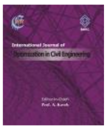
INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2021; 11(2): 231-248



CRACK DETECTION WITH XFEM IN PLATE STRUCTURES USING MDM OPERATOR

A. Kaveh[1*, †], S. R. Hoseini Vaez[2], P. Hosseini[3] and H. Fathi[2]
[1]*School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran*
[2]*Department of Civil Engineering, Faculty of Engineering, University of Qom, Qom*
[3]*Faculty of Engineering, Mahallat Institute of Higher Education, Mahallat, Iran*

ABSTRACT

A modified dolphin monitoring (MDM) is used to augment the efficiency of particle swarm optimization (PSO) and enhanced vibrating particle system (EVPS) for the numerical crack identification problems in plate structures. The extended finite element method (XFEM) is employed for modeling the fracture. The forward problem is untangled by some cycle loading phase via dynamic XFEM. Furthermore, the inverse problem is solved and compared via two PSO and EVPS algorithms. All the problems are also dissolved by means of fine and coarse meshing. The results illustrate that the function of XFEM-PSO-MDM and XFEM-EVPS-MDM is superior to XFEM-PSO and XFEM-EVPS methods. The algorithms coupled via MDM offer a higher convergence rate with more reliable results. The MDM is found to be a suitable tool which can promotes the ability of the algorithms in achieving the optimum solutions.

Keywords: Cack detection; plate structures; modified dolphin monitoring; paticle swarm optimization; enhanced vibrating particle system.

Received: 20 February 2021; Accepted: 25 May 2021

1. INTRODUCTION

Concerning medical imaging, quality detection, and structural health monitoring (SHM), is essential to locate and examine flaws within elastic settings. Cracking is a frequently-occurring flaw, for eliminating which various techniques have been proposed, e.g., electrical

[*] Corresponding author: School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran
[†] E-mail address: alikaveh@iust.ac.ir (A. Kaveh)

tomography, radiographic analysis methods, and ultrasonic tests [1]. Despite the regarding of such techniques in specific conditions and their extensive utilization in the study of structural damage, they are limited to particular flaws and laboratory settings. Furthermore, the use of non-destructive tests (NDTs) with structural responses subjected to dynamic excitation in the absence of a systematic inverse model cannot be a good approach for crack locating, particularly for composites where turbulence is induced by shared surface return waves and the corresponding boundaries. Since these problems are complicated in nature, it is required to adopt reliable computational approaches. In general, it is required to consecutively analyze forward problems by various initial suggestions, e.g., different geometric sites and sizes, in the process of locating cracks until an agreement is found between the analytical and measured quantities. Also, optimization algorithms are utilized to investigate such an agreement.

Researchers have increasingly employed the extended finite element method (XFEM) in recent decades. XFEM is grounded on the partition of unity method (PUM) [2, 3] and was introduced by Belytschko et al. [4, 5]. Also, it was proposed as a substitution for earlier techniques, including the boundary element method (BEM) and the finite element method (FEM). Studies further accepted XFEM as a competent approach to solve forward problems in inverse engineering [6,7]. XFEM provides several advantages. For example, it can model discontinuities such as holes and cracks on any heterogeneous surfaces [8] and does not require re-meshing at each step of analysis in solving forward problems. The latter can add to the capabilities of XFEM in comparison to the classical FEM since this characteristic can bring a significant decline in the computational cost.

Optimal design is an essential application of optimization approaches in knowledge engineering. This allows for properly coping with limitations in engineering [9]. The use of structural optimization produces an economical design with lower material requirements, a shorter computation time, and smaller effort [10]. As a robust approach, researchers employ metaheuristic optimization algorithms for structural purposes. Some of the metaheuristic optimization algorithms include Genetic Algorithms (GAs) [11], Charged System Search (CSS) [12], Ray Optimization (RO) [13, 14], Particle Swarm Optimization (PSO) [15], Colliding Bodies Optimization (CBO) [16], Enhanced Colliding Bodies Optimization (ECBO) [17], Dolphin Echolocation Optimization (DEO) [10, 18], Natural Forest Regeneration (NFR) [19], Tug of War Optimization (TWO) [20], Water Evaporation Optimization (WEO) [21], Simplified Dolphin Echolocation Algorithm (SDEA) [22], Ant Lion Optimizer (ALO) [23]. Some of the above topics can be found in Kaveh [24].

Studies adopted a large number of optimization algorithms to solve inverse problems. Rabinovich et al [6, 7] employed a GA and XFEM to study two-dimensional cracking under both dynamic and static excitation conditions. Waisman et al. [25] performed a performance evaluation on the model proposed by Rabinovich et al. [6,7] concerning elastostatic problems under various damage conditions. They studied a variety of damages, such as cracking, irregularly-shaped holes, and circular holes. Chatzi et al. [26] adopted various GAs with convergence acceleration to detect such flaws by using XFEM. Using multilevel coordinate search (MCS), Nanthakumar et al. [27] performed the flaw detection and measurement of piezoelectric plates by XFEM in order to solve forward problems in all iterations. Sun et al. [28-30] exploited optimization algorithms, e.g., the Broyden-Fletcher-

