppose that we have 6 teams that have the following strengths or fitnesses: 25, 9, 1, 4, 16 and 25. Therefore, using (15) the normalized fitnesses are: 24, 8, 0, 3, 15 and 24.

If $TEAM_1$ plays against $TEAM_5$, using (16) we find that $\Pr\{TEAM_1 \text{ beats } TEAM_5\} = 0.3846$ and $\Pr\{TEAM_5 \text{ beats } TEAM_1\} = 0.6154$. In addition, a random number is generated if this number is higher than 0.6154 then $TEAM_5$ wins otherwise, $TEAM_1$ wins.

Now, if $TEAM_1$ plays against $TEAM_6$, since both teams have the same winning probability, a random number is generated, if this number is higher than 0.5 then $TEAM_1$ wins, otherwise $TEAM_6$ wins.

2.3.4 Application of greediness

After that, a greediness process is applied. In other words, a comparison between the population before and after the competition phase is made and a new solution is accepted if it gives a better objective function value than the initial one.

2.3.5 Application of elitism

In this phase, the worst players are replaced by the best ones. The number of elite players is selected as the third of the PlayersSize (obviously, this can be changed).

2.3.6 Remove duplicates

In this phase if two successive players in the population are exactly the same, the second player is replaced by another one using the same procedure described in (Elsayed et al. 2014).

2.3.7 Termination criterion

The algorithm iterates for a number of fixtures specified by MaxNFix (maximum number of fixtures). In this version of the MVPA, this criterion is chosen as the stopping criterion however, it is obvious that other stopping criteria can be easily implemented by the user. For instance, another stopping criterion that can be implemented is the number of iterations or the amount of iterations performed without replacing the current MVP. However, this decision has to be studied carefully since the algorithm may be stopped before converging to the global optimum. On the other hand, unnecessary function evaluations may be avoided by stopping earlier.

The pseudocode of the MVPA is given in Algorithm 1.

Algorithm 1: MVPA pseudocode.

Inputs ObjFunction (objective function), ProblemSize (dimension of the problem), PlayersSize (number of players), TeamsSize (number of teams) and MaxNFix (maximum number of fixtures)

Output MVP

Initialization

for fixture=1: MaxNFix

    for i=1:TeamsSize

        $\mathrm{TEAM_i}$ = Select the team number i from the league's teams

        $\mathrm{TEAM_j}$ = Randomly select another team j from the league's teams where j≠i

        $\mathrm{TEAM_i = TEAM_i + rand \times (FranchisePlayer_i - TEAM_i) + 2 \times rand \times (MVP - TEAM_i)}$

        if $\mathrm{TEAM_i}$ wins against $\mathrm{TEAM_j}$

            $\mathrm{TEAM_i = TEAM_i + rand \times \left(TEAM_i - FranchisePlayer_j\right)}$

        else

            $\mathrm{TEAM_i = TEAM_i + rand \times \left(FranchisePlayer_j - TEAM_i\right)}$

        end if

    Check if there are players outside the search space

    end for

    Application of greediness

    Application of elitism

    Remove duplicates

end for

2.4 Illustrative example

In this section a detailed demonstration of MVPA is given.

Step 1 Definition of the optimization problem and initialization of optimization parameters

ObjFunction = Sphere function $f_{\text{Sphere}}(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$

ProblemSize = 2 (dimensions). Thus, each player of the population will have 2 skills S_1 and S_2, respectively.

Bounds = $[-5.12 \leq S_1 \leq 5.12]$, $[-5.12 \leq S_2 \leq 5.12]$,

PlayersSize = 20

TeamsSize = 5

MaxNFix = 10

Step 2 Initialization of the population

In this step, a population of 20 players with 2 dimensions is randomly generated. The obtained population is displayed in Table 2.

Step 2: Teams formation

Since there are 20 players and 5 teams, therefore, each team would have 4 players randomly selected from the initial population. Furthermore, the franchise player of each team and the league's MVP are determined. In this example, the MVP is the player number 8, he has the following skills (0.4801, −1.1036) and he has an

Table 2 Initialization of the population

Players	S_1	S_2	fitness
1	3.2228	1.5948	12.9296
2	4.1553	−4.7543	39.8701
3	−3.8197	3.5751	27.371
4	4.233	4.4441	37.668
5	1.3554	1.8302	5.1868
6	−4.1212	2.6393	23.9499
7	−2.2682	2.4897	11.3431
8	0.4801	−1.1036	1.4484
9	4.6849	1.5921	24.4828
10	4.7605	−3.367	33.999
11	−3.506	2.1099	16.7441
12	4.8189	−4.794	46.2042
13	4.6814	−2.2843	27.1335
14	−0.1498	−4.6472	21.6189
15	3.0749	−4.1254	26.4735
16	−3.6671	3.3122	24.4182
17	−0.8012	1.995	4.6221
18	4.2571	−1.8729	21.6309
19	2.9922	4.6103	30.2079
20	4.7052	−4.7673	44.8658

Table 3 Initialization of the population

Teams	Players				FranchisePlayer	FranchisePlayer's fitness
$TEAM_1$	8	9	16	17	8 (MVP)	1.4484
$TEAM_2$	3	10	11	14	11	16.7441
$TEAM_3$	4	13	15	18	18	21.6309
$TEAM_4$	1	2	7	12	7	11.3431
$TEAM_5$	5	6	19	20	5	5.1868

Table 4 Update of $TEAM_1$ in the competition phase

Teams	Players	Initial population		Population after individual competition		Population after team competition	
		S_1	S_2	S_1	S_2	S_1	S_2
$TEAM_1$	8	0.4801	−1.1036	0.4801	−1.1036	2.1731	−3.3172
	9	4.6849	1.5921	−3.1051	−3.8632	−3.6206	−7.7768
	16	−3.6671	3.3122	3.1858	0.5425	6.5457	−0.6571
	17	−0.8012	1.995	2.525	0.4607	5.4779	−0.7893

efficiency or a rating of (1.4484). Table 3 shows the formation of the 5 teams with the Franchise player of each team and the MVP.

Step 3 Competition phase

In this phase, players try to improve their skills individually to be better players and then as teams, they play against each other.

Step 3.1: Individual competition

For i = 1, the players' skills of $TEAM_1$ are updated using (9) and the updated players' skills are given in Table 4.

Step 3.2: Team competition

In this step, a second team is randomly selected among the league's teams to face the first team selected inside the loop. Obviously, the second team has to be different from the first one. Thus, for i = 1, $TEAM_1$ will play against $TEAM_4$ (randomly selected).

As aforementioned, the normalized fitness of each team is calculated using (15). Then, the probability that $TEAM_1$ beats $TEAM_4$ and the one that $TEAM_4$ beats $TEAM_1$ are calculated using (16). It is found in this case that, $\Pr\{TEAM_1 \text{ beats } TEAM_4\} = 1$ and $\Pr\{TEAM_4 \text{ beats } TEAM_1\} = 0$. Therefore, in this duel $TEAM_1$ wins.

The same procedure is repeated for the remaining teams: $TEAM_2$, $TEAM_3$, $TEAM_4$, and $TEAM_5$. It is worth mentioning that, in this step the population is checked to see if there are some players outside the upper and lower limits of each design variable and the obtained population is given in Table 5.

Table 5 Evolution of population

Players	Initial population			Population after competition			Population after greediness			Population after elitism and remove duplicates		
	S_1	S_2	Fitness	S_1	S_2	Fitness	S_1	S_2	Fitness	S_1	S_2	Fitness
1	3.2228	1.5948	12.9296	2.1977	1.9388	8.5884	2.1977	1.9388	8.5884	2.1977	1.9388	8.5884
2	4.1553	−4.7543	39.8701	−1.2554	5.12	27.7904	−1.2554	5.12	27.7904	−2.2682	2.4897	11.3431
3	−3.8197	3.5751	27.371	−2.992	−5.12	35.1662	−3.8197	3.5751	27.371	−3.506	2.1099	16.7441
4	4.233	4.4441	37.668	1.0518	−1.4942	3.3388	1.0518	−1.4942	3.3388	1.0518	−1.4942	3.3388
5	1.3554	1.8302	5.1868	−1.4979	0.5752	2.5745	−1.4979	0.5752	2.5745	−1.4979	0.5752	2.5745
6	−4.1212	2.6393	23.9499	−5.12	−0.9347	27.0881	−4.1212	2.6393	23.9499	−4.1212	2.6393	23.9499
7	−2.2682	2.4897	11.3431	−1.2687	−2.6744	8.7621	−1.2687	−2.6744	8.7621	−1.2687	−2.6744	8.7621
8	0.4801	−1.1036	1.4484	2.1731	−3.3172	15.7263	0.4801	−1.1036	1.4484	0.4801	−1.1036	1.4484
9	4.6849	1.5921	24.4828	−3.6206	−5.12	39.3235	4.6849	1.5921	24.4828	4.6849	1.5921	24.4828
10	4.7605	−3.367	33.999	−5.12	4.1035	43.0534	4.7605	−3.367	33.999	−0.8012	1.995	4.6221
11	−3.506	2.1099	16.7441	−0.294	−3.1714	10.1445	−0.294	−3.1714	10.1445	−0.294	−3.1714	10.1445
12	4.8189	−4.794	46.2042	−5.12	3.3066	37.1482	−5.12	3.3066	37.1482	0.4801	−1.1036	1.4484
13	4.6814	−2.2843	27.1335	1.6555	−1.4246	4.7702	1.6555	−1.4246	4.7702	1.6555	−1.4246	4.7702
14	−0.1498	−4.6472	21.6189	−1.4669	3.1896	12.3255	−1.4669	3.1896	12.3255	−1.4669	3.1896	12.3255
15	3.0749	−4.1254	26.4735	0.1834	0.1307	0.0507	0.1834	0.1307	0.0507	0.1834	0.1307	0.0507
16	−3.6671	3.3122	24.4182	5.12	−0.6571	26.6462	−3.6671	3.3122	24.4182	−3.6671	3.3122	24.4182
17	−0.8012	1.995	4.6221	5.12	−0.7893	26.8374	−0.8012	1.995	4.6221	−0.8012	1.995	4.6221
18	4.2571	−1.8729	21.6309	1.1384	−1.272	2.9139	1.1384	−1.272	2.9139	1.1384	−1.272	2.9139
19	2.9922	4.6103	30.2079	−4.4577	−2.7935	27.6747	−4.4577	−2.7935	27.6747	3.2228	1.5948	12.9296
20	4.7052	−4.7673	44.8658	−4.1010	3.9285	32.2510	−4.1010	3.9285	32.2510	1.3554	1.8302	5.1868

Step 4: Application of greediness

As previously mentioned, in this step a new solution is accepted only if it gives a better objective function value than the initial one. The updated players' skills after this step are given in Table 5.

Step 5: Application of elitism

In this step, the worst players in the population are replaced with the best ones. The updated players' skills after this step are given in Table 5.

Step 6: Remove duplicates

The last step is to remove duplicates from the population and the obtained population after the first iteration is given in Table 5.

3 Experiment study

The objective of this section is to assess the performances of the developed algorithm. To this end, the MVPA has been tested using 100 benchmarks or test functions. Furthermore, 4 experiments noted as E1, E2, E3 and E4 are conducted. For E1 and E2 the whole set of test function is considered, and the allowed number of function evaluations is selected as 2000 for E1 and as 5000 for E2. The goal here is to test the rapidity of the MVPA which is a suitable feature required for almost all engineering applications. For E3 and E4 only test functions where the dimension can be increased are considered. Therefore, 33 out of the 100 test functions are selected. The considered dimensions are 10 and 25 while the allowed maximum number of functions evaluations is selected as 10,000 and 25,000 for E3 and E4, respectively. The goal here is to test the scalability of the MVPA, in other words how the MVPA cope with higher dimension problems. Finally, the obtained results are compared with 13 well-known optimization algorithms.

For all experiments PlayersSize = 100 and TeamsSize = 20.

3.1 Test Procedure

The test procedure is executed as follows (Gavana 2015):

1. Every optimization algorithm is tested on all considered benchmarks, using 100 different runs or trials (i.e. using different random initial populations).
2. No tuning of the internal parameters of the optimization algorithms is allowed: all the algorithms are run with initial settings (the best settings found in the literature), regardless of the test problem, the dimension of the problem, the type of the problem (i.e., unimodal or multimodal), the starting generation or any other consideration.
3. The maximum number of functions evaluations is set to 2000, 5000, 10,000 and 25,000 for E1, E2, E3 and E4, respectively. If this limit is exceeded, the test is considered as "failed".

4. All the benchmarks treated in this paper have known global optimum values. An algorithm is considered as successful if the following condition is fulfilled:

$$\left|F_{\text{known minimum}} - F_{\text{algorithm minimum}}\right| \leq 10^{-6} \tag{19}$$

5. All the data and the results obtained like the percentage of success and the number of functions evaluations, are collected for statistical analysis afterward.

3.2 Test functions

There are many benchmarks or test functions that can be used in order to evaluate the performances of optimization algorithms in the literature. In this work, we have selected 100 test functions with different: dimensionalities, complexities and types. Table 6 shows a brief description of the 100 test functions with some features like name (taken from Gavana 2015), dimension, and expression. More details about these test functions can be found in Gavana (2015), Jamil and Yang (2013), Mishra (2013) and Adorio and Diliman (2005).

3.3 Tested Algorithms

As said earlier the MVPA has been compared with 13 optimization algorithms. A brief description of each algorithm is given below.

