Gerçek Dünya Kısıtlı Optimizasyon Problemlerinin Çözümü için En Değerli Oyuncu Algoritmasının Değerlendirilmesi

Araştırma Makalesi/Research Article

Sait Ali UYMAZ

Department of Computer Engineering, Konya Technical University, Konya, Turkey
sauymaz@ktun.edu.tr
(Geliş/Received:06.02.2021; Kabul/Accepted:10.08.2021)
DOI: 10.17671/gazibtd.875820

Özet— Gerçek-dünya kısıtlı optimizasyon problemlerinin, karar değişkenlerine ek olarak kısıtlamaları ve yerel minimum noktaları vardır. Kısıtlamalar nedeniyle bu problemlerin arama alanları çok küçük olduğu için çözülmesi zor ve zaman alıcıdır. Son yıllarda, bu tür problemleri çözmek için birçok yeni meta-sezgisel algoritma önerilmiş ve kısıt işleme teknikleriyle birleştirilmiştir. Spor etkinliklerinden esinlenerek yakın zamanda önerilen bir meta-sezgisel optimizasyon algoritması olan En Değerli Oyuncu Algoritması (MVPA), matematiksel test fonksiyonları üzerinde test edilmiştir. Bu çalışmada, MVPA algoritması kısıt işleme teknikleri ve bazı modifikasyonlar ile birleştirilerek 19 kısıtlı gerçek dünya mühendislik optimizasyon problemi üzerinde test edilmiştir. Sonuçlar, kısıtları sağlayan uygun çözümler bulmada yüksek bir başarı oranı göstermiştir.

Anahtar Kelimeler— en değerli oyuncu algoritması, kısıt işleme tekniği, metasezgisel, mühendislik problemleri

Evaluation of the Most Valuable Player Algorithm for Solving Real-World Constrained Optimization Problems

Abstract— Real-world constrained optimization problems have constraints and local minimums in addition to decision variables. They are time consuming and difficult to solve since the search spaces of these problems are very small due to the constraints. In recent years, many new metaheuristic algorithms have been proposed and combined with constraint handling techniques to solve such problems. The most valuable player algorithm (MVPA), a recently proposed metaheuristic optimization algorithm, inspired by sports events, has been tested on mathematical benchmark functions. In this study, the MVPA algorithm is combined with constraint handling techniques and some modifications and tested on 19 real-world constrained engineering optimization problems. The results showed a high success rate in finding feasible solutions.

Keywords— constraint handling technique, engineering problems, metaheuristic, the most valuable player algorithm

1. INTRODUCTION

Due to its nature, real world optimization problems often have constraints involving both equality and inequality. The most challenging part of this type of problems is the process of constraint handling. In the process of solving such problems, constraint handling technique is the key role to reach the successful results. The constraint (equality and inequality) functions narrow the search space and make the search process more difficult. The existence and excessive number of the constraints increase the complexity of the problem. Original versions of metaheuristic optimization methods were generally designed to search in unconstrained search spaces [1-3]. Constraint handling techniques are added to these optimization methods to guide the search in regions where suitable solutions are available. Constrained problems can be formally described as to [4]:

$$\begin{array}{llll} \text{Minimize} & f(X), & X=(x_1,x_2,\ldots,x_d)\in R^d & \\ \text{Subject to:} & g_i(X) \le 0, & i = 1,2,\ldots,p & \\ \text{and} & h_j(X) = 0, & j = 1,2,\ldots,m & (1) \\ \text{where} & l_i \le x_i \le u_i, & i = 1,2,\ldots,d & \end{array}$$

where $f(X)$ is the objective function and d is the dimension of the problem. $g(x)$ is the inequality constrained function and p is its number. $h(x)$ is the equality function and m is the number of these functions. l and u are lower and upper boundaries which define the search space, respectively.

In general, equality functions are transformed into inequality form same as Eq. 2 as follows:

$$|h_j(X)| - \epsilon \le 0, \quad j = 1,2,\ldots,m \qquad (2)$$

Where, ϵ is set to a small value.

In the literature, many different constraint handling techniques have been proposed to adapt algorithms to manage the search process in the regions where appropriate solutions exist [1, 2, 5-7]. Among these techniques, the oldest and the most widely used are the penalty methods [8-12]. The common ground of these approaches is to increase the fitness values of individuals in infeasible regions with penalty score calculated by the penalty functions [2]. Many metaheuristic optimization methods inspired from different disciplines have been proposed to solve the real-world constrained optimization problems [13-19].

The most valuable player algorithm (MVPA) proposed by Bouchekara [20] to solve unconstrained optimization problems. The performance of the algorithm has been proved on several benchmarks functions. MVPA has achieved the best overall results with lower function evaluation numbers. Due to its performance, it has been used in the solution of many different problems [21-24].

In this paper, the most valuable player algorithm (MVPA) which is inspired from the team sports events has been incorporated with constraint handling techniques and mutation strategy in order to deal with constrained real-world engineering optimization problems. Dynamic and self-adaptive constraint handling techniques were used to handle constraints in this study. Additionally, the effect of the number of elite players and teams in the population on the performance of the algorithm was evaluated.