Goldfarb-Shahno (BFGS) algorithm and a discrete artificial bee colony algorithm, in combination with XFEM to propose several flaw detection techniques. Likewise, Jung et al. [31-33] developed an approach for arbitrary-shaped (i.e., holes, voids, or cracks) scatterer detection and locating within elastic solids with heterogeneity by gradient-based optimization approaches and implicit dynamic XFEM. Nanthakumar et al. [34] employed XFEM and topography optimization to locate holes and cracks. Concerning static and dynamic loading, Zhang et al. [35] adopted the Nelder Mead (NM) [36], quasi-Newton (QN) [37], and dynamic XFEM approaches as local search techniques for the crack tip detection of plates. Livani et al. [38] detected several damages by exploiting active-inactive flaw (AIF) tactics, PSO, and the extended spectral finite element method (XSFEM). Khatir et al. [39] employed extended isogeometric analysis (XIGA) and XFEM coupled with Jaya optimization and PSO to detect cracks of plates. They found XIGA-Jaya to outperform the other approaches. Finally, Fathi et al. [40] presented a new method showed the superiority of the XFEM-EVPS against XFEM-PSO for crack detection in plate structures. They combined a dynamic XFEM with EVPS algorithm for solving an inverse problem. The efficiency and accuracy of the proposed method in recognizing the damage was well presented.

Kaveh and Farhoudi [41] recently introduced the dolphin monitoring (DM) method for convergence control in optimization. Later, Kaveh et al. [42] proposed the modified dolphin monitoring (MDM) for efficient population dispersion control and enhancing algorithms.

This work adopts the MDM operator for the performance improvement of the enhanced vibrating particle system (EVPS) [43] and PSO in detecting damages by the dynamic XFEM. It is a novel approach to adopt the MDM operator in order to enhance algorithm capability as it has not been practiced in detecting damages by XFEM and optimization algorithms. Controlling the convergence speed of algorithm and preventing from being trapped in local optima is performed by MDM. At each stage of algorithm's implementation, the purposefully standard aberration is regulated. As a result, PSO-MDM and EVPS-MDM exhibited improved performance as compared to PSO and EVPS. Furthermore, the dynamic XFEM with cycle loading play an essential role in accelerating the forward problem solving. This is a unique procedure which improves the whole process of damage detection. Ultimately, this study is pioneer for using MDM operator in damage detection, and the best and average results illustrate the efficiency of this method against the common method.

The present work pioneers the utilization of the MDM operator via PSO and EVPS coupled with dynamic XFEM for solving inverse problems. A number of new MDM procedures are employed to improve the ability of DM to obtain global search and avoid local minima trapping. The proposed dynamic XFEM functions in the form of a forward solver within elastic settings. The Newmark-Beta method [44] is exploited for the modeling of the acceleration, velocity, and displacement field. Eventually, the XFEM-EVPS and XFEM-PSO techniques were exploited to perform the efficiency assessment of the XFEM-EVPS-MDM and XFEM-PSO-MDM combined approaches.

2. DYNAMIC XFEM

2.1 Governing equation

Consider a body Ω with an initial traction-free crack in state of dynamic balance, as shown in Fig. 1. One can write the fundamental elasto-dynamic expression as:

$$\nabla.\sigma + f^{\,b} = \rho\ddot{u} \tag{1}$$

The boundary conditions are:

$$\begin{aligned} &u(\mathrm{x},\mathrm{t}) = \bar{u}\ (\mathrm{x},\mathrm{t}) && \text{on } \Gamma_u \\ &\sigma.n = f^{\,t} && \text{on } \Gamma_t \\ &\sigma.n = o && \text{on } \Gamma_c \end{aligned} \tag{2}$$

Furthermore, the initial conditions were met as:

$$\begin{aligned} &u(x\,,\mathrm{t}=0) = \bar{u}(0) \\ &\dot{u}(x\,,\mathrm{t}=0) = \bar{\dot{u}}(0) \end{aligned} \tag{3}$$

in which Γ_u is the displacement boundary, Γ_t is the tension boundary, Γ_c is the crack face boundary, σ denotes the stress tension, $\nabla.$ represents the divergence operator, $f^{\,b}$ stands for the body force, and ρ denotes the mass density. Also, u is the displacement vector, $\dot{u}$ is the velocity vector, $\ddot{u}$ is the acceleration vector, *n* denotes the external unit normal vector, $f^{\,t}$ stands for the tension in Γ_t, and $\bar{u}$ and $\bar{\dot{u}}$ represent the displacement and velocity in Γ_u, respectively.

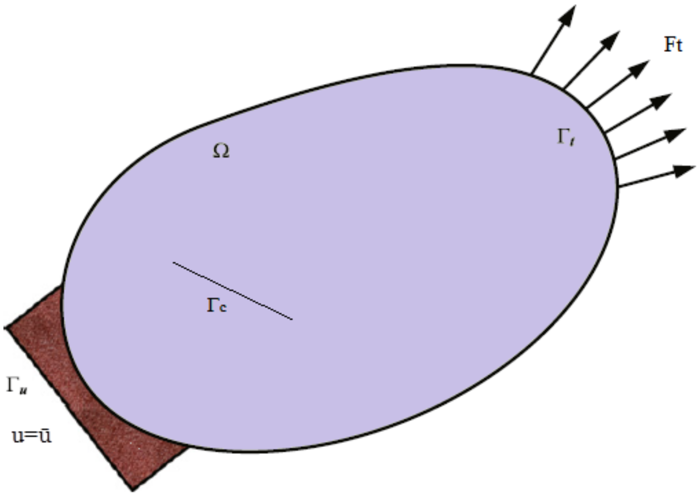


Figure 1. Domain of cracks and boundary conditions [24]

It is possible to represent the weak form of Eq. (1) as:

$$\int_\Omega \rho\ddot{u}\delta u d\Omega + \int_\Omega \sigma.\delta\varepsilon d\Omega = \int_\Omega f^{\,b}\delta u d\Omega + \int_\Gamma f^{\,t}\delta u d\Gamma \tag{4}$$

in which δu is the virtual displacement, and $\delta\varepsilon$ is the virtual strain.