3.3.1 Genetic Algorithm (GA)

GA was first used by Holland (1975). GA is the most famous global optimization method and it is based on the Darwin's theory about evolution. It uses a population of individuals or chromosomes that evolve to better solutions using the three genetic operations which are: selection, crossing and mutation. The selection aims to select individuals from the population based on their fitness, the crossover combines the features of two parent chromosomes to generate two new offsprings and mutation introduces diversity in the population (Tuncer and Yildirim 2012).

3.3.2 Simulated Annealing (SA)

SA which is credited to Kirkpatrick and colleagues is inspired by the annealing process of metals (Kirkpatrick et al. 1983). SA is a trajectory-based search algorithm, it uses a single agent (or solution) in order to explore the search space in a piecewise style. It starts from an initial solution at a high temperature, and gradually cooling down the metal. A new solution is accepted in two cases; (1) if it is better than the previous one and (2) with a probability. Iteration after iteration, the metal is cooled down slowly enough to reach the global optimum.

Table 6 The 100 test functions used to evaluate the performances of the MVPA

#	Name	Expression	Dimension			
			E1	E2	E3	E4
1	Ackley	$f_{\text{Ackley}}(\mathbf{x}) = -20\mathrm{e}^{\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)} - \mathrm{e}^{\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)} + 20 + \mathrm{e}$	2	2	10	25
2	Adjiman	$f_{\text{Adjiman}}(\mathbf{x}) = \cos(x_1)\sin(x_2) - \frac{x_1}{(x_2^2+1)}$	2	2	–	–
3	Alpine	$f_{\text{Alpine}}(\mathbf{x}) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	2	2	10	25
4	AMGM	$f_{\text{AMGM}}(\mathbf{x}) = \left(\frac{1}{n}\sum_{i=1}^{n}x_i - \sqrt[n]{\prod_{i=1}^{n}x_i}\right)^2$	2	2	10	25
5	Beale	$f_{Beale}(\mathbf{x}) = (x_1x_2 - x_1 + 1.5)^2 + \left(x_1x_2^2 - x_1 + 2.25\right)^2 + \left(x_1x_2^3 - x_1 + 2.625\right)^2$	2	2	–	–
6	Bird	$f_{Bird}(\mathbf{x}) = (x_1 - x_2)^2 + e^{[1-\sin(x_1)]^2}\cos(x_2) + e^{[1-\cos(x_2)]^2}\sin(x_1)$	2	2	–	–
7	Bohachevsky	$f_{\text{Bohachevsky}}(x) = \sum_{i=1}^{n-1}\left[x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7\right]$	2	2	10	25
8	BoxBetts	$f_{\text{BoxBetts}}(\mathbf{x}) = \sum_{i=1}^{k} g(x_i)^2$	3	3	–	–
		where: $g(x) = e^{-0.1(i+1)x_1} - e^{-0.1(i+1)x_2} - \left[\left(e^{-0.1(i+1)x_1}\right) - e^{-(i+1)}x_3\right]$ and $k = 10$				
9	Branin	$f_{\text{Branin}}(\mathbf{x}) = \left(-1.275\frac{x_1^2}{\pi^2} + 5\frac{x_1}{\pi} + x_2 - 6\right)^2 + \left(10 - \frac{5}{4\pi}\right)\cos(x_1) + 10$	2	2	–	–
10	Bukin4	$f_{\text{Bukin4}} = 100x_2^2 + 0.01\lvert x_1 + 10\rvert$	2	2	–	–
11	Bukin6	$f_{\text{Bukin6}} = 100\sqrt{\lvert x_2 - 0.01x_1^2\rvert} + 0.01\lvert x_1 + 10\rvert$	2	2	–	–
12	CarromTable	$f_{\text{CarromTable}}(x) = -\frac{1}{30}e^{2\left\lvert 1-\frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\rvert}\cos^2 x_1 \cos^2 x_2$	2	2	–	–
13	Chichinadze	$f_{\text{Chichinadze}}(\mathbf{x}) = x_1^2 - 12x_1 + 8\sin\left(\frac{5}{2}\pi x_1\right) + 10\cos\left(\frac{1}{2}\pi x_1\right) + 11 - 0.2\frac{\sqrt{5}}{e^{\frac{1}{2}(x_2-0.5)^2}}$	2	2	–	–

Table 6 continued

#	Name	Expression	Dimension			
			E1	E2	E3	E4
14	Cigar	$f_{\text{Cigar}}(\mathbf{x}) = x_1^2 + 10^6 \sum_{i=1}^{n} x_i^2$	2	2	10	25
15	Colville	$f_{\text{Colville}}(\mathbf{x}) = (x_1-1)^2 + 100\left(x_1^2 - x_2\right)^2 + 10.1(x_2-1)^2 + (x_3-1)^2 + (x_3-1)^2 + 90\left(x_3^2 - x_4\right)^2 + 10.1(x_4-1)^2 + 19.8\frac{x_4-1}{x_2}$	4	4	–	–
16	Corana	$f_{\text{Corana}}(\mathbf{x}) = \begin{cases} \sum_{i=1}^{n} 0.15 d_i [z_i - 0.05\text{sgn}(z_i)]^2 & \text{if} \lvert x_i - z_i \rvert < 0.05 \\ d_i x_i^2 & \text{otherwise} \end{cases}$ where: $z_i = 0.2\left\lvert\frac{x_i}{s_i}\right\rvert + 0.49999\text{sgn}(x_i), d_i = (1, 1000, 10, 100)$	4	4	–	–
17	CrossInTray	$f_{\text{CrossInTray}}(\mathbf{x}) = -0.0001\left(\left\lvert e^{\left\lvert 100 - \frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\rvert} \sin(x_1)\sin(x_2)\right\rvert + 1\right)$	2	2	–	–
18	CrossLegTable	$f_{\text{CrossLegTable}}(\mathbf{x}) = -\frac{1}{\left(\left\lvert e^{\left\lvert 100 - \frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\rvert} \sin(x_1)\sin(x_2)\right\rvert + 1\right)^{0.1}}$	2	2	–	–
19	CrownedCross	$f_{\text{CrownedCross}}(\mathbf{x}) = 0.0001\left(\left\lvert e^{\left\lvert 100 - \frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\rvert} \sin(x_1)\sin(x_2)\right\rvert + 1\right)^{0.1}$	2	2	–	–
20	Decanomial	$f_{\text{Decanomial}}(\mathbf{x}) = 0.001\left(\left\lvert x_2^4 + 12x_2^2 + 54x_2^2 + 108x_2 + 81.0\right\rvert + \left\lvert x_1^{10} - 20x_1^9 + 180x_1^8 - 960x_1^7 + 3360x_1^6 - 8064x_1^5 + 133340x_1^4 - 15360x_1^3 + 11520x_1^2 - 5120x_1 + 2624.0\right\rvert\right)^2$	2	2	–	–
21	DCS	$f_{\text{DCS}}(\mathbf{x}) = 0.1\sum_{i=1}^{n}\left[(x_i-\alpha)^2 - \cos\left(K\sqrt{\sum_{i=}^{n}(x_i-\alpha)^2}\right)\right]$ where: $K = 5\text{and}\alpha = 5.$	2	2	10	25
22	DixonPrice	$f_{\text{DixonPrice}}(\mathbf{x}) = (\mathrm{x_i} - 1)^2 + \sum_{i=2}^{n} i\left(2x_i^2 - x_{i-1}\right)^2$	2	2	10	25

Table 6 continued

#	Name	Expression	Dimension E1	E2	E3	E4
23	DropWave	$f_{\text{DropWave}}(\mathbf{x}) = -\frac{1+\cos\left(12\sqrt{\sum_{i=1}^{n} x_i^2}\right)}{2+0.5\sum_{i=1}^{n} x_i^2}$	2	2	10	25
24	Easom	$f_{\text{Easom}}(\mathbf{x}) = a - \frac{a}{e^{b}\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}} + e - e^{\frac{\sum_{i=1}^{n}\cos(cx_i)}{n}}$	2	2	10	25
		where: $a = 20$, $b = 0.2$ and $c = 2\pi$				
25	EggHolder	$f_{\text{EggHolder}}(\mathbf{x}) = -x_1 \sin\left(\sqrt{\lvert x_1 - x_2 - 47\rvert}\right) - (x_2 + 47)\sin\left(\sqrt{\left\lvert \frac{1}{2}x_1 + x_2 + 47\right\rvert}\right)$	2	2	–	–
26	Exp2	$f_{\text{Exp2}}(x) = \sum_{i=0}^{9}\left(e^{-i\frac{x_1}{10}} - 5e^{-i\frac{x_2}{10}} - e^{-i\frac{1}{10}} + 5e^{-i}\right)^2$	2	2	–	–
27	FreudensteinRoth	$F_{\text{FreudensteinRoth}}(\mathbf{x}) = \{x_1 - 13 + [(5 - x_2)x_2 - 2]x_2\}^2 + \{x_1 - 29 + [(x_2 + 1)x_2 - 14]x_2\}^2$	2	2	–	–
28	Gear	$f_{\text{Gear}}(\mathbf{x}) = \left\{\frac{1.0}{6.931} - \frac{x_1 x_2}{x_3 x_4}\right\}^2$	4	4	–	–
29	Giunta	$f_{\text{Giunta}}(\mathbf{x}) = 0.6 + \sum_{i=1}^{n}\left[\sin^2\left(1 - \frac{16}{15}x_i\right) - \frac{1}{50}\sin\left(4 - \frac{64}{15}x_i\right) - \sin\left(1 - \frac{16}{15}x_i\right)\right]$	2	2	–	–
30	GoldsteinPrice	$f_{\text{GoldsteinPrice}}(\mathbf{x}) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right]\left[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_1^2 - 36x_1x_2 + 27x_2^2)\right]$	2	2	–	–
31	Griewank	$f_{\text{Griewank}}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	2	2	–	–
32	Gulf	$f_{\text{Gulf}}(\mathbf{x}) = \sum_{i=1}^{m}\left(e^{-\frac{\lvert y_i - x_2\rvert^{x}-3}{x_1}} - t_i\right)$	3	3	–	–
		where: $t_i = \frac{i}{100}$ and $y_i = 25 + [-50\log(t_i)]^{\frac{2}{3}}$				
33	Hansen	$f_{\text{Hansen}}(\mathbf{x}) = \left[\sum_{i=0}^{4}(i+)\cos(ix_1 + i + 1)\right]\left[\sum_{j=0}^{4}(j+)\cos[(j+2)x_2 + j + 1]\right]$	2	2	10	25

Table 6 continued

<table>
<tr><th>#</th><th>Name</th><th>Expression</th><th colspan="4">Dimension</th></tr>
<tr><th></th><th></th><th></th><th>E1</th><th>E2</th><th>E3</th><th>E4</th></tr>
<tr><td>34</td><td>Hartmann3</td><td>$f_{\text{Hartmann3}}(\mathbf{x}) = -\sum_{i=1}^{4} c_i e^{-\sum_{j=1}^{n} a_{ij}\left(x_j - p_{ij}\right)^2}$
where:
<table>
<tr><th>i</th><th colspan="3">a_{ij}</th><th>c_i</th><th colspan="3">p_{ij}</th></tr>
<tr><td>1</td><td>3.0</td><td>10.0</td><td>30.0</td><td>1.0</td><td>0.689</td><td>0.1170</td><td>0.2673</td></tr>
<tr><td>2</td><td>0.1</td><td>10.0</td><td>35.0</td><td>1.2</td><td>0.4699</td><td>0.4387</td><td>0.7470</td></tr>
<tr><td>3</td><td>3.0</td><td>10.0</td><td>30.0</td><td>3.0</td><td>0.1091</td><td>0.8732</td><td>0.5547</td></tr>
<tr><td>4</td><td>0.1</td><td>10.0</td><td>35.0</td><td>3.2</td><td>0.0381</td><td>0.5743</td><td>0.8828</td></tr>
</table></td><td>3</td><td>3</td><td>–</td><td>–</td></tr>
<tr><td>35</td><td>Hartmann6</td><td>$f_{\text{Hartmann6}}(\mathbf{x}) = -\sum_{i=1}^{4} c_i e^{-\sum_{j=1}^{n} a_{ij}\left(x_j - p_{ij}\right)^2}$
where:
<table>
<tr><th>i</th><th colspan="6">a_{ij}</th><th>c_i</th></tr>
<tr><td>1</td><td>10.0</td><td>3.0</td><td>17.0</td><td>3.50</td><td>1.70</td><td>8.00</td><td>1.0</td></tr>
<tr><td>2</td><td>0.05</td><td>10.0</td><td>17.0</td><td>0.10</td><td>8.00</td><td>14.00</td><td>1.2</td></tr>
<tr><td>3</td><td>3.00</td><td>3.50</td><td>1.70</td><td>10.0</td><td>17.00</td><td>8.00</td><td>3.0</td></tr>
<tr><td>4</td><td>17.00</td><td>8.00</td><td>0.05</td><td>10.00</td><td>0.10</td><td>14.00</td><td>3.2</td></tr>
</table>
<table>
<tr><th>i</th><th colspan="6">p_{ij}</th></tr>
<tr><td>1</td><td>0.1312</td><td>0.1696</td><td>0.5569</td><td>0.0124</td><td>0.8283</td><td>0.5886</td></tr>
<tr><td>2</td><td>0.2329</td><td>0.4135</td><td>0.8307</td><td>0.3736</td><td>0.1004</td><td>0.9991</td></tr>
<tr><td>3</td><td>0.2348</td><td>0.1451</td><td>0.3522</td><td>0.2883</td><td>0.3047</td><td>0.6650</td></tr>
<tr><td>4</td><td>0.4047</td><td>0.8828</td><td>0.8732</td><td>0.5743</td><td>0.1091</td><td>0.0381</td></tr>
</table></td><td>6</td><td>6</td><td>–</td><td>–</td></tr>
<tr><td>36</td><td>HelicalValley</td><td>$f_{\text{HelicalValley}}(\mathbf{x}) = 100[z - 10\Psi(x_1, x_2)]^2 + \left(\sqrt{x_1^2 + x_2^2} - 1\right)^2 + x_3^2$
where: $2\pi\Psi(x, y) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{for} x > 0 \\ \pi + \arctan\left(\frac{y}{x}\right) & \text{for} x < 0 \end{cases}$</td><td>3</td><td>3</td><td>–</td><td>–</td></tr>
<tr><td>37</td><td>HimmelBlau</td><td>$f_{\text{HimmelBlau}}(\mathbf{x}) = \left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2$</td><td>2</td><td>2</td><td>–</td><td>–</td></tr>
</table>