2. MOST VALUABLE PLAYER ALGORITHM

Most Valuable Player Algorithm (MVPA) is a meta-heuristic algorithm proposed by Bouchekara [20]. MVPA is inspired from sport events where a group of teams which has a group of players compete to be champions. Two activities in these sport events underlie at the core of this algorithm. The first is the competition between teams to win the championship and the second is the competition between players to become the most valuable player (MVP).

In MVPA, a player represents a solution in the search domain and is represented in Eq. 3. A team consists of a group of players (Eq. 4) and the entire population consists of teams (Eq. 5).

$$Player_i = [x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,D}] \qquad (3)$$

$$Team_j = [Player_{j,1}, Player_{j,2}, Player_{j,3}, \ldots, Player_{j,T}] \qquad (4)$$

$$Population = [Team_1, Team_2, Team_3, \ldots, Team_{TS}] \qquad (5)$$

Where, D is dimension of the problems, T is the number of players in each team and TS is the number of teams in the population.

At the beginning of the algorithm, players are randomly generated in the problem space. After that, players are grouped into teams. The number of teams in the population is determined by the TeamSize parameter in the algorithm. Each player tries to become the best player of their teams (franchise player) and MVP at the same time. After initialization, the competition phase starts with two different types, one between players (Individual competition) and the other between teams (Team competition) [20].

Players are affected by two different types of players when they try to improve themselves in individual competition as follows:

$$Team_j = Team_j + rand \times (FranchisePlayer_j - Team_j) + 2 \times rand \times (MVP - Team_j) \qquad (6)$$

Where, $rand$ is a uniformly distributed random number between 0 and 1.

It can be seen in Eq. 6 that improvement of players is performed together (as a team) as in training.

In team competition, the selected $Team_j$ play against another randomly selected $Team_k$. Which team wins is determined by a probability-based mechanism described in [20]. If $Team_j$ wins the competition, the players of $Team_j$ are updated as in Eq. 7, otherwise as in Eq. 8.

$$Team_j = Team_j + rand \times (Team_j - FranchisePlayer_k) \quad (7)$$

$$Team_j = Team_j + rand \times (FranchisePlayer_k - Team_j) \quad (8)$$

After individual and team competition updating processes, greediness process is applied. Among the pre-update and post-update populations, those with better objective function are selected.

In the elitism process, the best individuals whose number is determined by the *ElitePlayer* parameter is replaced the worst individuals in the same number.

In the remove duplicate process, if two individuals in the population are the same, one of them is removed from the population and a new individual is produced in its place as described in Elsayed, et al. [25].

Pseudo code of most valuable player algorithm

Objective function, f(x)
Define parameters (TeamSize, ElitePlayer, NumberOfPlayers, MaxFEs)
Initialize a population of players with random solutions and players are grouped into teams.
Evaluate players and find Franchise Players of each team and MVP
For *i = 1: MaxFEs*
  For *j = 1: TeamSize*
    Randomly select $Team_k$ $(j \neq k)$
    Individual Competition for $Team_j$ *by Eq. 6*
    Team Competition between $Team_j$ *and* $Team_k$ *by Eqs. 7-8*
  end For
  Evaluate new players
  Determine new population with greediness process
  Elitism process
  Remove Duplicate process
  Update Franchise Players and MVP
end For
Output: MVP

3. PROPOSED METHOD

Original MVPA has been designed for unconstrained problems and tested on 100 mathematical benchmarks. To evaluate the performance of the method, the results were compared with 13 well-known meta-heuristic methods. Among these methods, MVPA achieved the best results [20].

In this paper, MVPA has been adapted for real-world constrained optimization problems. For this adaptation, two constraint handling techniques which are dynamic penalty and self-adaptive penalty have been added to the algorithm. In addition, due to the high complexity of the constrained problems, the mutation operator with different elite player number and team size was applied.

3.1. Mutation Strategy

Exploring the solution space of the constrained problems is important to achieve the quality results. Constraints make it difficult to find a feasible solution in search space. Therefore, exploring ability of the algorithm is of great importance. In this study, mutation strategy has been applied to improve the exploring ability of the algorithm.

After competition phases, mutation strategy is applied to randomly selected players in each iteration. Mutation is applied in all problem dimensions and the number of players to be mutated is as large as the problem dimension. The selected players are generated randomly in search space as follows:

$$Player_i = LB + rand \times (UB - LB) \quad (9)$$

where, i is a vector of selected players to be mutated, *LB* and *UB* contain minimum and maximum values of vectors for each dimension, respectively.

3.2. Constraints Handling Strategy

The solutions satisfied all the constraints are called feasible solutions. On the contrary, the solutions that do not satisfy at least one constraint are called infeasible solutions. Searching strategy on constrained optimization problems focuses on finding feasible solutions that reach the global optimum. In the literature, different kinds of strategies have been proposed to handle constraints. In this study, two strategies based on penalty functions which are dynamic and self-adaptive penalty techniques were used to give the algorithm the ability to handle constraints.

3.2.1. Dynamic Penalty

In this technique, the aim is to increase (or decrease depending on the objective function) the fitness values of infeasible individuals by adding penalty values. These penalty values increase in proportion to the constraint function values. Transformed objective function values are calculated by Eqs. 10-11.