2.2 Discontinuity

2.2.1 XFEM discretization

XFEM enabled the FE mesh-independent representation of discontinuity. Generally, the approximation displacement domain is shown as

$$\begin{aligned} u^h(x) &= u(\mathrm{x}) + \mathrm{u}^H(x) + u^{tip}(\mathrm{x}) \\ &= [\sum_{i\in L} N_i(x)u_i] + [\sum_{j\in L^S} N_j(x)(\mathrm{H(x)} - H(x_j))\mathrm{a}_j] + [\sum_{k=1}^{mt} N_k(x)\left(\sum_{l=1}^{mf}[\psi_l(\mathrm{x}) - \psi_l(x_k)b_k^l\right)] \end{aligned} \tag{5}$$

in which L denotes the set of the entire nodes, L^S is the set of completely crack-cut elements, k is the set of crack tip-restricting nodes, and $\mathrm{N_i(x)}$ stands for the FE shape function of node i. In addition, aj and b_k^l represent additional degrees of freedom (DOF). In Eq. (5) H(x) is the Heaviside enrichment function, proposed by Moës [3]. Also, $\psi_l(\mathrm{x})$ are crack tip enrichment, represented by Fleming et al. [45].

2.2.2 Motion equation

Generally, a differential equation is utilized to define elastodynamic structural behavior as:

$$M\ddot{u} + Ku = f \tag{6}$$

where *M* is the mass matrix, *K* is the stiffness matrix, and *f* denotes the external force vector. The mass matrix, stiffness matrix, and force vector are defined as:

$$M_{ij}^{rs} = \int_{\Omega} (\mathrm{N}_i^r)^T \rho N_j^s d\Omega \ ; \ \ (\mathrm{r,s = u,a,b}) \tag{7}$$

$$K_{ij}^{rs} = \int_{\Omega} (\mathrm{B}_i^r)^T E B_j^s d\Omega \ ; \ (\mathrm{r,s = u,a,b}) \tag{8}$$

$$f_i^u = \int_{\Gamma_t} N_i^T \bar{t} d\Gamma_t + \int_{\Omega} N_i^T b d\Omega \tag{9}$$

where *u* is the usual-enriched node index, *a* is the step-enriched node index, and *b* is the tip-enriched node index.

2.2.3 Time discretization

The present work adopted Newmark's time integral method, or the Newmark-Beta method [44]:

$$\dot{U}_{i+1} = \dot{U}_i \left[(1-\gamma)\Delta t\right]\ddot{U}_i + (\gamma\Delta t)\ddot{U}_{i+1} \tag{10}$$

$$U_{i+1} = U_i + (\Delta t)\dot{U}_i + \left[(0.5-\beta)(\Delta t)^2\right]\ddot{U}_i + [\beta(\Delta t)^2]\ddot{U}_{i+1} \tag{11}$$

where $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ denote variations in acceleration in a time step. They determine the stability and validation of method characteristics.

3. OPTIMIZATION ALGORITHMS

3.1 Modified dolphin monitoring method

Kaveh and Farhoudi proposed DM for convergence rate control [41]. It was later modified in the form of MDM by Kaveh et al. [42] for efficient population dispersion control and algorithm enhancement. MDM not only induces no structural alternations in algorithms but also includes features. MDM should be applied at the loop ends. It is defined as the average locations for each of the variables at a particular radius factor from the standard deviation. This factor equals to 15%. That is, a variable range equals the average value ± 15% of the standard deviation. It is required that the population percentage of a variable in a loop and the corresponding range (i.e., the dispersion index of the available population) equals to:

$$MP_I = 10 + 60[\frac{i-1}{LoopNumber-1}] \tag{12}$$

where MP$_i$ denotes the mandatory dispersion of the population in loop *i*, and the current population dispersion index should remain below MP$_i$ in all the loops. Also, LoopNumber refers to the total number of loops. It is worth noting that MP cannot become equal to 1, and the maximum MP is 0.7. This brings higher flexibility to the algorithm. Also, at an MP of 1, it is necessary to maintain the total population in the aforementioned range. An MP of 0.7, however, keeps a maximum search chance. The main MDM stages include:

1) MP calculation in all loops by Eq. (12);

2) Population calculation in the aforementioned range for all variables in all loops and utilizing it as the dispersion index of the available population;

3a) If the dispersion index of the available population is larger than the mandatory dispersion of the population, the algorithm is found to be running toward optimality (the same range) more rapidly than expected, and it is required to reduce the dispersion index of the population. For this purpose, MDM provides two mechanisms, including

Variable of interest replacement from the in-range population by:

3a.1) Variable of interest from the available out-of-range population

3a.2) Randomly-produced values in the allowed range for all variables. MDM makes use of both mechanisms simultaneously, at a probability of 50%.

3b) For a dispersion index of the available population below the mandatory dispersion, it is found that the algorithm has an unexpectedly slower convergence rate to optimality. In such a case, it is required to raise the dispersion index of the available population. For this purpose, MDM involves likewise two mechanisms, including

Variable of interest replacement from the out-of-range population by:

3b.1) the most desirable optimal variable to the stage.

3b.2) Values in a desirable range. These two mechanisms are likewise employed at a probability of 50% simultaneously. The former mechanism prevents the distancing of the algorithm from the best solution and optimizes the range up to that point while continuing seeking the obtained optimal solution up to the point.