Table 6 continued

#	Name	Expression	Dimension			
			E1	E2	E3	E4
38	HolderTable	$f_{\text{HolderTable}}(\mathbf{x}) = -\left\lvert e^{\left\lvert 1-\frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\rvert} \sin(x_1)\cos(x_2)\right\rvert$	2	2	–	–
39	Holzman	$f_{\text{Holzman}}(x) = \sum_{i=0}^{99}\left[e^{\frac{1}{x_1}(u_i-x_2)^{x_3}} - 0.1(i+1)\right]$	3	3	–	–
		where: $u_i = 25 + (-50\log[0.01(i+1)])^{\frac{2}{3}}$				
40	Hosaki	$f_{\text{Hosaki}}(\mathbf{x}) = \left(1 - 8x_1 + 7x_1^2 - \frac{7}{3}x_1^3 + \frac{1}{4}x_1^4\right)x_2^2 e^{-x_1}$	2	2	–	–
41	Infinity	$f_{\text{Infinity}}(\mathbf{x}) = \sum_{i=1}^{n} x_i^6\left[\sin\left(\frac{1}{x_i}\right) + 2\right]$	2	2	10	25
42	Kowalik	$f_{\text{Kowalik}}(\mathbf{x}) = \sum_{i=0}^{10}\left[a_i - \frac{x_1\left(b_i^2+b_i x_2\right)}{b_i^2+b_i x_3+x_4}\right]^2$	4	4	–	–
		where: $a = (0.1957, 0.1947, 0.1735, 0.1600, 0.0844, 0.0627, 0.0456, 0.0342, 0.0323, 0.0235, 0.0246)$				
		and $b = \left(4, 2, 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{10}, \frac{1}{12}, \frac{1}{14}, \frac{1}{16}\right)$				
43	Langermann	$f_{\text{Langermann}}(\mathbf{x}) = -\sum_{i=1}^{5} c_i e^{-\sum_{j=1}^{D}\frac{(x_j-a_{ij})^2}{\pi}} \cos\left(\pi\sum_{j=1}^{D}\left(x_j - a_{ij}\right)^2\right)$	2	2	–	–
		where: $a = \begin{pmatrix} 3 & 5 \\ 5 & 2 \\ 2 & 1 \\ 1 & 4 \\ 7 & 9 \end{pmatrix}$ and $c = \begin{pmatrix} 0.806 \\ 0.517 \\ 1.5 \\ 0.908 \\ 0.965 \end{pmatrix}$				
44	Leon	$f_{\text{Leon}}(\mathbf{x}) = (1 - x_1)^2 + 100\left(x_2 - x_1^2\right)^2$	2	2	–	–

Table 6 continued

#	Name	Expression	Dimension			
			E1	E2	E3	E4
45	Levy	$f_{\text{Levy}}(\mathbf{x}) = \sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_i + 1)\right] + (y_n - 1)^2$ where: $y_i = 1 + \frac{x_i - 1}{4}$	2	2	10	25
46	Levy13	$f_{\text{levy13}}(\mathbf{x}) = (x_1 - 1)^2\left[\sin^2(3\pi x_2) + 1\right] + (x_2 - 1)^2\left[\sin^2(2\pi x_2) + 1\right] + \sin^2(3\pi x_2)$	2	2	–	–
47	Matyas	$f_{\text{Matyas}}(x) = 0.26\left(x_1^2 + x_1^2\right) - 0.48x_1x_2$	2	2	–	–
48	McCormick	$f_{\text{McCormick}}(x) = -x_1 + 2x_2 + (x_1 - x_2)^2 + \sin(x_1 + x_2) + 1$	2	2	–	–
49	Michalewicz	$f_{\text{Michalewicz}}(\text{x}) = -\sum_{i=1}^{2}\sin(x_i)\sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$ where: $m = 10$	2	2	–	–
50	Mishra01	$f_{\text{Mishra01}}(\text{x}) = (1 + x_n)^{x_n}$ where: $x_n = n - \sum_{i=1}^{n-1} x_i$	2	2	10	25
51	Mishra02	$f_{\text{Mishra01}}(\text{x}) = (1 + x_n)^{x_n}$ where: $x_n = n - \sum_{i=1}^{n-1}\frac{(x_i + x_{i+1})}{2}$	2	2	10	25
52	MultiModal	$f_{\text{MultiModal}}(x) = \left(\sum_{i=1}^{n}\lvert x_i\rvert\right)\left(\prod_{i=1}^{n}\lvert x_i\rvert\right)$	2	2	10	25
53	NeedleEye	$f_{\text{NeedleEye}}(\mathbf{x}) = \begin{cases} 1 & if\lvert x_i\rvert < eye\,\forall i \\ \sum_{i=1}^{n}(100 + \lvert x_i\rvert) & if\lvert x_i\rvert > eye \\ 0 & \text{otherwise} \end{cases}$ where: eye $= 0.0001$	2	2	10	25
54	NewFunction01	$f_{\text{NewFunction01}}(\text{x}) = \left\lvert\cos\left(\sqrt{\left\lvert x_1^2 + x_2\right\rvert}\right)\right\rvert^{0.5} + \frac{(x_1 + x_2)}{100}$	2	2	–	–

Table 6 continued

#	Name	Expression	Dimension			
			E1	E2	E3	E4
55	NewFunction02	$f_{\text{NewFunction02}}(x) = \left\lvert \sin\left(\sqrt{\lvert x_1^2 + x_2 \rvert}\right)\right\rvert^{0.5} + \frac{x_1+x_2}{100}$	2	2	–	–
56	NewFunction03	$f_{\text{NewFunction03}}(x) = 0.01x_1 + 0.1x_2 + \left\{x_1 + \sin^2\left[\left(\cos(x_1) + \cos(x_2)\right)^2\right] + \cos^2\left[\left(\sin(x_1) + \sin(x_2)\right)^2\right]\right\}^2$	2	2	–	–
57	Pathological	$f_{\text{Pathological}}(x) = \sum_{i=1}^{n-1} \frac{\sin^2\left(\sqrt{100x_i^2 + x_i^2}\right) - 0.5}{0.001(x_i - x_{i+1})^4 + 0.50}$	2	2	10	25
58	Paviani	$f_{\text{Paviani}}(\mathbf{x}) = \sum_{i=1}^{10}\left[\log^2(10 - x_i) + \log^2(x_i - 2)\right] - \left(\prod_{i=1}^{10} x_i^{10}\right)^{0.2}$	10	10	10	25
59	PenHolder	$f_{\text{PenHolder}}(x) = -e\left\lvert e\left\lvert -\frac{\sqrt{x_1^2+x_2^2}}{\pi} + 1\right\rvert \cos(x_1)\cos(x_2)\right\rvert^{-1}$	2	2	–	–
60	PermFunction01	$f_{PermFunction01}(x) = \sum_{k=1}^{n}\left\{\sum_{k=1}^{n}(j^k + \beta)\left[\left(\frac{x_i}{j}\right)^k - 1\right]\right\}^2$	2	2	–	–
61	Plateau	$f_{\text{Plateau}}(x) = 30 + \sum_{i=1}^{n} x_i$	2	2	10	25
62	Powell	$f_{\text{Powell}}(x) = (x_3 + 10x_1)^2 + 5(x_2 - x_4)^2 + (x_1 - 2x_2)^4 + 10(x_3 - x_4)^4$	4	4	–	–
63	Power	$f_{\text{Power}}(x) = \sum_{k=1}^{n}\left[\left(\sum_{i=1}^{n} x_i^k\right) - b_k\right]^2$	4	4	–	–
64	Price1	$f_{\text{Price1}}(\mathbf{x}) = (\lvert x_1 \rvert - 5)^2 + (\lvert x_2 \rvert - 5)^2$	2	2	–	–
65	Price2	$f_{\text{Price2}}(x) = 1 + \sin^2(x_1) + \sin^2(x_2) - 0.1e^{\left(-x_1^2 - x_2^2\right)}$	2	2	–	–
66	Price4	$f_{\text{Price4}}(\mathbf{x}) = \left(2x_1^3x_2 - x_2^3\right)^2 + \left(6x_1 - x_2^2 + x_2\right)^2$	2	2	–	–
67	Quintic	$f_{\text{Quintic}}(x) = \sum_{i=1}^{n}\left\lvert x_i^5 - 3x_i^4 + 4x_i^3 + 2x_i^2 - 10x_i - 4\right\rvert$	2	2	10	25
68	Rana	$f_{\text{Rana}}(x) = \sum_{i=1}^{n}\left[x_i \sin\left(\sqrt{\lvert x_1 - x_i + 1\rvert}\right)\cos\left(\sqrt{\lvert x_1 + x_i + 1\rvert}\right) + (x_1 + 1)\sin\left(\sqrt{\lvert x_1 + x_i + 1\rvert}\right)\cos\left(\sqrt{\lvert x_1 - x_i + 1\rvert}\right)\right]$	2	2	–	–

Table 6 continued

#	Name	Expression	Dimension			
			E1	E2	E3	E4
69	Rastrigin	$f_{\text{Rastrigin}}(x) = 10n\sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i)\right]$	2	2	10	25
70	Rosenbrock	$f_{\text{Rosenbrock}}(x) = \sum_{i=1}^{n-1}\left[100\left(x_i^2 - x_{i+1}\right)^2 + (x_i - 1)^2\right]$	2	2	10	25
71	Schaffer	$f_{\text{Schaffer}}(\mathbf{x}) = 0.5 + \frac{\sin^2\left(x_1^2+x_2^2\right)^2 - 0.5}{1+0.001\left(x_1^2+x_2^2\right)^2}$	2	2	–	–
72	Schwefel06	$f_{\text{Schwefel06}}(\mathbf{x}) = \max(\lvert x_1 + 2x_2 - 7\rvert, \lvert 2x_1 + x_2 - 5\rvert)$	2	2	–	–
73	Schwefel22	$f_{\text{Schwefel22}}(\mathbf{x}) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	2	2	10	25
74	Schwefel26	$f_{\text{Schwefel26}}(\mathbf{x}) = -418.9829n - \sum_{i=1}^{n} x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	2	2	10	25
75	Schwefel36	$f_{\text{Schwefel36}}(\mathbf{x}) = -x_1x_2(72 - 2x_1 - 2x_2)$	2	2	–	–
76	Shekel05	$f_{\text{Shekel05}}(x) = \sum_{i=1}^{m}\frac{1}{c_i + \sum_{j=1}^{n}\left(x_j - a_{ij}\right)^2}$ where: $a = \begin{bmatrix} 4.0 & 4.0 & 4.0 & 4.0 \\ 1.0 & 1.0 & 1.0 & 1.0 \\ 8.0 & 8.0 & 8.0 & 8.0 \\ 6.0 & 6.0 & 6.0 & 6.0 \\ 3.0 & 7.0 & 3.0 & 7.0 \end{bmatrix}$ and $c = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.6 \end{bmatrix}$	4	4	–	–
77	Shubert	$f_{\text{Shubert}}(x) = \left(\sum_{i=1}^{5} i\cos[(i+1)x_1 + i]\right)\left(\sum_{i=1}^{5} i\cos[(i+1)x_2 + i]\right)$	2	2	–	–
78	Sodp	$f_{\text{Sodp}}(\mathbf{x}) = \sum_{i=1}^{n}\lvert x_i\rvert^{i+1}$	2	2	10	25
79	Sphere	$f_{\text{Sphere}}(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	2	2	10	25