In this technique, objective function and constraint functions are calculated separately but evaluated together for each individual (player). P(x) is a penalty value which is calculated dynamically. If an individual move away from the feasible areas, its penalty value increases as in Eq. 11 and therefore extended objective function value increases as in Eq.10. This causes the searching process to be concentrated in the feasible areas.

$$\emptyset(x) = f(x) + p(x) \quad (10)$$

$$p(x) = \sum_{i=1}^{p} r1 \times \max\,(g_i(x), 0)^2 + \sum_{j=1}^{m} r2 \times (|h_i(x)| - \epsilon) \quad (11)$$

φ(x) is the extended objective function and *p(x)* is the penalty value. *r1* and *r2* are penalty factors.

3.2.2. Self-Adaptive Penalty

The self-adaptive penalty function used in this study was proposed in [9]. Two types of penalties are calculated when calculating the fitness value of individuals. The penalty values added are related to the number of currently eligible individuals. As the number of eligible individuals increases, their penalty values decrease. Combined fitness value is calculated by Eq. 12.

$$F(x) = d(x) + p(x) \tag{12}$$

Where $d(x)$ is the distance function and $p(x)$ is the penalty function. The detailed formulization of $d(x)$ and $p(x)$ are found in [9, 26].

4. EXPERIMENTS

Experimental studies have been carried out in two stages. The first stage is to add constraint handling techniques to the algorithm to solve constrained optimization problems. The second stage is to examine the effects of mutation strategy and parameter settings. Parameter analysis focuses on the number of players to be applied elitism process and the number of teams in relation to the population.

All experiments were conducted on 19 real-world constrained mechanical engineering problems in Table 1. These problems have decision variables between 2 and 30, inequality constraints between 1 and 86, and equality constraints between 0 and 3 [4]. In Table 1, D is the number of decision variables, g is the number of inequality constraints and h is the number of equality constraints. f(x*) is the best known feasible solution. These problems have non-linear functions (objective and constraint) with many local minimum and very small feasible regions compared to the solution spaces [27].

The MATLAB codes of this 19 real-world engineering problems are available in "https://github.com/P-N-Suganthan/2020-RW-Constrained-Optimization". This benchmark suite has a diverse set of non-linear and non-convex functions and constraints with different levels of difficulty. Objective functions, constraints, bounds, decision variables, constants and values of these constants are described and presented in Kumar et al [4].

4.1. Experimental Settings

All experiments have been implemented on MATLAB in a PC having Microsoft Windows 10 operating system with INTEL Core i7 CPU and 16 Gb RAM. The maximum number of functions evaluation numbers ($MaxFEs$) is determined by a rule using the number of decision variables of the problems as in Eq. 13.

$$MaxFEs = \begin{cases} 1 \times 10^5, & if\ D \leq 10 \\ 2 \times 10^5, & if\ 10 < D \leq 30 \end{cases} \tag{13}$$

Algorithm is implemented independently 25 times for each engineering problem in Table 1. The population number was set to 100 in this study as recommended in [20]. Results of the proposed method for 25 runs are evaluated in terms of minimum, mean, median, worst and standard deviation values. In addition, to evaluate the performance of the method, three criteria which are important for constraint problems were adopted in this study as follows:

Table 1. Summary of the 19 real-world constrained mechanical engineering problems

Problem	**Name**	**D**	**g**	**h**	***f(x*)***
1	Weight Minimization of a Speed Reducer	7	11	0	2.9944244658E+03
2	Optimal Design of Industrial refrigeration System	14	15	0	3.2213000814E-02
3	Tension/compression spring design (case 1)	3	3	0	1.2665232788E-02
4	Pressure vessel design	4	4	0	5.8853327736E+03
5	Welded beam design	4	5	0	1.6702177263E+00
6	Three-bar truss design problem	2	3	0	2.6389584338E+02
7	Multiple disk clutch brake design problem	5	7	0	2.3524245790E-01
8	Planetary gear train design optimization problem	9	10	1	5.2576870748E-01
9	Step-cone pulley problem	5	8	3	1.6069868725E+01
10	Robot gripper problem	7	7	0	2.5287918415E+00
11	Hydro-static thrust bearing design problem	4	7	0	1.6254428092E+03
12	Four-stage gear box problem	22	86	0	3.5359231973E+01
13	10-bar truss design	10	3	0	5.2445076066E+02
14	Rolling element bearing	10	9	0	1.4614135715E+04
15	Gas Transmission Compressor Design (GTCD)	4	1	0	2.9648954173E+06
16	Tension/compression spring design (case 2)	3	8	0	2.6138840583E+00
17	Gear train design Problem	4	1	1	0.0000000000E+00
18	Himmelblau's Function	5	6	0	-3.0665538672E+04
19	Topology Optimization	30	30	0	2.6393464970E+00

1. Mean Constraint Violation (MV), which means average of constraint violation over 25 runs, is calculated as in Eq. 14.

$$\bar{v} = \frac{\sum_{i=1}^{p} \max(g_i(\vec{x}), 0) + \sum_{j=p+1}^{m} \max(|h_j(\vec{x})| - \epsilon, 0)}{m} \tag{14}$$

where ϵ is set to 0.0001.