3.2 Particle swarm optimization

PSO is based on flocking behavior among birds. It was introduced by Kennedy [46] as a general optimization approach and is employed to solve optimization problems. A large number of independent particles (entities) are generated randomly within the search space where each particle represents a solution and suggest a suitable location via a velocity. A swarm contains N particles moving in a d-dimensional search space, and the memory of each particle facilitates the remembering of the best previous position. Neighborhood *i* refers to a set of particles that have topographical connections with particle *i*. It may comprise the entire or some of the population. In order to detect other particles that affect individuals, several topologies are exploited. Also, initialization is performed randomly for every individual. In each stage of exploration, particles are updated by using two values, the first of which is $\vec{P_i}$ that belongs to the best ever-experienced position, which is known as personal best. The second value, on the other hand, is $\overrightarrow{\mathrm{P_g}}$ as the best ever-achieved position within the population, which is known as the global best. Once these two values have been identified, the position $\overrightarrow{\mathrm{X_1}}$ and velocity $\overrightarrow{\mathrm{V_1}}$ of the entity, which represent the particle location and direction, respectively, undergo an improvement:

$$V_{id}(t+1)=\chi(V_{id}(t)+C_1r_1(P_{id}(t)-X_{id}(t))+C_2r_2(P_{gd}(t)-X_{id}(t))) \tag{13}$$

$$X_{id}(t+1)=X_{id}(t)+V_{id}(t+1) \tag{14}$$

in which *i=1, 2, ...,* N, N represents the size of the swarm, r_1 and r_2 are random uniformly-distributed numbers ranging from 0 to 1. Also, C_1 and C_2 are acceleration factors representing entity attraction toward the success of the same entity and the neighbors, respectively. As the constriction factor, χ is calculated as [47]:

$$\chi=\frac{2}{\varphi-2+\sqrt{\varphi^2-4\varphi}} \tag{15}$$

where φ=2.05, and $\varphi = \varphi_1 + \varphi_2 > 4$. Fig. 2 demonstrates the general architecture of PSO.

3.3 Enhanced vibrating particles system

As a metaheuristic optimization algorithm, the vibrating particle system (VPS) [48] was developed based on vibrating systems with one DEF and viscose damping. Also, the enhanced VPS (EVPS) [43] makes use of a number of processes contributing to convergence rate enhancement, global search capability enhancement of standard VPS, and local optimal avoidance. The production of initial factors in the permissible range is performed as:

$$x_i^j = x_{\min} + rand.(\mathrm{x}_{\max}-\mathrm{x}_{\min}) \tag{16}$$

in which x_i^j represents variable *j* of particle *i*, x_{min} stands for the starting point of the permissible search space for variable *i*, x_{max} is the ending point of the permissible search space for variable *i*, and *rand* refers to a random quantity ranging from 0 to 1. EVPS involves:

(i) HB (historically best position within the population): It is the best candidate until the corresponding iteration.

(ii) GP (a good particle): It is randomly chosen among partially best solutions in every iteration.

(iii) BP (a bad particle): It is randomly chosen among partially worst solutions in every iteration.

(iv) OHB (one of the historically best positions within the population): It is a randomly-chosen row of the "memory."

Memory in EVPS serves as the HB and saves the memorysize number of the GB from the population. The best solution that is better than the worst *Memory* value in iteration replaces it within the memory. A descending function is defined as:

$$D = (\frac{iter}{iter_{\max}})^{-\alpha} \tag{17}$$

in which *iter* represents the number of the current iteration, $iter_{max}$ denotes the total number of iterations, and α is a constant. The generation of the next agents is updated at a probability of $\omega_{1,2}$ and ω_3 by one of:

$$x_i^j = \begin{cases} [\mathrm{D.A.rand1} + \mathrm{OHB}^j] & \text{(a)} \\ [\mathrm{D.A.rand}\,2 + GP^j] & \text{(b)} \\ [\mathrm{D.A.rand}\,3 + BP^j] & \text{(c)} \end{cases}$$

$$A = \begin{cases} (\pm 1)(\mathrm{OHB}^j - x_i^j) & \text{(a)} \\ (\pm 1)(\mathrm{GP}^j - x_i^j) & \text{(b)} \\ (\pm 1)(\mathrm{BP}^j - x_i^j) & \text{(c)} \end{cases} \tag{18}$$

$$\omega_1 + \omega_2 + \omega_3 = 1$$

where $\omega_{1,2}$ and ω_3 have relative importance to GB, GP, and BP, while *rand1*, *rand2*, and *rand3* are random uniformly-disseminated numbers within the range of [0, 1]. In addition, +1 and -1 are randomly utilized, and it is worth mentioning that BP, GP, and OHB are autonomously determined for all the agents.

4. NUMERICAL PROBLEMS

The proposed methods are compared in two numerical examples in terms of effective crack coordinate detection. The examples were a 1×1 plate. Also, a plane-stress setting was assumed. The modulus of elasticity was set to 2e11 N/m, while Poisson's ratio was selected to be 0.3. According to Fig. 3, the top plate edge was subjected to uniform dynamic traction loading. Furthermore, two joint supports hinged the bottom edge at the rollers and corners between the joints. To calculate the displacement in a forward XFEM analysis, it is required to uniformly arrange sensors along the edge range (excluding the bottom edge). There are fourteen nodes in the edge space in the sensors. The two-dimensional displacements of the nodes are measured in each of the loading phases, being recorded in the A-Matrix. Then, the A-Matrix is treated to be the plate node displacement pattern under cracking after the completion of loading. The entire problems are addressed by a consistent series of 20×20 and 60×60 uniform structural meshes.