Table 6 continued

#	Name	Expression	Dimension			
			E1	E2	E3	E4
80	Stochastic	$f_{\text{Stochastic}}(\mathbf{x}) = \sum_{i=1}^{n} rand[0,1]_i \left\| x_i - \frac{1}{i} \right\|$	2	2	10	25
81	StretchedV	$f_{\text{StretchedV}}(\mathbf{x}) = \sum_{i=1}^{n-1} t^{\frac{1}{4}} [\sin(50t^{0.1}) + 1]^2$ where: $t = x_{i+1}^2 + x_i^2$	2	2	–	–
82	StyblinskiTang	$f_{\text{StyblinskiTang}}(\mathbf{x}) = \sum_{i=1}^{n} \left(x_i^4 - 16x_i^2 + 5x_i\right)$	2	2	–	–
83	TestTubeHolder	$f_{\text{TestTubeHolder}}(\mathbf{x}) = -4\left\| e^{\left\|\cos\left(\frac{1}{200}x_1^2 + \frac{1}{200}x_2^2\right)\right\|} \sin(x_1)\cos(x_2) \right\|$	2	2	–	–
84	ThreeHumpCamel	$f_{\text{ThreeHumpCamel}}(\mathbf{x}) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$	2	2	–	–
85	Treccani	$f_{\text{Treccani}}(\mathbf{x}) = x_1^4 + 4x_1^3 + 4x_1^2 + x_2^2$	2	2	–	–
86	Trefethen	$f_{\text{Trefethen}}(\mathbf{x}) = 0.25x_1^2 + 0.25x_2^2 + e^{\sin(50x_1)} - \sin(10x_1 + 10x_2) + \sin(60e^{x_2}) + \sin[70\sin(x_1)] + \sin[\sin(80x_2)]$	2	2	–	–
87	Trid	$f_{\text{Trid}}(x) = \sum_{i=1}^{n}(x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$	6	6	–	–
88	Ursem1	$f_{\text{Ursem1}}(x) = -\sin(2x_1 - 0.5\pi) - 3\cos(x_2) - 0.5x_1$	2	2	–	–
89	Ursem3	$f_{\text{Ursem3}}(\mathbf{x}) = -\sin(2.2\pi x_1 + 0.5\pi)\frac{2-\|x_1\|}{2}\frac{3-\|x_1\|}{2} - \sin(2.2\pi x_2 + 0.5\pi)\frac{2-\|x_2\|}{2}\frac{3-\|x_2\|}{2}$	2	2	–	–
90	Ursem4	$f_{\text{Ursem4}}(\mathbf{x}) = -3\sin(0.5\pi x_1 + 0.5\pi)\frac{2-\sqrt{x_1^2 + x_2^2}}{4}$	2	2	–	–
91	UrsemWaves	$f_{\text{UrsemWaves}}(\mathbf{x}) = -0.9x_1^2 + \left(x_2^2 - 4.5x_2^2\right)x_1x_2 + 4.7\cos\left[2x_1 - x_2^2(2 + x_1)\right]\sin(2.5\pi x_1)$	2	2	–	–
92	Vincent	$f_{\text{Vincent}}(x) = -\sum_{i=1}^{n}\sin(10\log(x))$	2	2	10	25
93	Wavy	$f_{\text{Wavy}}(\mathbf{x}) = 1 - \frac{1}{n}\sum_{i=1}^{n}\cos(kx_i)e^{-\frac{x_i^2}{2}}$ where: $k = 10$	2	2	10	25

Table 6 continued

#	Name	Expression	Dimension			
			E1	E2	E3	E4
94	Wolfe	$f_{\text{Wolfe}}(\mathbf{x}) = \frac{4}{3}\left(x_1^2 + x_2^2 - x_1x_2\right)^{0.75} + x_3$	3	3	–	–
95	XinSheYang02	$f_{\text{XinSheYang02}}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \lvert x_i \rvert}{e^{\sum_{i=1}^{n} \sin\left(x_i^2\right)}}$	2	2	10	25
96	YaoLiu04	$f_{\text{YaoLiu04}}(x) = \max_{i}\{\lvert x_i \rvert, 1 \leq i \leq n\}$	2	2	10	25
97	Zacharov	$f_{\text{Zacharov}}(\mathbf{x}) = \sum_{i=1}^{n} x_i^2 + \left(\frac{1}{2}\sum_{i=1}^{n} ix_i\right)^2 + \left(\frac{1}{2}\sum_{i=1}^{n} ix_i\right)^4$	2	2	10	25
98	ZeroSum	$f_{\text{ZeroSum}}(\mathbf{x}) = \begin{cases} 0 & if \sum_{i=1}^{n} x_i 0 \\ 1 + \left(10000\left\lvert \sum_{i=1}^{n} x_i \right\rvert\right)^{0.5} & \text{otherwise} \end{cases}$	2	2	10	25
99	Zettl	$f_{\text{Zettl}}(\mathbf{x}) = \frac{1}{4}x_1 + \left(x_1^2 - 2x_1 + x_2^2\right)^2$	2	2	–	–
100	Zirilli	$f_{\text{Zirilli}}(\mathbf{x}) = 0.25x_1^4 - 0.5x_1^2 + 0.1x_1 + 0.5x_2^2$	2	2	–	–

3.3.3 Differential Evolution (DE)

DE was initially developed by Price and Storn in 1995 while trying to solve the Chebyshev polynomial fitting problem (Storn and Price 1997). It stems from the genetic annealing algorithm which was also developed by Price (Qing 2009). The DE starts with an initial population randomly generated, then this population evolves using the three evolutionary operations, namely, differential mutation, crossover and selection which are executed in sequence (Qing 2009; Price et al. 2005).

3.3.4 Particle Swarm Optimization (PSO)

PSO is a population based stochastic optimization method developed by Eberhart and Kennedy in 1995. It is inspired from social behavior of bird flocking or fish schooling (Eberhart and Kennedy 1995; Kennedy and Eberhart 1995). The PSO uses a population of particles where each particle has a position and a velocity and keeps track of its coordinates associated with the best solution it has achieved so far, and the overall best solution, and its position, obtained so far by any particle. Then, at each iteration, the velocity and the position of each particle are updated using these two best values (Eberhart and Kennedy 1995; Kennedy and Eberhart 1995).

3.3.5 Harmony Search (HS)

The HS algorithm was originally inspired by the improvisation process of Jazz musicians and it was first developed by Geem et al. (2001). In the HS algorithm, each musician (equivalent to a decision variable) plays (or generates) a note (equivalent to a value) for finding the perfect state of harmony (i.e. the global optimum) all together. This perfectly pleasing harmony is determined by the audio aesthetic standard (Yang 2010b).

3.3.6 Electromagnetism-like mechanism algorithm (EM)

The EM algorithm was proposed by Birbil and Fang (2003). It is based on the attraction–repulsion mechanism of electromagnetism theory to move the sample points toward the optimality (Jolai et al. 2012). This algorithm considers the population as charged particles where each particle is subject to forces from other charges and consequently moves to better solution space (Bouchekara 2013a).

3.3.7 Artificial Bee Colony (ABC)

ABC is an optimization algorithm based on the intelligent foraging behavior of honey bee swarm, it was first developed by Karaboga (2005). In ABC algorithm, the bee colony is composed of three groups of bees: employed bees, onlookers and scouts. The position of a food source is equivalent to a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (equivalent to fitness) of the associated solution (Binitha and Sathya 2012).

For each food source, there is only one employed bee. Therefore, the number of employed bees in the colony is equal to the number of food sources around the hive. Employed bees exploit the food sources and share the information about the food sources with onlooker bees which wait in the hive. Once an onlooker bee selects a food source it becomes an employed bee. An employed bee whose food source has been abandoned becomes a scout and starts searching a new food source in the vicinity of the hive (Karaboga 2005).

3.3.8 Gravitational Search Algorithm (GSA)

GSA, is nature-inspired metaheuristic created by Rashedi et al. (2009). This algorithm is based on the Newtonian gravity where a population of objects attract each other by the gravity force (the performance of each object is related to its masse), and this force causes a global movement of all objects toward the objects with heavier masses (Rashedi et al. 2009).

3.3.9 Firefly Algorithm (FA)

FA is a swarm-based metaheuristic for constrained optimization problems, developed by Yang (2009). It is inspired from the flashing behavior of fireflies. In FA agents or fireflies (where an agent represents a solution of the optimization problem) communicate with each other via bioluminescent glowing which enables them to explore the search space. An agent or a firefly, glows proportionally to its quality. Consequently each brighter firefly attracts its partners (regardless of their sex), which makes the search space being explored more efficiently (Yang 2010a; Binitha and Sathya 2012).

3.3.10 Teaching–Learning-Based Optimization (TLBO)

TLBO is a new metaheuristic introduced recently by Rao et al. (2011). It is based on the principle of sharing knowledge by a teacher with his students in a classroom environment (this constitutes the so called teacher phase) and then sharing knowledge by learners with their classmates (constitutes the so called learner phase) (Črepinšek et al. 2012; Boudjefdjouf et al. 2015; Bouchekara et al. 2014a).

3.3.11 League Championship Algorithm (LCA)

The LCA is an optimization algorithm inspired by sport and it was introduced by Husseinzadeh Kashan (2011). In LCA each individual (team) of the population (league) represents a feasible solution to the problem being solved. These teams compete in an artificial league for several weeks (iterations). Based on the league schedule at each week, teams play in pairs and the outcome (win or loss) is determined based on each team playing strength (fitness). In the recovery period, keeping track of the previous week events, each team devises the required changes in its formation to set up a new formation (a new solution) for the next week contest (Pourali and Aminnayeri 2012; Bouchekara et al. 2014b).

3.3.12 Differential Search Algorithm (DSA)

DSA is an evolutionary algorithm for solving real-valued numerical optimization problems developed by Civicioglu in 2012 (Civicioglu 2012). It is inspired by migration of superorganisms (due to the variation of the capacity and efficiency of food areas) using the concept of Brownian-like random-walk movement. In DSA, a population of artificial-superorganism migrates to the global optimum of the problem. However, during this migration process, the artificial-superorganism tests if some randomly selected positions or locations are suitable to stop over temporarily during the migration. If such suitable location is found, the members of the artificial-superorganism that made such discovery immediately settle at the found location and then continue their migration from this location on (Civicioglu 2012; Bouchekara and Abido 2014).

3.3.13 Black Hole (BH)

The BH algorithm, which was introduced by Hatamlou (2013), is a population-based optimization algorithm inspired from the BH phenomenon. In the BH a population of stars move toward the best candidate, called the BH, in each iteration. If a star crosses the event horizon of the BH it is sucked and it is replaced by a newly and randomly generated star in the search space (Hatamlou 2013; Bouchekara 2013b).

3.4 Test Results

The 4 experiments are run and a summary of the obtained results is displayed in Table 7. This table shows the overall success of each optimization algorithm investigated in this paper, considering for every test function 100 random starting points. Furthermore, the detailed obtained results for each experiment performed in this paper are given in Tables 8, 9, 10, 11, 12, 13, 14 and 15.

From Table 7, it can be noticed that for E1, the MVPA was able to solve, on average, 69.32% of all the test functions for all the 100 random starting points using, on average, 1061.4 functions evaluations. For this experiment the overall success of MVPA is higher than the second-best algorithm i.e. the DE algorithm by 14.36% while the MVPA has an average number of functions evaluations lower than DE by 142.4 evaluation. Recall that the objective of E1 is to test the rapidity of convergence of algorithms since a low number of functions evaluations is allowed. Therefore, the MVPA is fast converging algorithm than the remaining tested algorithms.

For E2, using the same set of test functions than E1 but with higher number of functions evaluations (i.e. 5000 rather than 2000 for E1) the MVPA improved its overall success by 8.27%, however, the number of functions evaluations has increased from 1061.4 to 1273.8.

When the dimensions of problems increase like in E3 and E4, it can be noticed that the MVPA still has an advantage over the remaining algorithms. For E3 the MVPA was able to solve, on average, 56.27% of the tested benchmarks using, on average, 3738.6 functions evaluations which is higher than the second best algorithm, the DE, by 24.3% while the number of functions evaluations of the

Table 7 Summary of the performances of tested optimization algorithms

Optimization algorithms	E1		E2		E3		E4	
	Overall Success (%)	Functions evaluations	Overall success (%)	Functions evaluations	Overall success (%)	Functions evaluations	Overall success (%)	Functions evaluations
MVPA	69.32	1061.4	77.59	1273.8	56.27	3738.6	40.48	7300.0
DE	54.96	1203.8	74.82	1664.0	31.97	3928.2	25.36	7626.7
ABC	40.48	1178.3	62.39	2062.3	27.42	3851.5	29.88	10,240.5
TLBO	20.38	1168.9	46.91	2902.9	31.91	3231.2	33.64	9135.6
PSO	37.67	1406.7	73.66	2370.2	27.70	2705.8	20.06	7322.8
BH	37	871.7	45.54	1466.3	25.36	2242.3	27.24	3790.1
LCA	22.95	1133.3	44.41	2582.9	19.85	6726.7	12.03	15,331.3
EM	12.07	1031.2	15.66	2049.7	20.39	1203.1	23.97	2421.0
DSA	10.08	1413.0	30.61	3379.5	12.67	6931.6	23.94	12,836.3
GA	16.07	263.2	16.07	254.6	15.67	4768.7	6.06	223.3
SA	9.01	894.3	10.71	1843.4	15.12	3223.3	14.03	9175.1
GSA	11.3	409.3	11.62	976.0	13.45	2518.5	10.12	5726.4
HS	8.12	1116.6	13.84	3179.3	12.27	3854.3	13.24	7874.1
FA	12.23	529.9	13.03	1255.0	5.85	3895.2	6.30	5923.8

Table 8 Detailed percentage of success of the tested optimization algorithms for E1