2. Feasibility Rate (FR) is the ratio of runs with feasible solutions to the total number of runs. FR is calculated as in Eq. 15.

$$FR = \frac{Total\ Fesible\ Runs}{Total\ Runs} \quad (15)$$

3. Success Rate (SR) is the ratio of runs obtained from feasible solution satisfying Eq. 16 to the total number of runs.

$$f(\bar{x}) - f(\bar{x}^*) \leq 10^{-4} \quad (16)$$

4.2. Performance Comparison

Two-stage experiments were conducted in this study. Initially, standard parameters recommended by [20] were used to demonstrate performance of the method on constrained engineering problems.

Stage 1

- Performance of the algorithm with constraint handling techniques on real-world constrained engineering problems.

Stage 2

- Performance of the algorithm with and without mutation strategy.
- The effect of ElitePlayer parameter on the algorithm.
- The effect of TeamSize parameter on the algorithm.

5. RESULTS

The detailed results obtained from the algorithm with constraint handling techniques are demonstrated in this section. Results were analyzed in two stages. First, the pure algorithm was combined with the handling technique and the results are shown in Table 2. In the second stage, the effect of the mutation strategy on performance of the method is shown in Table 3. Table 4 and Table 5 show the effects of number of elite player and teams, respectively. The comparisons are based on the mean values, FR, MV and SR criteria of 25 independent runs. As seen in Table 2, dynamic penalty technique is significantly superior to self-adaptive penalty technique. Self-adaptive penalty technique is only better than dynamic penalty technique on four-stage gear box problem which has many constraints. In second stage studies, dynamic penalty technique was used because of its performance.

Table 2. Performance of MVPA with constraint handling techniques

	MVPA with Dynamic Penalty				**MVPA with Self-Adaptive Penalty**			
Problem	**Mean**	**FR**	**MV**	**SR**	**Mean**	**FR**	**MV**	**SR**
1	2.994E+03	**1.00**	0.00E+00	0.96	2.994E+03	0.92	2.34E-10	0.68
2	1.635E-01	**0.84**	1.01E-02	0.24	1.503E+00	0.96	1.12E-08	0.00
3	1.272E-02	1.00	0.00E+00	**1.00**	1.278E-02	1.00	0.00E+00	0.84
4	6.410E+03	**1.00**	0.00E+00	0.00	6.108E+03	0.00	7.70E+00	0.00
5	1.670E+00	**1.00**	0.00E+00	0.96	1.672E+00	0.88	3.80E-08	0.36
6	2.639E+02	**1.00**	0.00E+00	1.00	2.639E+02	0.64	5.05E-12	0.64
7	2.352E-01	1.00	0.00E+00	**1.00**	2.806E-01	1.00	0.00E+00	0.32
8	7.689E-01	**0.96**	9.09E-04	0.00	6.683E-01	0.88	1.09E+04	0.00
9	1.705E+01	**1.00**	0.00E+00	0.12	1.739E+01	0.00	1.18E-04	0.00
10	**3.016E+00**	1.00	0.00E+00	0.00	8.210E+00	1.00	0.00E+00	0.00
11	2.552E+03	**1.00**	0.00E+00	0.00	2.068E+03	0.68	5.76E+01	0.00
12	1.123E+02	0.00	1.01E+00	0.00	8.371E+01	**0.64**	1.14E+01	0.00
13	5.321E+02	**1.00**	0.00E+00	0.00	5.324E+02	0.00	2.20E-04	0.00
14	1.696E+04	**1.00**	0.00E+00	0.00	1.698E+04	0.16	1.44E-03	0.00
15	2.965E+06	**1.00**	0.00E+00	0.00	2.965E+06	0.64	2.05E-08	0.00
16	2.699E+00	**1.00**	0.00E+00	0.00	2.659E+00	0.04	1.09E+03	0.00
17	2.416E-30	1.00	0.00E+00	**1.00**	1.129E+00	1.00	0.00E+00	0.04
18	-3.067E+04	**1.00**	0.00E+00	1.00	-3.067E+04	0.00	3.85E-06	0.00
19	2.639E+00	1.00	0.00E+00	**1.00**	8.808E+01	1.00	0.00E+00	0.00

Table 3 shows that,

- According to FR metric, the rate of finding feasible solutions with the mutation strategy increases in problems 2, 8 and 12.
- According to the SR metric, the quality of the feasible solutions found with the mutation strategy increases in problems 1, 2, 5, 9 and 11.
- Although the FR, MV and SR values are the same in problems 4, 10, 13 and 17, the quality of the average values of the solutions increases with the mutation strategy.
- When the mutation strategy was not used, FR, MV and SR values did not change in problems 3 and 16, but partial improvement was observed in the average values.
- Solution qualities remained the same for 6, 7, 14, 15, 18 and 19 problems.