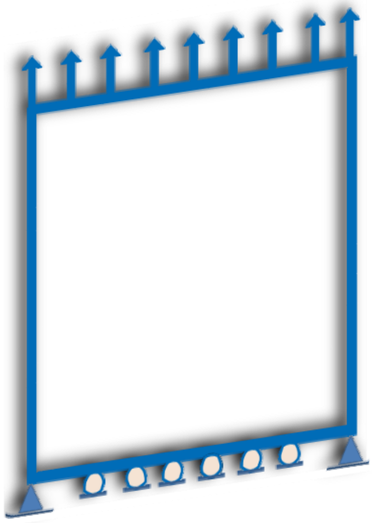


Figure 3. Traction-loaded plate

The top plate edge is subjected to an impulsive load of P=3×10^4 in 0.4s. The alternating loading cycle has a time step of 0.05s. The maximum load occurs by a rising step of 0.25 at 0.2s, before reducing at the same step to zero at 0.4s. Such XFEM dynamic loading is of high effectiveness and accelerates the solving of forward problems.

Additionally, to solve inverse problems in an optimization algorithm, a maximum of 500 iterations are applied, while the random population is 30. Twenty repetitions are performed on each of the optimization problems to prevent statistical simulation uncertainties. Ultimately, the optimal solution with the smallest objective function quantity of the two approaches is selected to represent the responses.

To solve the optimization problems, the present work employed an objective function defined as [49]:

$$F(\mathrm{U}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\mathrm{A}_i^a(\mathrm{U}) - \mathrm{A}_i^c(\mathrm{U}))^2} \tag{19}$$

in which U denotes the loading-induced displacement under cracking, n is the loading step, A_i^a is the matrix i of measured displacements, and A_i^c is the matrix i of the computed displacements.

4.1 XFEM-PSO-MDM versus XFEM-PSO

In order to perform the performance evaluation of XFEM-PSO-MDM and XFEM-PSO, it is required to identify the crack parameters (x1, y1) = (0, 0.25) and (x2, y2) = (0.4, 0.5) under tensile loading. Fig. 4 illustrates the crack and plate. It should be noted that (xi, yi) ϵ [0, 1], i= 1, 2.

Such problems are analyzed to evaluate the dynamic XFEM and proposed approaches in terms of convergence. XFEM-PSO-MDM showed higher performance than XFEM-PSO as it could carry out more rapid crack detection under both mesh sizes. Fig. 6 shows the real crack versus the detected ones of the fine and coarse meshes. Tables 1 and 2 provide the inverse analysis results. In these tables, the proposed approaches found the locations in the optimal scenario of both mesh sizes. The XFEM-PSO-MDM results were observed to converge to the real quantities within 18 and 31 iterations for 20×20 and 60×60 meshes in the in best situation, respectively. Furthermore, the XFEM-PSO results converged to the real quantities by 55 iterations under 20×20 meshes and by 33 iterations under 60×60 meshes. As can be inferred, XFEM-PSO was not close to XFEM-PSO-MDM in terms of the convergence rate. Also, Fig. 5 demonstrates the convergence histories of the problem for the two mesh sizes.