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
1	96	0	0	0	0	0	0	0	65	0	0	0	0	0
2	100	100	0	100	9	100	0	100	0	0	33	0	0	100
3	94	0	0	1	0	0	0	0	62	0	0	0	0	3
4	100	100	100	100	100	100	100	100	100	100	100	94	100	100
5	93	34	0	99	0	0	0	0	19	0	0	0	0	0
6	97	3	0	95	0	0	0	0	48	0	0	0	0	9
7	100	4	0	100	0	1	0	0	89	0	0	0	0	99
8	100	100	0	96	63	89	1	44	64	54	10	95	30	81
9	100	100	0	100	2	11	0	1	54	0	1	1	0	7
10	100	0	0	100	0	0	0	1	71	0	0	0	0	100
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	100	49	0	68	0	0	0	0	5	0	0	0	0	33
13	81	0	0	85	0	0	0	0	25	0	0	0	0	88
14	100	0	0	7	0	0	0	0	7	0	0	0	0	9
15	7	0	0	0	0	0	0	0	0	0	0	0	0	0
16	97	0	0	0	0	0	0	0	1	0	0	0	0	0
17	100	90	0	100	0	3	0	7	81	9	0	2	4	100
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	100	0	0	100	0	0	0	0	20	0	0	0	0	46
21	46	4	0	44	0	0	0	0	44	0	0	0	0	0
22	100	31	0	99	0	0	0	0	55	1	0	0	0	1
23	53	1	0	10	0	0	0	0	15	0	0	0	0	0
24	86	98	100	96	96	94	99	98	84	98	99	100	90	96
25	9	1	0	7	0	0	0	0	0	0	0	0	0	0
26	100	100	0	100	1	15	0	4	37	1	0	1	0	98
27	95	1	0	94	0	0	0	0	28	0	0	0	0	1
28	100	100	98	100	100	100	100	100	100	100	100	100	100	100
29	100	100	100	100	20	79	0	97	86	46	0	5	4	100
30	100	23	0	100	0	0	0	0	67	0	0	0	0	2
31	13	0	0	0	0	0	0	0	2	0	0	0	0	0
32	36	2	0	10	0	0	0	0	0	0	0	0	0	0
33	26	0	0	3	0	0	0	0	13	0	0	0	0	0
34	49	57	0	70	0	0	0	0	3	0	0	0	0	29
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	100	26	0	98	0	0	0	0	59	0	0	0	0	11
38	30	29	0	33	0	0	0	0	5	0	0	0	0	30
39	36	2	0	10	0	0	0	0	0	0	0	0	0	0
40	100	100	95	100	5	42	0	87	69	12	0	1	2	100
41	100	100	100	100	100	100	47	100	100	100	100	100	100	100

Table 8 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	2	0	0	5	0	0	0	0	5	0	0	0	0	0
44	96	62	0	95	1	0	0	0	2	0	0	1	0	0
45	100	99	0	100	0	12	0	63	96	9	0	0	4	100
46	100	15	0	100	0	0	0	0	86	1	0	0	0	100
47	100	100	100	100	100	100	100	98	0	0	100	51	1	55
48	100	100	0	100	10	38	0	2	73	1	0	2	2	91
49	24	27	0	38	0	4	0	28	22	3	0	0	0	38
50	100	100	100	100	100	100	100	100	0	0	100	0	0	100
51	100	100	100	100	100	100	100	100	0	0	100	0	0	100
52	100	99	0	100	59	96	1	100	100	78	0	12	100	100
53	100	96	0	100	23	73	1	76	99	40	7	35	13	100
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0
56	100	100	0	100	0	93	2	77	0	0	2	0	0	96
57	58	5	0	4	4	0	2	1	29	2	0	2	0	0
58	0	0	0	0	0	0	0	0	0	0	0	0	0	0
59	100	100	0	100	10	0	1	8	27	15	0	1	34	100
60	100	92	69	100	4	0	0	0	26	1	0	1	0	0
61	100	100	100	100	100	100	76	100	100	100	100	100	100	100
62	28	1	0	0	0	0	0	0	1	0	0	0	0	0
63	0	0	0	0	0	0	0	0	0	0	0	0	0	0
64	99	0	0	23	0	0	0	0	87	0	0	0	0	45
65	0	0	0	0	0	0	0	0	0	0	0	0	0	0
66	100	28	0	99	0	0	0	2	27	2	0	0	3	23
67	76	0	90	0	0	0	0	0	59	0	0	0	0	0
68	0	0	0	0	0	0	0	0	0	0	0	0	0	0
69	65	0	0	66	0	0	0	0	56	0	0	0	0	36
70	77	2	80	60	0	0	0	0	1	0	0	1	0	0
71	12	0	0	0	0	0	0	0	1	0	0	0	0	0
72	58	0	0	0	0	0	0	0	5	0	0	0	0	0
73	97	0	0	0	0	0	0	0	70	0	0	0	0	0
74	73	0	0	62	0	0	0	0	20	0	0	0	0	18
75	98	0	0	19	0	0	0	0	1	0	0	0	0	0
76	23	0	0	0	0	0	0	0	0	0	0	0	0	0
77	5	0	0	5	0	0	0	0	5	0	0	0	0	0
78	100	100	0	100	49	100	0	100	100	89	0	91	19	100
79	100	99	0	100	0	30	0	84	100	9	0	4	2	100
80	14	0	1	0	0	0	0	0	5	0	0	0	0	0
81	100	100	0	99	100	100	100	100	100	100	100	100	100	100
82	100	30	0	100	0	0	0	0	45	0	0	0	0	99
83	49	0	0	51	0	0	0	0	17	1	0	0	0	26

Table 8 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
84	100	30	0	99	0	0	0	4	100	1	0	0	0	85
85	100	94	0	100	3	4	0	26	95	4	0	0	0	100
86	12	0	0	0	0	0	0	0	1	0	0	0	0	0
87	0	0	0	0	0	0	0	0	0	0	0	0	0	0
88	100	100	0	100	2	27	0	83	57	23	0	0	1	100
89	100	0	0	5	0	0	0	0	31	0	0	0	0	21
90	100	0	0	23	0	0	0	0	39	0	0	0	0	6
91	39	97	74	44	37	100	100	100	0	0	72	0	0	100
92	99	59	0	78	5	0	0	4	66	2	0	0	2	94
93	92	10	0	96	0	0	0	0	78	0	0	0	0	81
94	100	100	100	100	0	100	100	100	0	0	100	0	0	100
95	95	0	0	0	0	0	0	0	46	0	0	0	0	0
96	100	0	0	0	0	0	0	0	59	0	0	0	0	0
97	100	94	100	100	2	14	0	61	96	5	0	0	1	100
98	100	96	100	100	0	100	100	100	0	0	98	0	0	75
99	100	93	0	100	1	7	0	0	91	0	1	1	0	16
100	97	84	0	100	1	6	0	39	98	1	0	0	0	100

MVPA is lower than the one of DE by 189.6 evaluations. Finally for E4, the MVPA was able to solve, on average, 40.48% of the tested benchmarks using, on average, 7300.0 functions evaluations which is higher than the second best algorithm, the TLBO, by 6.84% while the number of functions evaluations of the MVPA is lower than the one of TLBO by 1835.6 evaluations.

Therefore, the obtained results show clearly that MVPA is one of the best optimization algorithms as far as the current experiments are considered.

The performances of tested algorithms are also depicted in Figs. 4, 5, 6 and 7 for E1, E2 E3 and E4, respectively. In these figures, the overall successes of algorithms are drawn as bars while the numbers of functions evaluations are drawn as curves.

4 Conclusion

In this paper, a new optimization algorithm called the Most Valuable Player Algorithm (MVPA) is developed. This algorithm is inspired from sport. The performances of the MVPA have been assessed using 100 benchmarks and via 4 experiments. Then, the obtained results have been compared with the ones obtained using 13 well-known optimization algorithms. As illustrated in this paper, for all the investigated experiments, the MVPA is the algorithm that has achieved the best results (i.e. the best overall success percentage) using lower computational efforts (i.e. lower number of functions evaluations).

Table 9 Detailed number of function evaluations of the tested optimization algorithms for E1

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
1	1834	–	–	–	–	–	–	–	1181	–	–	–	–	–
2	389	597	–	402	910	914	–	1036	–	–	1143	–	–	724
3	1529	–	–	1801	–	–	–	–	1026	–	–	–	–	1926
4	203	152	100	101	112	201	103	102	102	106	61	195	152	80
5	1106	1922	–	1576	–	–	–	–	1249	–	–	–	–	–
6	1158	2051	–	1701	–	–	–	–	963	–	–	–	–	1865
7	1089	1901	–	1655	–	2001	–	–	757	–	–	–	–	1468
8	496	773	–	645	1053	1350	101	1183	730	1282	1196	920	1293	1228
9	831	1753	–	1206	1781	1878	–	1951	705	–	51	1457	–	1676
10	1067	–	–	1542	–	–	–	1351	1019	–	–	–	–	1278
11	–	–	–	–	–	–	–	–	–	–	–	–	–	–
12	992	1916	–	1152	–	–	–	–	1567	–	–	–	–	1782
13	1246	–	–	1720	–	–	–	–	1159	–	–	–	–	1720
14	1527	–	–	1922	–	–	–	–	1472	–	–	–	–	1870
15	1985	–	–	–	–	–	–	–	–	–	–	–	–	–
16	1806	–	–	–	–	–	–	–	1233	–	–	–	–	–
17	848	1791	–	1386	–	1501	–	1708	576	1590	–	1786	1547	1233
18	–	–	–	–	–	–	–	–	–	–	–	–	–	–
19	–	–	–	–	–	–	–	–	–	–	–	–	–	–
20	1095	–	–	1631	–	–	–	–	814	–	–	–	–	1842
21	1208	1926	–	1840	–	–	–	–	1139	–	–	–	–	–
22	1015	1938	–	1600	–	–	–	–	1165	1801	–	–	–	1326
23	1364	1951	–	1861	–	–	–	–	887	–	–	–	–	–
24	201	151	100	101	114	201	101	101	101	101	51	3	152	76
25	868	1851	–	944	–	–	–	–	–	–	–	–	–	–

Table 9 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
26	816	1668	–	1116	1513	1901	–	1826	904	2001	–	1206	–	1545
27	1210	1951	–	1712	–	–	–	–	1254	–	–	–	–	1826
28	250	335	387	216	265	423	610	408	369	636	352	486	386	300
29	631	1283	190	875	1019	1760	–	1482	482	1701	–	987	1483	803
30	1050	1968	–	1619	–	–	–	–	958	–	–	–	–	1751
31	1736	–	–	–	–	–	–	–	1231	–	–	–	–	–
32	1675	1901	–	1866	–	–	–	–	–	–	–	–	–	–
33	1375	–	–	1984	–	–	–	–	1139	–	–	–	–	–
34	933	1816	–	1509	–	–	–	–	1230	–	–	–	–	1709
35	–	–	–	–	–	–	–	–	–	–	–	–	–	–
36	–	–	–	–	–	–	–	–	–	–	–	–	–	–
37	1092	1928	–	1608	–	–	–	–	786	–	–	–	–	1717
38	915	1854	–	1039	–	–	–	–	1436	–	–	–	–	1743
39	1675	1901	–	1866	–	–	–	–	–	–	–	–	–	–
40	730	1560	171	1047	1275	1869	–	1707	604	1764	–	701	1164	945
41	201	189	111	130	122	257	101	164	106	194	115	314	188	127
42	–	–	–	–	–	–	–	–	–	–	–	–	–	–
43	1483	–	–	1451	–	–	–	–	726	–	–	–	–	–
44	1449	1845	–	1234	1323	–	–	–	552	–	–	844	–	–
45	800	1748	–	1160	–	1889	–	1741	547	1795	–	–	1742	961
46	1042	1931	–	1541	–	–	–	–	739	1651	–	–	–	1540
47	206	172	240	101	854	201	201	287	–	–	223	1419	1909	290
48	737	1496	–	1001	1121	1796	–	1976	872	1851	–	489	1577	1572
49	775	1684	–	1102	–	1851	–	1728	748	1834	–	–	–	952
50	204	151	100	101	303	201	201	101	–	–	103	–	–	79

Table 9 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
51	207	159	140	101	1393	201	201	183	–	–	208	–	–	180
52	580	1253	–	814	1102	1482	601	1152	354	1479	–	1057	1220	558
53	706	1483	–	996	1068	1670	101	1481	461	1450	1094	1003	1370	908
54	–	–	–	–	–	–	–	–	–	–	–	–	–	–
55	–	–	–	–	–	–	–	–	–	–	–	–	–	–
56	493	1060	–	552	–	1403	901	1443	–	–	1626	–	–	1280
57	1336	1371	–	739	976	–	1101	1251	659	1801	–	1662	–	–
58	–	–	–	–	–	–	–	–	–	–	–	–	–	–
59	788	1401	–	1160	1461	–	1901	1701	829	1611	–	543	1500	1237
60	1055	1785	162	1394	1690	–	–	–	1080	1951	–	1057	–	–
61	202	187	159	137	130	243	305	143	104	221	90	613	171	119
62	1750	2051	–	–	–	–	–	–	701	–	–	–	–	–
63	–	–	–	–	–	–	–	–	–	–	–	–	–	–
64	1349	–	–	1892	–	–	–	–	858	–	–	–	–	1787
65	–	–	–	–	–	–	–	–	–	–	–	–	–	–
66	1035	1872	–	1515	–	–	–	1851	678	1901	–	–	1558	1574
67	1882	–	182	–	–	–	–	–	1264	–	–	–	–	–
68	–	–	–	–	–	–	–	–	–	–	–	–	–	–
69	1362	–	–	1862	–	–	–	–	1010	–	–	–	–	1783
70	1709	2051	180	1794	–	–	–	–	1234	–	–	1353	–	–
71	1600	–	–	–	–	–	–	–	701	–	–	–	–	–
72	1929	–	–	–	–	–	–	–	1107	–	–	–	–	–
73	1875	–	–	–	–	–	–	–	1220	–	–	–	–	–
74	1312	–	–	1816	–	–	–	–	1085	–	–	–	–	1790
75	1435	–	–	1859	–	–	–	–	1511	–	–	–	–	–