Table 3. Results of the constrained MVPA with and without mutation strategy

	Without Mutation				With Mutation			
Problem	**Mean**	**FR**	**MV**	**SR**	**Mean**	**FR**	**MV**	**SR**
1	2.994E+03	1.00	0.000E+00	0.96	2.994E+03	1.00	0.000E+00	**1.00**
2	1.635E-01	0.84	1.014E-02	0.24	4.936E-02	**1.00**	0.000E+00	**0.32**
3	**1.272E-02**	1.00	0.000E+00	1.00	1.277E-02	1.00	0.000E+00	1.00
4	6.410E+03	1.00	0.000E+00	0.00	**6.371E+03**	1.00	0.000E+00	0.00
5	1.670E+00	1.00	0.000E+00	0.96	1.670E+00	1.00	0.000E+00	**1.00**
6	2.639E+02	1.00	0.000E+00	1.00	2.639E+02	1.00	0.000E+00	1.00
7	2.352E-01	1.00	0.000E+00	1.00	2.352E-01	1.00	0.000E+00	1.00
8	7.689E-01	0.96	9.091E-04	0.00	7.484E-01	**1.00**	0.000E+00	0.00
9	1.705E+01	1.00	0.000E+00	0.12	1.628E+01	1.00	0.000E+00	**0.40**
10	3.016E+00	1.00	0.000E+00	0.00	**2.810E+00**	1.00	0.000E+00	0.00
11	2.552E+03	1.00	0.000E+00	0.00	2.284E+03	1.00	0.000E+00	**0.08**
12	1.123E+02	0.00	1.013E+00	0.00	1.218E+02	**0.04**	**5.286E-01**	0.00
13	5.321E+02	1.00	0.000E+00	0.00	**5.281E+02**	1.00	0.000E+00	0.00
14	1.696E+04	1.00	0.000E+00	0.00	1.696E+04	1.00	0.000E+00	0.00
15	2.965E+06	1.00	0.000E+00	0.00	2.965E+06	1.00	0.000E+00	0.00
16	**2.699E+00**	1.00	0.000E+00	0.00	2.903E+00	1.00	0.000E+00	0.00
17	2.416E-30	1.00	0.000E+00	1.00	**0.000E+00**	1.00	0.000E+00	1.00
18	-3.067E+04	1.00	0.000E+00	1.00	-3.067E+04	1.00	0.000E+00	1.00
19	2.639E+00	1.00	0.000E+00	1.00	2.639E+00	1.00	0.000E+00	1.00

Table 4. Results of constrained MVPA with mutation strategy and different number of elite players (N: Population size)

	N/2				N/3				N/4			
Prob.	**Mean**	**FR**	**MV**	**SR**	**Mean**	**FR**	**MV**	**SR**	**Mean**	**FR**	**MV**	**SR**
1	2.994E+03	1.00	0.00E+00	1.00	2.994E+03	1.00	0.00E+00	1.00	2.994E+03	1.00	0.00E+00	1.00
2	4.664E-02	0.92	5.07E-03	0.28	4.936E-02	**1.00**	0.00E+00	**0.32**	1.840E+00	0.96	2.54E-03	0.08
3	1.279E-02	1.00	0.00E+00	1.00	1.277E-02	1.00	0.00E+00	1.00	**1.275E-02**	1.00	0.00E+00	1.00
4	**6.091E+03**	1.00	0.00E+00	0.00	6.371E+03	1.00	0.00E+00	0.00	6.371E+03	1.00	0.00E+00	0.00
5	1.670E+00	1.00	0.00E+00	0.92	1.670E+00	1.00	0.00E+00	**1.00**	1.670E+00	1.00	0.00E+00	0.88
6	2.639E+02	1.00	0.00E+00	0.96	2.639E+02	1.00	0.00E+00	**1.00**	2.639E+02	1.00	0.00E+00	**1.00**
7	2.352E-01	1.00	0.00E+00	1.00	2.352E-01	1.00	0.00E+00	1.00	2.352E-01	1.00	0.00E+00	1.00
8	**6.882E-01**	1.00	0.00E+00	0.00	7.484E-01	1.00	0.00E+00	0.00	7.791E-01	1.00	0.00E+00	0.00
9	1.705E+01	1.00	0.00E+00	0.12	**1.628E+01**	1.00	0.00E+00	**0.40**	1.709E+01	1.00	0.00E+00	0.16
10	2.903E+00	1.00	0.00E+00	0.00	**2.810E+00**	1.00	0.00E+00	0.00	2.963E+00	1.00	0.00E+00	0.00
11	2.572E+03	1.00	0.00E+00	0.00	**2.284E+03**	1.00	0.00E+00	**0.08**	2.753E+03	1.00	0.00E+00	0.00
12	9.999E+01	0.00	7.61E-01	0.00	1.218E+02	**0.04**	5.29E-01	0.00	1.027E+02	0.00	5.75E-01	0.00
13	5.322E+02	1.00	0.00E+00	0.00	**5.281E+02**	1.00	0.00E+00	0.00	5.313E+02	1.00	0.00E+00	0.00
14	1.696E+04	1.00	0.00E+00	0.00	1.696E+04	1.00	0.00E+00	0.00	1.696E+04	1.00	0.00E+00	0.00
15	2.965E+06	1.00	0.00E+00	0.00	2.965E+06	1.00	0.00E+00	0.00	2.965E+06	1.00	0.00E+00	0.00
16	2.903E+00	1.00	0.00E+00	0.00	2.903E+00	1.00	0.00E+00	0.00	**2.699E+00**	1.00	0.00E+00	0.00
17	7.378E-25	1.00	0.00E+00	1.00	**0.000E+00**	1.00	0.00E+00	1.00	**0.000E+00**	1.00	0.00E+00	1.00
18	-3.067E+04	1.00	0.00E+00	1.00	-3.067E+04	1.00	0.00E+00	1.00	-3.067E+04	1.00	0.00E+00	1.00
19	2.639E+00	1.00	0.00E+00	1.00	2.639E+00	1.00	0.00E+00	1.00	2.639E+00	1.00	0.00E+00	1.00