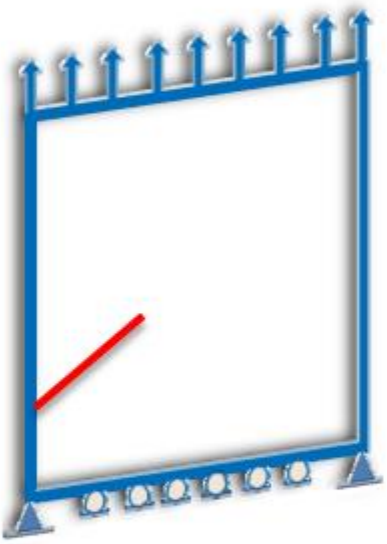


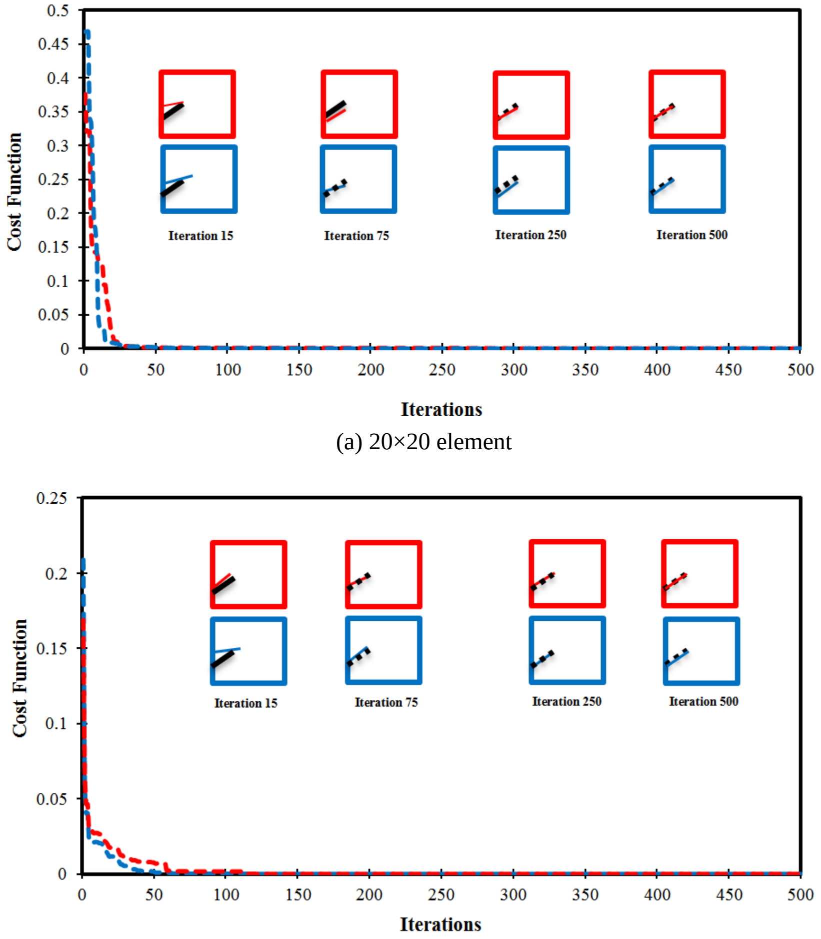
Figure 4. Crack within the plate

Table 1: Crack locating results for 20×20 meshes XFEM-PSO-MDM & XFEMPSO

Result		Nodes and damage extents				Result		Nodes and damage extents			
		x_1	y_1	x_2	y_2			x_1	y_1	x_2	y_2
XFEM-PSO-MDM	Actual	0	0.250	0.400	0.500	XFEM-PSO	Actual	0	0.250	0.400	0.500
	Mean	0.197183	0.339806	0.240002	0.38561		Mean	0.143718	0.316212	0.285739	0.414243
	Best	0	0.25	0.400003	0.500001		Best	0	0.25	0.400003	0.500001

Table 2: Crack locating results for 60×60 meshes XFEM-PSO-MDM & XFEMPSO

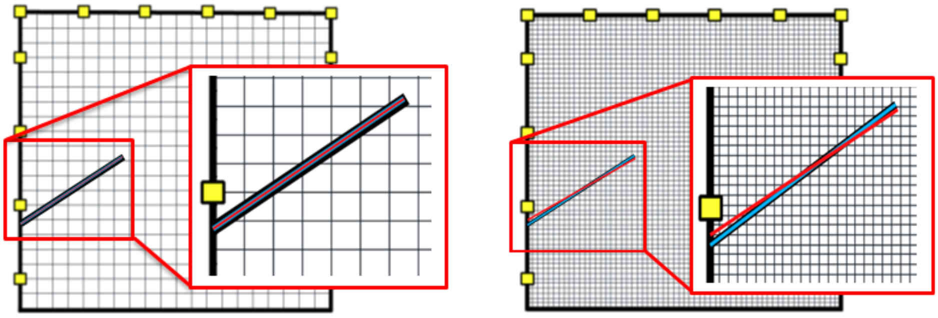
Result		Nodes and damage extents				Result		Nodes and damage extents			
		x_1	y_1	x_2	y_2			x_1	y_1	x_2	y_2
XFEM-PSO-MDM	Actual	0	0.250	0.400	0.500	XFEM-PSO	Actual	0	0.250	0.400	0.500
	Mean	0.140684	0.325343	0.266668	0.666648		Mean	0.188083	0.340261	0.250012	0.366637
	Best	0	0.25	0.400003	0.500001		Best	0	0.25013	0.400009	0.499999



(a) 20×20 element

(b) 60×60 element

Figure 5. Convergence histories for (a) 20×20 meshes, (b) 60×60 meshes (the blue lines represent the XFEM-PSO-MDM results, while the red lines stand for the XFEM-PSO results)



(a) 20×20 meshing (b) 60×60 meshing

Figure 6. Crack locating results for (a) Fine and (b) Coarse meshes. Black line: Real crack; Blue line: Identified crack with XFEM-PSO-MDM; Red line: Identified crack with XFEM-PSO

4.2 XFEM-EVPS-MDM versus XFEM-EVPS

XFEM-EVPS-MDM and XFEM-EVPS were evaluated in terms of internal crack locating capability. The actual defect parameters included (x_1, y_1) = (0.7, 0.5) and (x_2, y_2) = (0.8, 0.602). Fig. 7 depicts the internal inclined crack in the plate. Likewise, XFEM-EVPS-MDM was observed to outperform XFEM-EVPS since it could perform an accurate detection of cracks under both mesh sizes. Fig. 9 compares the actual and detected cracks at two mesh sizes. In addition, Tables 3 and 4 provide the inverse analysis results of the fine and coarse mesh sizes, respectively. The XFEM-EVPS-MDM results for the best coarse gridding scenario converged to the real quantity in 5 and 15 iterations for the coarse and fine meshes sizes in the best condition, respectively. On the other hand, the XFEM-EVPS results converged to the real quantity in 42 and 63. As can be seen, XFEM-EVPS-MDM had a larger convergence rate under fine and coarse meshes. Fig. 8 represents the convergence histories at the two mesh sizes.