Table 9 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
76	1939	–	–	–	–	–	–	–	–	–	–	–	–	–
77	1455	–	–	1841	–	–	–	–	1134	–	–	–	–	–
78	492	995	–	675	840	1265	–	942	307	1371	–	843	1290	537
79	778	1700	–	1168	–	1811	–	1706	539	1729	–	1248	1539	962
80	1841	–	1717	–	–	–	–	–	1303	–	–	–	–	–
81	218	208	–	144	137	234	235	139	120	158	123	77	164	113
82	990	1921	–	1470	–	–	–	–	797	–	–	–	–	1479
83	1353	–	–	1824	–	–	–	–	815	1501	–	–	–	1832
84	970	1931	–	1546	–	–	–	1526	666	1751	–	–	–	1669
85	874	1828	–	1299	1764	1964	–	1741	578	1526	–	–	–	1129
86	1577	–	–	–	–	–	–	–	1616	–	–	–	–	–
87	–	–	–	–	–	–	–	–	–	–	–	–	–	–
88	762	1582	–	1031	1157	1857	–	1712	585	1671	–	–	759	918
89	1511	–	–	1911	–	–	–	–	1314	–	–	–	–	1900
90	1402	–	–	1940	–	–	–	–	1256	–	–	–	–	1934
91	236	471	187	103	1870	258	201	227	–	–	509	–	–	235
92	1055	1814	–	1623	1379	–	–	1514	662	1776	–	–	1443	1532
93	1173	1921	–	1626	–	–	–	–	976	–	–	–	–	1704
94	203	225	325	103	–	201	201	457	–	–	386	–	–	453
95	1639	–	–	–	–	–	–	–	1137	–	–	–	–	–
96	1611	–	–	–	–	–	–	–	1018	–	–	–	–	–
97	850	1797	174	1208	775	1872	–	1803	593	1801	–	–	1957	1034
98	212	210	113	105	–	213	201	258	–	–	508	–	–	273
99	886	1839	–	1300	1603	1958	–	–	949	–	1701	1203	–	1542
100	867	1824	–	1275	1827	1751	–	1852	761	1801	–	–	–	1112

Table 10 Detailed percentage of success of the tested optimization algorithms for E2

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
1	100	100	0	100	0	0	0	0	73	0	0	0	0	100
2	100	100	0	100	15	100	4	100	0	2	56	0	0	100
3	100	98	0	100	0	0	0	1	72	0	0	0	0	100
4	100	100	100	100	100	100	100	100	100	100	100	96	100	100
5	95	100	0	97	0	64	0	1	65	1	0	0	0	1
6	94	98	0	99	1	20	0	1	44	5	0	0	1	100
7	100	100	0	100	0	100	0	100	98	65	0	0	2	100
8	100	100	0	99	75	100	0	94	94	99	42	100	72	99
9	100	100	0	100	2	100	0	36	64	26	0	1	4	100
10	100	100	0	100	1	100	0	100	95	89	0	0	2	100
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	100	100	0	72	2	14	0	22	10	8	0	0	3	100
13	82	71	0	84	0	67	0	100	55	32	0	0	2	100
14	100	99	0	100	0	86	0	100	32	2	0	0	0	100
15	100	38	0	59	0	0	0	0	0	0	0	0	0	0
16	100	56	0	100	0	58	0	0	39	0	0	0	0	100
17	100	100	0	100	4	74	0	100	90	99	1	5	48	100
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	100	86	0	100	0	100	0	100	20	20	0	0	0	100
21	65	91	0	86	1	0	0	1	61	6	0	0	0	45
22	100	100	0	100	1	77	0	0	96	2	0	0	0	7
23	71	83	0	85	0	0	0	1	23	0	0	0	0	29
24	88	96	100	98	98	97	95	98	90	97	95	100	92	98
25	12	13	0	9	0	4	0	1	0	0	0	0	0	1

Table 10 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
26	100	100	0	100	5	100	0	100	59	82	0	2	26	100
27	94	100	0	96	0	8	0	1	63	1	0	0	0	83
28	100	100	98	100	100	100	100	100	100	100	100	100	100	100
29	100	100	100	100	30	100	0	100	86	100	1	29	46	100
30	100	100	0	100	0	99	0	0	74	3	0	1	0	72
31	22	23	0	33	0	0	0	0	6	0	0	0	1	48
32	99	37	0	75	0	0	0	0	0	0	0	0	0	0
33	32	48	0	47	0	0	0	0	14	0	0	0	0	25
34	59	72	0	81	0	69	0	62	3	32	0	0	1	73
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	71	14	0	0	0	0	0	0	0	0	0	0	0	0
37	100	100	0	100	2	23	0	5	83	16	0	0	1	100
38	31	55	0	34	2	15	0	20	6	25	0	0	2	61
39	99	37	0	75	0	0	0	0	0	0	0	0	0	0
40	100	100	96	100	8	100	0	100	82	100	0	10	26	100
41	100	100	100	100	100	100	58	100	100	100	100	100	100	100
42	18	0	0	39	0	0	0	0	0	0	0	0	0	0
43	7	4	0	2	0	0	0	0	3	0	0	0	0	0
44	100	100	0	93	1	0	0	0	9	2	0	1	0	0
45	100	100	0	100	2	100	0	100	100	100	0	1	40	100
46	100	100	0	100	0	99	0	100	97	67	0	0	1	100
47	100	100	100	100	100	100	100	98	1	4	100	91	3	67
48	100	100	0	100	14	100	0	83	81	86	0	10	25	100
49	36	40	0	38	0	37	0	45	25	40	0	0	2	32
50	100	100	100	100	100	100	100	100	0	12	100	0	0	100

Table 10 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
51	100	100	100	100	100	100	100	100	0	0	100	0	0	100
52	100	100	0	100	86	100	0	100	100	100	7	29	100	100
53	100	100	0	100	55	100	5	100	99	100	10	72	30	100
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0
56	100	100	0	100	0	100	5	100	0	0	12	0	0	100
57	91	53	0	4	5	3	3	1	37	20	0	8	25	9
58	95	0	0	0	0	0	0	0	0	0	0	0	0	0
59	100	100	0	100	35	85	0	100	50	100	0	0	98	100
60	100	100	70	100	5	27	0	0	52	6	0	3	0	0
61	100	100	100	100	100	100	92	100	100	100	100	100	100	100
62	100	52	0	100	0	13	0	0	1	0	0	0	0	0
63	6	6	0	0	0	0	0	0	0	0	0	0	0	0
64	100	100	0	100	0	7	0	48	93	18	0	0	3	100
65	0	0	0	0	0	0	0	0	0	0	0	0	0	0
66	100	100	0	100	6	98	0	25	44	31	0	0	26	59
67	100	76	88	100	0	0	0	0	80	0	0	0	0	94
68	0	0	0	0	0	0	0	0	0	0	0	0	0	0
69	64	84	0	100	0	0	0	46	88	8	0	0	0	100
70	100	100	82	100	0	0	0	0	4	0	0	0	0	0
71	9	15	0	29	0	0	0	0	7	0	0	0	0	1
72	100	94	0	100	0	0	0	0	8	0	0	0	0	0
73	100	96	0	100	0	1	0	22	74	0	0	0	0	100
74	68	60	0	73	0	1	0	1	27	5	0	0	28	100
75	100	54	0	53	0	1	0	0	6	0	0	0	0	12

Table 10 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
76	43	35	0	48	0	0	0	0	0	0	0	0	0	17
77	9	16	0	19	0	0	0	0	7	0	0	0	0	18
78	100	100	0	100	81	100	0	100	100	100	7	99	71	100
79	100	100	0	100	9	100	0	100	100	100	0	5	19	100
80	35	97	2	22	0	1	0	0	25	0	0	0	0	0
81	100	100	0	98	100	100	100	100	100	100	100	100	100	100
82	100	100	0	100	1	100	0	100	44	82	0	0	1	100
83	48	47	0	71	0	0	0	7	22	27	0	0	2	100
84	100	100	0	100	1	92	0	100	100	63	0	0	2	100
85	100	100	0	100	8	99	0	100	100	99	0	1	9	100
86	14	22	0	29	0	0	0	0	14	0	0	0	0	2
87	97	19	0	0	0	0	0	0	0	0	0	0	0	0
88	100	100	0	100	10	100	0	100	79	100	0	5	12	100
89	100	100	0	100	0	20	0	72	36	1	0	0	0	100
90	100	100	0	100	0	84	0	1	44	1	0	0	0	100
91	43	94	71	41	100	100	100	100	0	0	72	0	0	100
92	100	98	0	100	4	5	0	86	67	97	0	0	17	100
93	79	93	0	99	1	3	0	94	96	40	0	0	1	100
94	100	100	100	100	91	100	100	100	0	0	100	0	0	100
95	86	99	0	96	0	0	0	0	73	0	0	0	0	1
96	100	100	0	100	0	40	0	0	71	0	0	0	0	100
97	100	100	100	100	1	100	0	100	96	99	0	0	11	100
98	100	97	100	100	0	100	100	99	97	0	100	0	0	85
99	100	100	0	100	1	100	0	69	100	43	0	1	15	100
100	97	100	0	99	2	100	0	100	100	98	0	1	14	100

Table 11 Detailed number of function evaluations of the tested optimization algorithms for E2

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
1	1832	4512	–	2853	–	–	–	–	1602	–	–	–	–	3033
2	389	597	–	413	1916	975	2526	1063	–	4351	2254	–	–	732
3	1591	3706	–	2603	–	–	–	3651	1392	–	–	–	–	2809
4	203	152	100	101	112	201	103	102	102	106	61	184	152	80
5	1113	2232	–	1568	–	4263	–	3251	2614	5001	–	–	–	3527
6	1147	2579	–	1719	1731	4536	–	2951	1056	4321	–	–	3646	2741
7	1089	2453	–	1660	–	3551	–	2878	891	4236	–	–	3782	1486
8	496	773	–	672	1526	1419	–	2215	1175	1998	2599	1056	1852	1545
9	831	1753	–	1216	2599	2636	–	4394	942	3986	–	3604	3133	2884
10	1067	2697	–	1569	2745	3323	–	2588	1305	3953	–	–	3870	1306
11	–	–	–	–	–	–	–	–	–	–	–	–	–	–
12	992	2169	–	1139	3836	4690	–	4237	2110	4620	–	–	4517	2134
13	1247	3371	–	1725	–	4652	–	3480	2312	4393	–	–	3526	1761
14	1527	4309	–	2343	–	4647	–	4068	2510	4626	–	–	–	2331
15	2518	3927	–	4718	–	–	–	–	–	–	–	–	–	–
16	1804	3651	–	3572	–	4786	–	–	3939	–	–	–	–	2972
17	848	1799	–	1386	2768	3939	–	2758	799	3220	3401	2325	3496	1237
18	–	–	–	–	–	–	–	–	–	–	–	–	–	–
19	–	–	–	–	–	–	–	–	–	–	–	–	–	–
20	1095	3849	–	1613	–	3540	–	3591	1523	4626	–	–	–	2150
21	1473	2841	–	2049	4057	–	–	4401	2215	3901	–	–	–	3995
22	1015	2236	–	1618	1275	4103	–	–	2085	3826	–	–	–	3691
23	1936	3111	–	2407	–	–	–	4751	2224	–	–	–	–	4192
24	201	151	100	101	114	201	101	101	101	101	51	3	152	76
25	868	2620	–	1073	–	3801	–	3701	–	–	–	–	–	4781

Table 11 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
26	816	1668	–	1113	3316	2366	–	2767	1548	4004	–	1579	3526	1614
27	1197	2927	–	1729	–	4757	–	4001	2433	4301	–	–	–	3657
28	250	335	325	216	265	423	610	408	369	636	352	486	386	300
29	631	1283	190	875	2343	1757	–	1503	786	2112	1751	2146	3293	801
30	1050	2287	–	1573	–	4037	–	–	1017	4018	–	3030	–	3716
31	1805	4108	–	3069	–	–	–	–	1881	–	–	–	4551	4061
32	2317	3037	–	2249	–	–	–	–	–	–	–	–	–	–
33	1406	3451	–	2620	–	–	–	–	1473	–	–	–	–	3998
34	921	1894	–	1452	–	3160	–	3936	2401	4453	–	–	4604	2076
35	–	–	–	–	–	–	–	–	–	–	–	–	–	–
36	3641	3665	–	–	–	–	–	–	–	–	–	–	–	–
37	1092	2321	–	1654	3372	4466	–	4141	1205	4420	–	–	4750	2982
38	917	2066	–	1001	2496	4371	–	4439	2526	4353	–	–	3754	1956
39	2317	3037	–	2249	–	–	–	–	–	–	–	–	–	–
40	730	1560	224	1047	2509	2195	–	1753	948	2605	–	2369	3598	902
41	201	189	111	130	122	257	172	163	108	198	113	318	175	127
42	3835	–	–	4334	–	–	–	–	–	–	–	–	–	–
43	1352	2476	–	1226	–	–	–	–	810	–	–	–	–	–
44	1468	1928	–	1247	891	–	–	–	2432	3851	–	2651	–	–
45	800	1751	–	1165	3343	2463	–	1926	578	3085	–	188	3562	994
46	1042	2344	–	1531	–	3892	–	3032	1009	4221	–	–	4745	1534
47	206	172	240	101	854	201	201	262	3055	3751	219	2087	3410	449
48	737	1496	–	963	1638	2172	–	3584	1471	3767	–	2828	3367	1668
49	752	1689	–	1133	–	2625	–	1970	1041	2937	–	–	2576	992
50	204	151	100	101	303	201	201	101	–	4343	102	–	–	80