In the light of these analyzes, it has been seen that the mutation strategy has a successful effect on the results. Therefore, the method with the mutation strategy was used in subsequent analyzes.

As seen in Table 4,

- For all parameters, results are the same in six problems (1, 7, 14, 15, 18 and 19).
- FR values of results with all parameter values are the same except for problems 2 and 12. N/3 parameter results are better than others in these problems.
- In terms of quality of feasible solutions, N/3 parameter value is slightly better than others.

According to Table 5,

- When the number of teams was 20, 10 and 5, feasible solutions were found for all runs (25 runs) in 18, 16 and 14 problems, respectively.
- When FR values are equal for all parameters, according to SR values, 20 Teams better than others in problems 3, 5, 6, 9 and 11.
- In 10 problems where FR and SR values are equal for all team size, 20 Teams achieve better results in terms of mean values.

Table 5. The results of the constrained MVPA with mutation strategy and different number of team size

	TeamSize=5				TeamSize=10				TeamSize=20			
Prob.	**Mean**	**FR**	**MV**	**SR**	**Mean**	**FR**	**MV**	**SR**	**Mean**	**FR**	**MV**	**SR**
1	2.994E+03	**1.00**	0.000E+00	1.00	2.994E+03	0.96	2.961E-18	0.80	2.994E+03	**1.00**	0.000E+00	1.00
2	1.635E-01	**1.00**	0.000E+00	0.16	4.617E-02	0.96	2.536E-03	0.32	4.936E-02	**1.00**	0.000E+00	0.32
3	1.282E-02	1.00	0.000E+00	0.76	1.287E-02	1.00	0.000E+00	0.92	1.277E-02	1.00	0.000E+00	**1.00**
4	6.410E+03	1.00	0.000E+00	0.00	**6.371E+03**	1.00	0.000E+00	0.00	**6.371E+03**	1.00	0.000E+00	0.00
5	1.670E+00	1.00	0.000E+00	0.84	1.670E+00	1.00	0.000E+00	0.92	1.670E+00	1.00	0.000E+00	**1.00**
6	2.639E+02	1.00	0.000E+00	0.88	2.639E+02	1.00	0.000E+00	0.96	2.639E+02	1.00	0.000E+00	**1.00**
7	2.352E-01	1.00	0.000E+00	1.00	2.352E-01	1.00	0.000E+00	1.00	2.352E-01	1.00	0.000E+00	1.00
8	5.900E-01	0.92	1.000E-02	0.04	6.600E-01	**1.00**	0.000E+00	0.00	7.484E-01	**1.00**	0.000E+00	0.00
9	1.608E+01	0.96	4.752E-05	0.48	1.712E+01	**1.00**	0.000E+00	0.24	1.628E+01	**1.00**	0.000E+00	**0.40**
10	2.993E+00	1.00	0.000E+00	0.00	2.876E+00	1.00	0.000E+00	0.00	**2.810E+00**	1.00	0.000E+00	0.00
11	2.766E+03	0.92	1.085E-05	0.00	2.599E+03	1.00	0.000E+00	0.04	2.284E+03	**1.00**	0.000E+00	**0.08**
12	8.252E+01	0.00	6.886E-01	0.00	9.538E+01	0.00	6.618E-01	0.00	1.218E+02	**0.04**	5.286E-01	0.00
13	5.342E+02	1.00	0.000E+00	0.00	5.309E+02	1.00	0.000E+00	0.00	**5.281E+02**	1.00	0.000E+00	0.00
14	1.696E+04	1.00	0.000E+00	0.00	1.696E+04	1.00	0.000E+00	0.00	1.696E+04	1.00	0.000E+00	0.00
15	2.965E+06	1.00	0.000E+00	0.00	2.965E+06	1.00	0.000E+00	0.00	2.965E+06	1.00	0.000E+00	0.00
16	2.903E+00	1.00	0.000E+00	0.00	**2.699E+00**	1.00	0.000E+00	0.00	2.903E+00	1.00	0.000E+00	0.00
17	1.552E-26	1.00	0.000E+00	1.00	1.112E-30	1.00	0.000E+00	1.00	**0.000E+00**	1.00	0.000E+00	1.00
18	-3.067E+04	1.00	0.000E+00	1.00	-3.067E+04	1.00	0.000E+00	1.00	-3.067E+04	1.00	0.000E+00	1.00
19	2.639E+00	1.00	0.000E+00	1.00	2.639E+00	1.00	0.000E+00	1.00	2.639E+00	1.00	0.000E+00	1.00