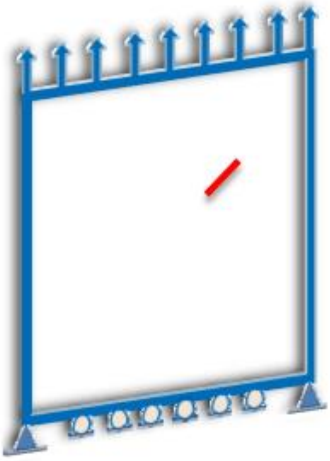


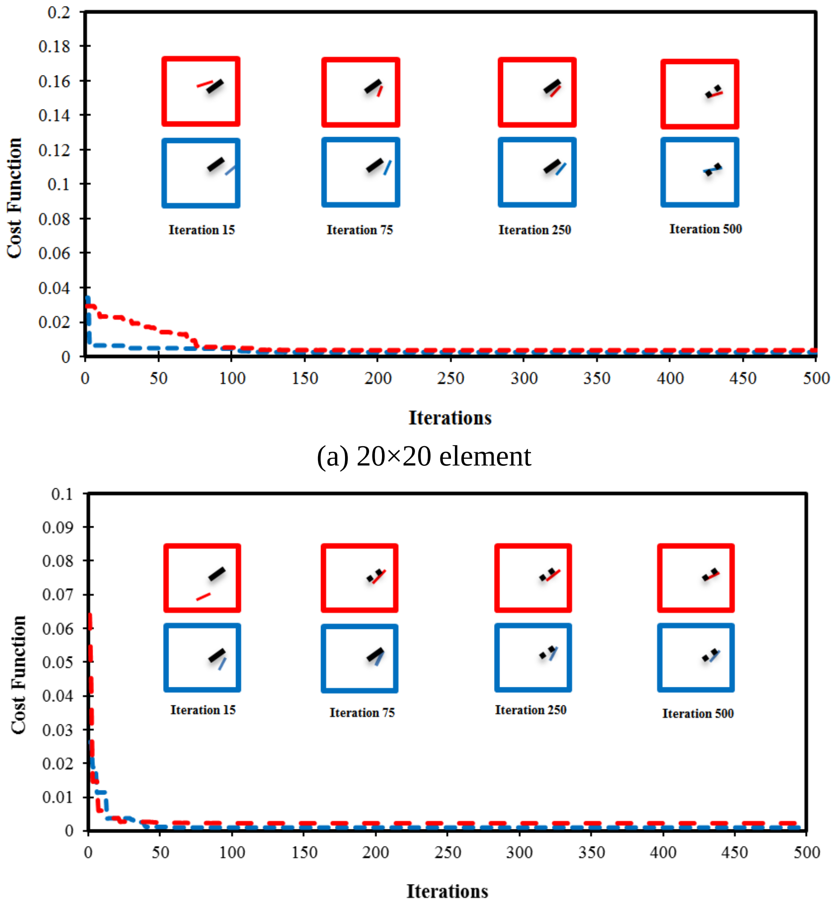
Figure 7. Crack within the plate

Table 3: Crack locating results for 20×20 meshes XFEM-EVPS-MDM & XFEM-EVPS

Result		Nodes and damage extents				Result		Nodes and damage extents			
		x_1	y_1	x_2	y_2			x_1	y_1	x_2	y_2
XFEM-EVPS-MDM	Actual	0.700	0.500	0.800	0.602	XFEM-EVPS	Actual	0.700	0.500	0.800	0.602
	Mean	0.685117	0.280446	0.699223	0.28828		Mean	0.763375	0.456309	0.7085	0.392571
	Best	0.785337	0.577892	0.803233	0.449297		Best	0.746949	0.556516	0.815109	0.461497

Table 4: Crack locating results for 60×60 meshes XFEM-EVPS-MDM & XFEM-EVPS

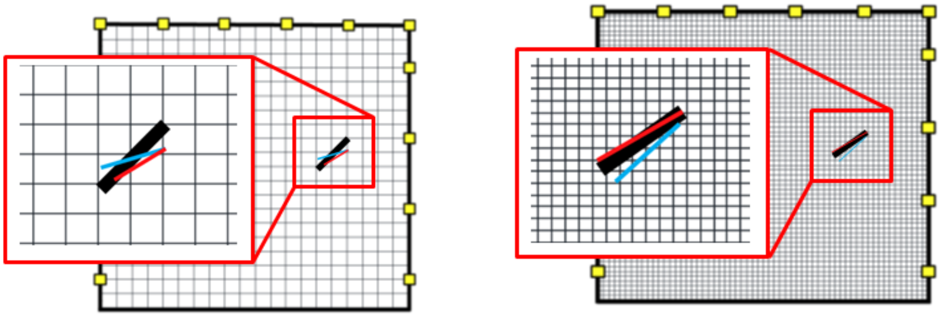
Result		Nodes and damage extents				Result		Nodes and damage extents			
		x_1	y_1	x_2	y_2			x_1	y_1	x_2	y_2
XFEM-EVPS-MDM	Actual	0.700	0.500	0.800	0.602	XFEM-EVPS	Actual	0.700	0.500	0.800	0.602
	Mean	0.685385	0.618349	0.710251	0.609488		Mean	0.769643	0.550228	0.747001	0.506383
	Best	0.729311	0.455981	0.845812	0.538678		Best	0.707735	0.502841	0.806927	0.605468



(a) 20×20 element

(b) 60×60 element

Figure 8. Convergence histories for (a) 20×20 meshes, (b) 60×60 meshes, (the blue lines represent the XFEM-EVPS-MDM results, while the red lines stand for the XFEM-EVPS results)



(a) 20×20 meshing (b) 60×60 meshing

Figure 9. Crack locating results for (a) 20×20 meshes and (b) 60×60 meshes. Black line: Real crack; Blue line: Identified crack with XFEM-EVPS-MDM; Red line: Identified crack with XFEM-EVPS

5. CONCLUSIONS

The integration of optimization algorithm and the MDM operator yielded a proper numerical crack detection technique in plates. It is worth noting that MDM results in no alternations in the main optimization phases and solely functions as an operator in ultimate solutions of loops to improve the optimization algorithm in behavior. The proposed technique was employed to solve a time-limited forward problem by loading cycles via dynamic XFEM. Furthermore, PSO, EVPS, and the combined algorithms of PSO-MDM and EVPS-MDM were employed to solve an inverse problem. It was observed that MDM improved the performance of PSO and EVPS in crack tip coordinate detection. PSO-MDM and EVPS-MDM exhibited larger convergence rates and higher reliability in comparison to PSO and EVPS. Eventually, the findings revealed that the MDM-coupled approaches outperformed the individual optimization algorithms, and this technique can seemingly enhance the capability of these algorithms in finding optimal solutions.