Table 11 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
51	207	159	140	101	1393	201	201	183	–	–	204	–	–	179
52	580	1277	–	824	1790	1469	–	1155	331	1646	3137	1608	1248	569
53	706	1474	–	973	2605	1760	2621	1733	487	2311	2721	1989	2589	923
54	–	–	–	–	–	–	–	–	–	–	–	–	–	–
55	–	–	–	–	–	–	–	–	–	–	–	–	–	–
56	493	1060	–	537	–	1488	4021	1583	–	–	2880	–	–	1336
57	1701	3542	–	776	1903	2901	3434	1101	1167	4059	–	3286	3967	3018
58	4321	–	–	–	–	–	–	–	–	–	–	–	–	–
59	788	1401	–	1167	2356	3727	–	2850	1513	2591	–	–	2712	1194
60	1055	1819	146	1405	2387	4251	–	–	2587	3676	–	2646	–	–
61	202	187	159	137	130	243	672	162	107	211	91	621	166	116
62	2593	3470	–	3760	–	4759	–	–	2696	–	–	–	–	–
63	3783	4151	–	–	–	–	–	–	–	–	–	–	–	–
64	1365	3095	–	2166	–	4958	–	4223	1078	4426	–	–	4761	2132
65	–	–	–	–	–	–	–	–	–	–	–	–	–	–
66	1035	2327	–	1494	3102	3300	–	3613	1684	4199	–	–	3042	2527
67	1993	4748	317	3186	–	–	–	–	1753	–	–	–	–	3975
68	–	–	–	–	–	–	–	–	–	–	–	–	–	–
69	1334	3233	–	1941	–	–	–	4444	2049	4270	–	–	–	2079
70	1840	2648	218	1956	–	–	–	–	2489	–	–	–	–	–
71	1689	4104	–	3789	–	–	–	–	1556	–	–	–	–	3777
72	2077	4698	–	3221	–	–	–	–	2000	–	–	–	–	–
73	1879	4648	–	2862	–	4701	–	4810	1439	–	–	–	–	2754
74	1317	3006	–	1838	–	5001	–	4751	2103	4741	–	–	4263	2614
75	1479	3010	–	2209	–	4851	–	–	2785	–	–	–	–	4189

Table 11 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
76	1974	3317	–	4372	–	–	–	–	–	–	–	–	–	4550
77	1400	3145	–	2622	–	–	–	–	1082	–	–	–	–	4071
78	492	995	–	697	1518	1290	–	993	299	1543	3372	1066	2773	538
79	778	1704	–	1150	2635	2262	–	1867	492	2877	–	3569	4021	962
80	2827	3900	1399	4144	–	4851	–	–	2112	–	–	–	–	–
81	218	208	–	136	130	233	150	140	128	149	118	70	166	113
82	990	2181	–	1438	1573	3612	–	2990	1056	4010	–	–	4362	1453
83	1272	2966	–	1918	–	–	–	4115	1253	4207	–	–	4140	2462
84	970	2199	–	1520	1757	3734	–	3554	708	4077	–	–	4110	1750
85	874	1818	–	1303	2385	3477	–	2303	670	3062	–	3724	4331	1109
86	1929	3978	–	3725	–	–	–	–	2686	–	–	–	–	4778
87	3582	3956	–	–	–	–	–	–	–	–	–	–	–	–
88	762	1582	–	1059	2535	2223	–	1742	757	2670	–	1649	3700	955
89	1511	3604	–	2296	–	4821	–	4627	2094	3901	–	–	–	2296
90	1402	3309	–	2161	–	4494	–	4951	1465	4851	–	–	–	2657
91	235	433	193	101	2249	278	201	228	–	–	723	–	–	252
92	1072	2109	–	1713	1594	4461	–	3554	927	3381	–	–	3547	1537
93	1210	2786	–	1644	2043	4751	–	4132	1522	4299	–	–	3407	1783
94	203	225	325	103	3616	204	201	428	–	–	373	–	–	482
95	1627	3958	–	2682	–	–	–	–	1713	–	–	–	–	4776
96	1611	3829	–	2566	–	4862	–	–	1354	–	–	–	–	3830
97	850	1810	194	1208	1981	2351	–	1980	598	3333	–	–	3636	1046
98	212	222	104	105	–	209	201	259	2823	–	580	–	–	560
99	886	1831	–	1274	4953	2603	–	3976	1098	3929	–	2376	3998	2992
100	901	1876	–	1264	3474	2663	–	2120	790	3278	–	2314	3706	1108

Table 12 Detailed percentage of success of the tested optimization algorithms for E3

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
1	33	0	0	0	0	0	0	0	0	0	0	0	0	0
3	99	0	0	0	0	0	0	0	0	0	0	0	0	0
4	100	100	31	100	100	100	60	7	100	3	0	77	11	100
7	100	100	0	100	0	100	0	0	100	1	0	0	5	7
14	100	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	10	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	92	94	100	89	92	89	97	94	68	95	95	100	92	98
33	23	41	0	40	0	0	0	0	30	0	0	0	0	0
41	100	100	0	99	100	100	0	97	100	63	0	100	0	100
45	21	0	0	0	0	0	0	0	0	0	0	0	0	0
50	100	100	100	100	0	100	99	100	0	0	5	0	0	100
51	100	92	96	100	0	100	92	100	0	0	4	0	0	100
52	100	100	0	97	100	100	2	100	100	100	0	3	100	100
53	100	100	0	100	100	100	94	55	100	80	89	100	97	100
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0
58	100	0	0	0	0	0	0	0	0	0	0	0	0	0
61	87	20	69	100	100	100	0	92	41	54	0	100	100	100
67	98	0	0	0	0	0	0	0	0	0	0	0	0	0
69	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	0	0	4	0	0	0	0	0	0	0	0	0	0	0
73	65	0	0	0	0	0	0	0	0	0	0	0	0	0
74	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78	100	63	0	58	81	100	0	10	100	22	0	19	0	100

Table 12 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
79	100	0	0	0	0	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0	0	0	0	0
92	99	0	0	0	0	0	0	0	0	0	0	0	0	0
93	0	0	0	0	0	0	0	0	0	0	0	0	0	0
95	0	0	0	0	0	0	0	0	0	0	0	0	0	0
96	11	0	0	0	0	0	0	0	0	0	0	0	0	0
97	100	0	17	0	0	0	0	0	0	0	0	0	0	0
98	19	4	100	72	0	64	0	0	98	0	0	0	0	0

Table 13 Detailed number of function evaluations of the tested optimization algorithms for E3

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
1	7184	–	–	–	–	–	–	–	–	–	–	–	–	–
3	6780	–	–	–	–	–	–	–	–	–	–	–	–	–
4	964	2109	7513	635	1069	3587	4283	9058	2616	9534	–	1721	8037	5578
7	1089	4631	–	3315	–	6884	–	–	1143	8901	–	–	8027	9190
14	7870	–	–	–	–	–	–	–	–	–	–	–	–	–
21	–	–	–	–	–	–	–	–	–	–	–	–	–	–
22	7638	–	–	–	–	–	–	–	–	–	–	–	–	–
23	–	–	–	–	–	–	–	–	–	–	–	–	–	–
24	201	301	200	201	212	401	101	201	201	201	101	3	152	151
33	2114	5916	–	5286	–	–	–	–	1876	–	–	–	–	–
41	1009	2183	–	7691	535	3266	–	7702	2250	8585	–	1615	–	3436
45	3859	–	–	–	–	–	–	–	–	–	–	–	–	–
50	467	1483	3477	410	–	1937	1495	6265	–	–	5881	–	–	2230
51	542	1860	4720	471	–	2522	2343	6869	–	–	5901	–	–	2389
52	960	3083	–	6653	562	3266	3451	6251	778	5362	–	6844	1118	1140
53	1038	2213	–	3975	1725	2996	3438	6945	1052	5835	3698	1519	3572	5215
57	–	–	–	–	–	–	–	–	–	–	–	–	–	–
58	4292	–	–	–	–	–	–	–	–	–	–	–	–	–
61	1134	1981	5578	5999	828	2810	–	8218	1855	8097	–	3927	2220	3760
67	8228	–	–	–	–	–	–	–	–	–	–	–	–	–
69	–	–	–	–	–	–	–	–	–	–	–	–	–	–
70	–	–	8566	–	–	–	–	–	–	–	–	–	–	–
73	9018	–	–	–	–	–	–	–	–	–	–	–	–	–
74	–	–	–	–	–	–	–	–	–	–	–	–	–	–
78	1269	5060	–	9177	3491	4436	–	9031	3594	8937	–	6934	–	5426

Table 13 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
79	3594	–	–	–	–	–	–	–	–	–	–	–	–	–
80	–	–	–	–	–	–	–	–	–	–	–	–	–	–
92	4084	–	–	–	–	–	–	–	–	–	–	–	–	–
93	–	–	–	–	–	–	–	–	–	–	–	–	–	–
95	–	–	–	–	–	–	–	–	–	–	–	–	–	–
96	9299	–	–	–	–	–	–	–	–	–	–	–	–	–
97	3783	–	7864	–	–	–	–	–	–	–	–	–	–	–
98	3312	1651	233	3326	–	3439	–	–	7060	–	–	–	–	–

Table 14 Detailed percentage of success of the tested optimization algorithms for E4

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	18	0	0	0	0	0	0	0	0	0	0	0	0	0
4	100	0	0	99	100	100	0	0	100	44	0	54	0	100
7	100	100	0	100	2	100	0	0	100	73	0	0	37	1
14	2	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	82	88	100	96	89	83	97	95	76	90	97	99	94	96
33	20	39	0	41	0	2	0	0	30	0	0	0	2	0
41	90	0	0	0	100	100	0	1	100	98	0	100	0	100
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	100	3	0	100	0	19	0	5	0	0	0	0	0	92
51	100	1	0	100	0	6	1	2	0	0	0	0	0	97
52	100	100	0	95	100	100	100	100	100	100	0	1	100	100
53	100	100	0	100	100	100	100	86	100	100	100	100	100	100
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	0	0	0	0	0	0
61	3	0	0	5	100	100	0	0	0	85	0	42	4	100
67	0	0	0	0	0	0	0	0	0	0	0	0	0	0
69	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0	0	0	0	0	0
73	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78	100	31	0	1	100	100	0	8	100	100	0	3	0	100

Table 14 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
79	100	0	0	0	0	100	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0	0	0	0	0
92	98	0	0	0	0	0	0	0	0	0	0	0	0	0
93	0	0	0	0	0	0	0	0	0	0	0	0	0	0
95	100	100	100	100	100	100	36	100	100	100	11	64	100	100
96	0	0	0	0	0	0	0	0	0	0	0	0	0	0
97	100	0	0	0	0	100	0	0	0	0	0	0	0	0
98	23	100	0	0	0	0	0	0	93	0	0	0	0	0

Table 15 Detailed number of function evaluations of the tested optimization algorithms for E4

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
1	–	–	–	–	–	–	–	–	–	–	–	–	–	–
3	23,966	–	–	–	–	–	–	–	–	–	–	–	–	–
4	3417	–	–	2941	1723	8969	–	–	9742	21,999	–	9744	–	17,921
7	1095	4547	–	3334	10,839	6899	–	–	1134	20,830	–	–	15,842	23,744
14	24,413	–	–	–	–	–	–	–	–	–	–	–	–	–
21	–	–	–	–	–	–	–	–	–	–	–	–	–	–
22	–	–	–	–	–	–	–	–	–	–	–	–	–	–
23	–	–	–	–	–	–	–	–	–	–	–	–	–	–
24	201	301	200	201	212	401	101	201	201	201	101	3	152	151
33	1683	5722	–	5547	–	24,551	–	–	2303	–	–	–	12,052	–
41	6387	–	–	–	917	4790	–	23,901	8652	18,982	–	7264	–	11,923
45	–	–	–	–	–	–	–	–	–	–	–	–	–	–
50	1638	17,101	–	1505	–	15,101	–	23,521	–	–	–	–	–	9395
51	1773	14,501	–	1647	–	17,751	14,901	22,601	–	–	–	–	–	9078
52	1291	3453	–	18,407	641	2741	278	12,257	903	6860	–	8703	1462	1679
53	940	1987	–	3468	892	2144	1404	14614	997	6707	2624	1081	2474	9611
57	–	–	–	–	–	–	–	–	–	–	–	–	–	–
58	–	–	–	–	–	–	–	–	–	–	–	–	–	–
61	5288	–	–	22,781	3880	5402	–	–	–	19717	–	17,557	20,363	14155
67	–	–	–	–	–	–	–	–	–	–	–	–	–	–
69	–	–	–	–	–	–	–	–	–	–	–	–	–	–
70	–	–	–	–	–	–	–	–	–	–	–	–	–	–
73	–	–	–	–	–	–	–	–	–	–	–	–	–	–
74	–	–	–	–	–	–	–	–	–	–	–	–	–	–
78	2185	5869	–	23,801	2166	4589	–	23,051	5120	15,554	–	19,713	–	13,641