Table 6. The detailed results of the proposed method

Problem	Min	Mean	Median	Max	Std.	FR	MV	SR
1	2.994E+03	2.994E+03	2.994E+03	2.994E+03	0.000E+00	**1.00**	0.00E+00	**1.00**
2	3.221E-02	4.936E-02	8.178E-01	5.935E+00	1.429E+00	**1.00**	0.00E+00	**0.32**
3	1.270E-02	1.277E-02	1.285E-02	1.322E-02	1.562E-04	**1.00**	0.00E+00	**1.00**
4	6.060E+03	6.371E+03	6.399E+03	7.333E+03	3.847E+02	**1.00**	0.00E+00	**0.00**
5	1.670E+00	1.670E+00	1.670E+00	1.670E+00	1.010E-11	**1.00**	0.00E+00	**1.00**
6	2.639E+02	2.639E+02	2.639E+02	2.639E+02	4.277E-05	**1.00**	0.00E+00	**1.00**
7	2.352E-01	2.352E-01	2.352E-01	2.352E-01	1.133E-16	**1.00**	0.00E+00	**1.00**
8	5.300E-01	7.484E-01	8.909E-01	2.157E+00	4.623E-01	**1.00**	0.00E+00	**0.00**
9	1.607E+01	1.628E+01	1.653E+01	1.712E+01	4.885E-01	**1.00**	0.00E+00	**0.40**
10	2.544E+00	2.810E+00	2.962E+00	4.866E+00	5.431E-01	**1.00**	0.00E+00	**0.00**
11	1.625E+03	2.284E+03	2.359E+03	4.474E+03	7.123E+02	**1.00**	0.00E+00	**0.08**
12	5.905E+01	1.218E+02	1.303E+02	2.932E+02	5.657E+01	**0.04**	5.29E-01	**0.00**
13	5.248E+02	5.281E+02	5.284E+02	5.326E+02	2.678E+00	**1.00**	0.00E+00	**0.00**
14	1.696E+04	1.696E+04	1.696E+04	1.706E+04	2.034E+01	**1.00**	0.00E+00	**0.00**
15	2.965E+06	2.965E+06	2.965E+06	2.965E+06	1.781E+01	**1.00**	0.00E+00	**0.00**
16	2.659E+00	2.903E+00	2.872E+00	3.203E+00	1.593E-01	**1.00**	0.00E+00	**0.00**
17	0.000E+00	0.000E+00	6.299E-20	1.220E-18	2.465E-19	**1.00**	0.00E+00	**1.00**
18	-3.067E+04	-3.067E+04	-3.067E+04	-3.067E+04	3.561E-12	**1.00**	0.00E+00	**1.00**
19	2.639E+00	2.639E+00	2.639E+00	2.639E+00	2.209E-15	**1.00**	0.00E+00	**1.00**

At the end of all the experimental studies performed, it is seen that, the best results were obtained using the dynamic penalty technique and mutation strategy. When effect of the number of elite player and teams in the population is examined, it is clear that the best parameter values are N/3 for the number of elite players and 20 for the number of teams. The detailed results of the final method are given in Table 6.

As seen in Table 6, proposed method has achieved the feasible solutions for all independent runs except problem 12. In problem 12, only one time has been found the feasible solution. According to the SR parameter, the quality of the feasible solutions in 8 problems is satisfying. In problem 2, 9 and 11, results are partly of good quality.

6. DISCUSSION AND CONCLUSION

This paper introduces a modification of MVPA designed for the real-world constrained engineering problems. Dynamic and self-adaptive penalty techniques depending on the value and count of constraint violation and mutation strategy were used for adaptation to such problems. Self-adaptive penalty is better than dynamic penalty in only one problem called Four-stage gear box problem. This problem is the most challenging problem in this set. This problem has 22 design parameters, 86 constraints and its feasible region is in ratio less than 0.0001 with many local minima. Self-adaptive penalty is more complicated and has better exploration ability. However, dynamic penalty uses the search process more efficiently. Additionally, the performance of the method was tested by setting two parameters with different values.

Although the proposed method is successful and consistent in finding feasible solutions, the quality of the solutions can be further improved. Therefore, as recommendation for further studies, the ability of exploration and exploitation of the algorithm can be strengthened by developing different exploitation/exploration balance strategies for MVPA. In addition, various adaptive versions of the algorithm can be developed.

REFERENCES

[1] R. Mallipeddi and P. N. Suganthan, "Ensemble of Constraint Handling Techniques", *IEEE Transactions on Evolutionary Computation*, 14(4), 561-579, 2010.

[2] E. Mezura-Montes and C. A. C. Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future", *Swarm and Evolutionary Computation*, 1(4), 173-194, 2011.

[3] Z. Michalewicz, “A survey of constraint handling techniques in evolutionary computation methods”, *Evolutionary Programming*, 4, 135-155, 1995.