Table 15 continued

#	MVPA	PSO	GA	DE	EM	TLBO	GSA	LCA	BH	DSA	FA	SA	HS	ABC
79	13,936	–	–	–	–	16574	–	–	–	–	–	–	–	–
80	–	–	–	–	–	–	–	–	–	–	–	–	–	–
92	14,883	–	–	–	–	–	–	–	–	–	–	–	–	–
93	–	–	–	–	–	–	–	–	–	–	–	–	–	–
95	246	1009	247	261	518	605	11,948	2505	1302	4677	15,046	9337	2773	1348
96	–	–	–	–	–	–	–	–	–	–	–	–	–	–
97	14,483	–	–	–	–	17,381	–	–	–	–	–	–	–	–
98	13,575	18,739	–	–	–	–	–	–	7547	–	–	–	–	–

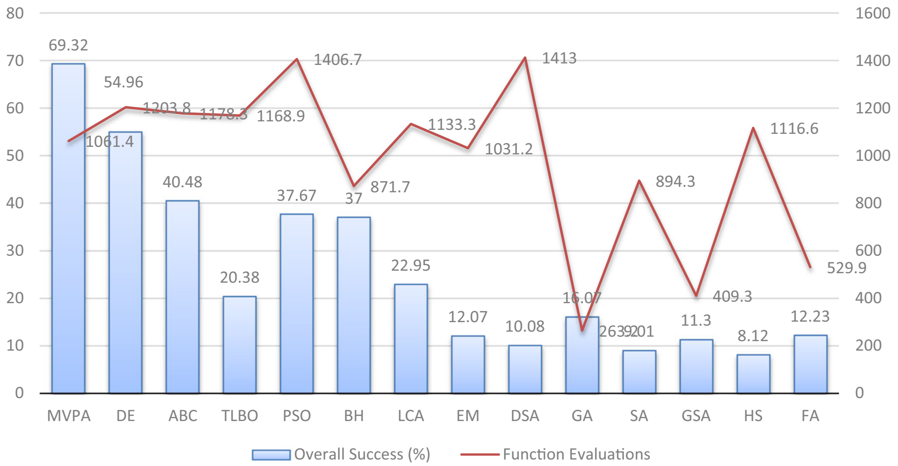

Fig. 4 Performances of the tested optimization algorithms for E1

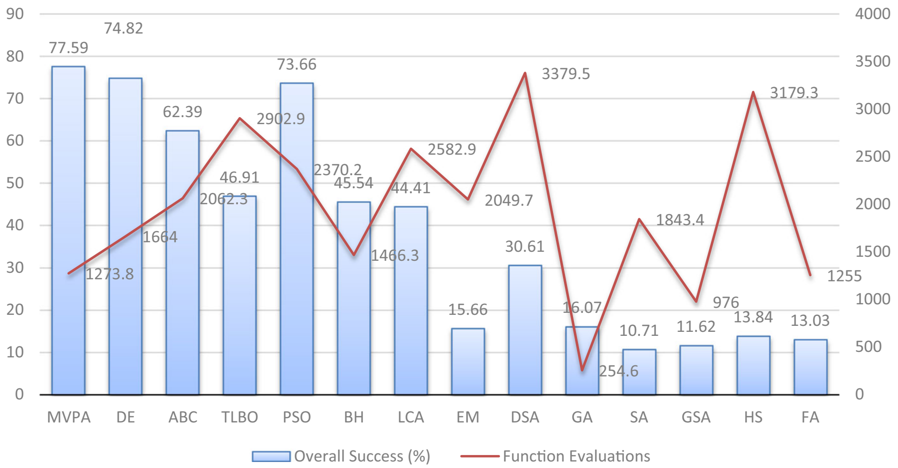

Fig. 5 Performances of the tested optimization algorithms for E2

Finally, it has been demonstrated that the MVPA is conceptually very simple, efficient, fast, reliable and easy to implement and/or to use. However, the MVPA is in its infancy, in other words it is still the fruit of only one mind, therefore the author hopes that this paper inspires future works to improve the proposed algorithm and apply it to various problems and in different fields. Moreover, it is worth to mention

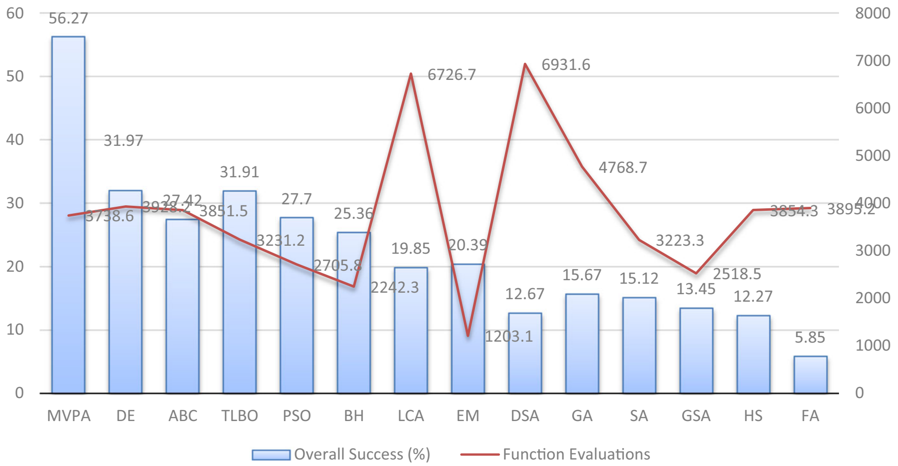

Fig. 6 Performances of the tested optimization algorithms for E3

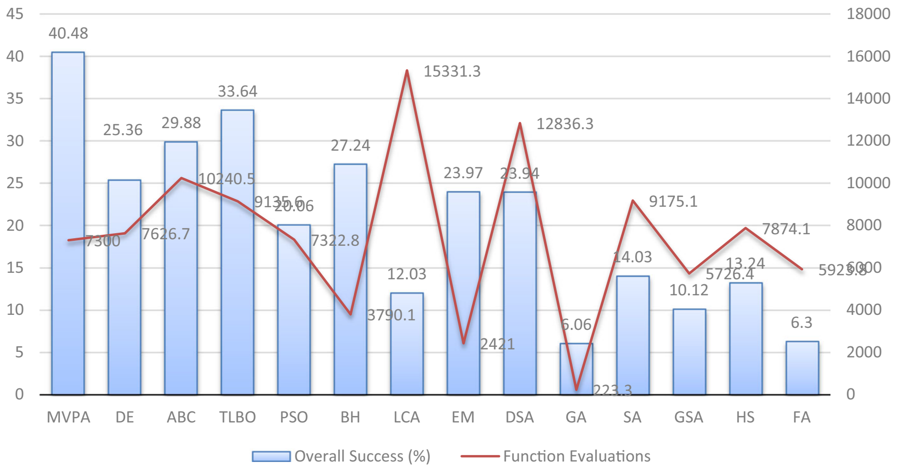

Fig. 7 Performances of the tested optimization algorithms for E4

that, in some preliminary computational study it has been found that TLBO is slightly better on higher dimensions >50. This point is one of the challenges of the future versions of the MVPA i.e. exploring higher dimensions' problems.

References

Adorio E, Diliman U (2005) MVF–multivariate test functions library in c for unconstrained global optimization. http://www.geocities.ws/eadorio/mvf.pdf. Accessed 01 May 2017

Ali MM, Khompatraporn C, Zabinsky ZB (2005) A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. J Glob Optim 31:635–672. doi:10.1007/s10898-004-9972-2

Askarzadeh A (2014) Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. Commun Nonlinear Sci Numer Simul 19:1213–1228. doi:10.1016/j.cnsns.2013.08.027

Binitha S, Sathya SS (2012) A survey of bio inspired optimization algorithms. Int J Soft Comput Eng 2:137–151

Birbil Şİ, Fang S-C (2003) An electromagnetism-like mechanism for global optimization. J Glob Optim 25:263–282. doi:10.1023/A:1022452626305

Bouchekara H (2013a) Electromagnetic device optimization based on electromagnetism-like mechanism. Appl Comput Electromagn Soc 28:241–248

Bouchekara H (2013b) optimal design of electromagnetic devices using a black-hole-based optimization technique. IEEE Trans Magn 49:5709–5714. doi:10.1109/TMAG.2013.2277694

Bouchekara HRE-H, Abido MA (2014) Optimal power flow using differential search algorithm. Electr Power Compon Syst 42:1683–1699. doi:10.1080/15325008.2014.949912

Bouchekara HREH, Abido MA, Boucherma M (2014a) Optimal power flow using teaching–learning-based optimization technique. Electr Power Syst Res 114:49–59. doi:10.1016/j.epsr.2014.03.032

Bouchekara HREH, Abido MA, Chaib AE, Mehasni R (2014b) Optimal power flow using the league championship algorithm: a case study of the Algerian power system. Energy Convers Manag 87:58–70. doi:10.1016/j.enconman.2014.06.088

Boudjefdjouf H, Mehasni R, Orlandi A et al (2015) Diagnosis of multiple wiring faults using time-domain reflectometry and teaching–learning-based optimization. Electromagnetics 35:10–24. doi:10.1080/02726343.2015.971659

Brown K (2015) The combination of probabilities. http://www.mathpages.com/home/index.htm. Accessed 1 Jan 2015

Civicioglu P (2012) Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. Comput Geosci 46:229–247. doi:10.1016/j.cageo.2011.12.011

Črepinšek M, Liu S-H, Mernik L (2012) A note on teaching–learning-based optimization algorithm. Inf Sci (Ny) 212:79–93. doi:10.1016/j.ins.2012.05.009

Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth International symposium on micro machine and human science, pp 39–43

Elsayed SM, Sarker RA, Essam DL (2014) A new genetic algorithm for solving optimization problems. Eng Appl Artif Intell 27:57–69. doi:10.1016/j.engappai.2013.09.013

Fister I, Yang XS, Brest J, Fister D (2013) A brief review of nature-inspired algorithms for optimization. Electrotech Rev 80:116–122

Gavana A (2015) Global optimization benchmarks and AMPGO. http://infinity77.net/global_optimization/index.html. Accessed 1 Jan 2015

Geem ZW, Kim J-H, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76:60–68. doi:10.1177/003754970107600201

Hatamlou A (2013) Black hole: a new heuristic optimization approach for data clustering. Inf Sci (NY) 222:175–184. doi:10.1016/j.ins.2012.08.023

Holland John H (1975) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press, Ann Arbor

Husseinzadeh Kashan A (2011) An efficient algorithm for constrained global optimization and application to mechanical engineering design: league championship algorithm (LCA). CAD Comput Aided Des 43:1769–1792. doi:10.1016/j.cad.2011.07.003

Jamil M, Yang XS (2013) A literature survey of benchmark functions for global optimisation problems. Int J Math Model Numer Optim 4:150. doi:10.1504/IJMMNO.2013.055204

Jolai F, Tavakkoli-Moghaddam R, Golmohammadi A, Javadi B (2012) An Electromagnetism-like algorithm for cell formation and layout problem. Expert Syst Appl 39:2172–2182. doi:10.1016/j.eswa.2011.07.030

Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University

Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 139:18–27. doi:10.1016/j.compstruc.2014.04.005

Kennedy J, Eberhart R (1995) Particle swarm optimization. Proceedings of 1993 IEEE international conference on neural networks, vol 4, pp 1942–1948. doi: 10.1109/ICNN.1995.488968

Kirkpatrick S, Gellat CD Jr, Vecchi MP (1983) Optimization by simmulated annealing. Science 220(80):671–680. doi:10.1007/BF01009452

Mishra SK (2013) Global optimization of some difficult benchmark functions by host-parasite coevolutionary algorithm. Econ Bull 33:1–18

Oxford (2015) Oxford learner's dictionaries. http://www.oxfordlearnersdictionaries.com/

Pourali Z, Aminnayeri M (2012) A novel discrete league championship algorithm for minimizing earliness/tardiness penalties with distinct due dates and batch delivery consideration. In: Advanced intelligent computing, pp 139–146

Price KV, Storn RM, Lampinen JA (2005) Differential evolution—a practical approach to global optimization. Springer, Berlin

Qing A (2009) Differential evolution fundamentals fundamentals and applications in electrical engineering. Wiley, New York

Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Des 43:303–315. doi:10.1016/j.cad.2010.12.015

Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci (NY) 179:2232–2248. doi:10.1016/j.ins.2009.03.004

Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359. doi:10.1023/A:1008202821328

Tuncer A, Yildirim M (2012) Dynamic path planning of mobile robots with improved genetic algorithm. Comput Electr Eng 38:1564–1572. doi:10.1016/j.compeleceng.2012.06.016

Xing B, Gao W-J (2014) Innovative computational intelligence: a rough guide to 134 clever algorithms. Springer, Cham

Yang X (2009) Firefly algorithms for multimodal optimization. In: Watanabe O, Zeugmann T (eds) 5th symposium on stochastic algorithms, foundations and applications. Lecture notes in computer science, vol 5792, pp 169–178

Yang X (2010a) Nature-Inspired Metaheuristic Algorithms, 2nd edn. Luniver Press, Second Edi

Yang X-S (2010b) Engineering optimization an introduction with metheuristic application. Wiley, New York

Yang X-S (2011) Review of metaheuristics and generalized evolutionary walk algorithm 3:14. doi:10.1504/IJBIC.2011.039907