[4] A. Kumar, G. H. Wu, M. Z. Ali, R. Mallipeddi, P. N. Suganthan, and S. Das, "A test-suite of non-convex constrained optimization problems from the real-world and some baseline results", *Swarm and Evolutionary Computation*, 56, 2020, doi: 10.1016/j.swevo.2020.100693.

[5] C. Y. Si, J. J. Hu, T. Lan, L. Wang, and Q. D. Wu, "A combined constraint handling framework: an empirical study", *Memetic Computing*, 9(1), 69-88, 2017.

[6] Z. Michalewicz and M. Schoenauer, "Evolutionary Algorithms for Constrained Parameter Optimization Problems", *Evolutionary computation*, 4(1), 1-32, 1996.

[7] C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art", *Computer methods in applied mechanics and engineering*, 191(11-12), 1245-1287, 2002.

[8] J. J. Liu, S. H. Zhang, C. Z. Wu, J. W. Liang, X. Y. Wang, and K. L. Teo, "A hybrid approach to constrained global optimization", *Applied Soft Computing*, 47, 281-294, 2016.

[9] B. Tessema and G. G. Yen, "A self adaptive penalty function based algorithm for constrained optimization", **2006 IEEE Congress on Evolutionary Computation (CEC)**, Vancouver, BC, Canada, 246-253, 2006.

[10] Z. J. Liu, B. Q. Lu, and Y. Cao, "A new optimization method based on restructuring penalty function for solving constrained minimization problems", **2006 IEEE International Conference on Granular Computing**, Atlanta, USA, 510-514, 2006.

[11] B. Tessema and G. G. Yen, "An Adaptive Penalty Formulation for Constrained Evolutionary Optimization", *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 39(3), 565-578, 2009.

[12] A. Cinar and M. Kiran, "The Performance of Penalty Methods on Tree-Seed Algorithm for Numerical Constrained Optimization Problems", *International Arab Journal of Information Technology*, 17(5), 799-807, 2020.

[13] K. M. Ang, W. H. Lim, N. A. M. Isa, S. S. Tiang, and C. H. Wong, "A constrained multi-swarm particle swarm optimization without velocity for constrained optimization problems", *Expert Systems with Applications*, 140, 2020, doi: 10.1016/j.eswa.2019.112882.

[14] K. Gupta, K. Deep, and J. C. Bansal, "Spider monkey optimization algorithm for constrained optimization problems", *Soft Computing*, 21(23), 6933-6962, 2017.

[15] D. Karaboga and B. Akay, "A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems", *Applied Soft Computing*, 11(3), 3021-3031, 2011.

[16] M. Kohli and S. Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems", *Journal of Computational Design and Engineering*, 5(4), 458-472, 2018.

[17] M. G. H. Omran and A. Salman, "Constrained optimization using CODEQ", *Chaos, Solitons & Fractals*, 42(2), 662-668, 2009.

[18] A. Trivedi, K. Sanyal, P. Verma, and D. Srinivasan, "A Unified Differential Evolution Algorithm for Constrained Optimization Problems", **2017 IEEE Congress on Evolutionary Computation (CEC)**, San Sebastián, Spain, 1231-1238, 2017.

[19] Ü. Atila, M. Dorterler, and İ. Şahin, “Yapay Alg Algoritmasının Tasarım Optimizasyon Problemlerindeki Performansı Üzerine Bir Çalışma: Basınç Yayı Örneği”, *Bilişim Teknolojileri Dergisi*, 11(4), 349-355, 2018, doi:10.17671/gazibtd.452992.

[20] H. R. E. H. Bouchekara, "Most Valuable Player Algorithm: a novel optimization algorithm inspired from sport", *Operational Research*, 20(1), 139-195, 2020.

[21] X. Liu, Q. F. Luo, D. Y. Wang, M. Abdel-Baset, and S. Q. Jiang, "An Improved Most Valuable Player Algorithm with Twice Training Mechanism", **In International Conference on Intelligent Computing**, Springer, Cham, 854-865, 2018.

[22] M. A. M. Ramli and H. R. E. H. Bouchekara, "Wind Farm Layout Optimization Considering Obstacles Using a Binary Most Valuable Player Algorithm", *Ieee Access*, 8, 131553-131564, 2020.

[23] K. Srilakshmi, P. R. Babu, and P. Aravindhababu, "An enhanced most valuable player algorithm based optimal power flow using Broyden's method", *Sustainable Energy Technologies and Assessments*, 42, 2020, doi: 10.1016/j.seta.2020.100801.

[24] A. Korashy, S. Kamel, A. R. Youssef, and F. Jurado, "Most Valuable Player Algorithm for Solving Direction Overcurrent Relays Coordination Problem", **Proceedings of 2019 International Conference on Innovative Trends in Computer Engineering (Itce 2019)**, Aswan, Egypt, 466-471, 2019.

[25] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "A new genetic algorithm for solving optimization problems", *Engineering Applications of Artificial Intelligence*, 27, 57-69, 2014.

[26] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems", *Computers in Industry*, 41(2), 113-127, 2000.

[27] Z. Yan, J. Wang, and G. C. Li, "A collective neurodynamic optimization approach to bound-constrained nonconvex optimization", *Neural Networks*, 55, 20-29, 2014.