REFERENCES

1. Hellier C, Shakinovsky M. *Handbook of Nondestructive Evaluation* 2001; **10**, Mcgraw-hill New York.
2. Melenk JM, Babuška I. The partition of unity finite element method: basic theory and applications, *Comput Meth Appl Mech Eng* 1996; **139**(1-4): 289-314.
3. Duarte CA, Babuška I, Oden JT. Generalized finite element methods for three-dimensional structural mechanics problems, *Comput Struct* 2000; **77**(2): 215-32.
4. Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing, *Int J Numer Meth Eng* 1999; **45**(5): 601-20.
5. Moës N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing, *Int J Numer Meth Eng* 1999; **46**(1): 131-50.

6. Rabinovich D, Givoli D, Vigdergauz S. XFEM-based crack detection scheme using a genetic algorithm, *Int J Numer Meth Eng* 2007; **71**(9): 1051-80.
7. Rabinovich D, Givoli D, Vigdergauz S. Crack identification by 'arrival time'using XFEM and a genetic algorithm, *Int J Numer Meth Eng* 2009; **77**(3): 337-59.
8. Sukumar N, Huang ZY, Prévost JH, Suo Z. Partition of unity enrichment for bimaterial interface cracks, *Int J Numer Meth Eng* 2004; **59**(8): 1075-1102.
9. Kaveh A, Hosseini P. A simplified dolphin echolocation optimization method for optimum design of trusses, *Int J Optim Civil Eng* 2014; **4**(3): 381-97.
10. Kaveh A, Farhoudi N. A new optimization method: Dolphin echolocation, *Adv Eng Softw* 2013; **59**: 53-70.
11. Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, University of Michigan Press, 1975.
12. Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search, *Acta Mech* 2010; **213**(3-4): 267-89.
13. Kaveh A, Khayatazad M. A new meta-heuristic method: Ray Optimization, *Comput Struct* 2012; **112–113**: 283-94.
14. Kaveh A, Ilchi Ghazaan M, Bakhshpoori T. An improved ray optimization algorithm for design of truss structures, *Period Polytech Civil Eng* 2013; **57**(2): 97–112.
15. Eberhart R, Kennedy J. A new optimizer using particles warm theory, *In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995, Oct. 4-6. pp. 39-45.
16. Kaveh A, Mahdavi VR. Colliding bodies optimization: a novel meta- heuristic method, *Comput Struct* 2014; **139**: 18–27.
17. Kaveh A, Ilchi Ghazaan M. Enhanced colliding bodies optimization for design problems with continuous and discrete variables, *Adv Eng Softw* 2014; **77**: 66-75.
18. Kaveh A, Farhoudi N. Layout optimization of braced frames using differential evolution algorithm and dolphin echolocation optimization, *Period Ploytech Civil Eng* 2015; **59**(3): 441-9.
19. Moez H, Kaveh A, Taghizadieh N. Natural forest regeneration algorithm for optimum design of truss structures with continuous and discrete variables, *Period Polytech Civil Eng* 2016; **60**(2): 257–67.
20. Kaveh A, Zolghadr A. A novel meta-heuristic algorithm: tug of war optimization, *Int J Optim Civil Eng* 2016; **6**(4): 469–92.
21. Kaveh A, Bakhshpoori T. Water evaporation optimization: A novel physically inspired optimization algorithm, *Comput Struct* 2016; **167**: 69–85.
22. Kaveh A, Hoseini Vaez SR, Hosseini P, Fallah N. Detection of damage in truss structures using Simplified Dolphin Echolocation algorithm based on modal data, *Smart Struct Syst* 2016; **18**(5): 983–1004.
23. Mirjalili S. The antlion optimizer, *Adv Eng Softw* 2015; **83**: 80–98.
24. Kaveh A. *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, 3rd edition Springer, Switzerland, 2021.

25. Waisman H, Chatzi E, Smyth AW. Detection and quantification of flaws in structures by the extended finite element method and genetic algorithms, *Int J Numer Meth Eng* 2010; **82**(3): 303-28.
26. Chatzi EN N, Hiriyur B, Waisman H, Smyth AW. Experimental application and enhancement of the XFEM–GA algorithm for the detection of flaws in structures, *Comput Struct* 2011; **89**(7-8): 556-70.
27. Nanthakumar S, Lahmer T, Rabczuk T. Detection of flaws in piezoelectric structures using extended FEM, *Int J Numer Meth Eng* 2013; **96**(6): 373-89.
28. Sun H, Waisman H, Betti R. Nondestructive identification of multiple flaws using XFEM and a topologically adapting artificial bee colony algorithm, *Int J Numer Meth Eng* 2013; **95**(10): 871-900.
29. Sun H, Waisman H, Betti R. A multiscale flaw detection algorithm based on XFEM, *Int J Numer Meth Eng* 2014; **100**(7): 477-503.
30. Sun H, Waisman H, Betti R. A two-scale algorithm for detection of multiple flaws in structures modeled with XFEM, in nondestructive characterization for composite materials, aerospace engineering, civil infrastructure, and homeland security, *International Society for Optics and Photonics*, 2014.
31. Jung J, Jeong C, Taciroglu E. Identification of a scatterer embedded in elastic heterogeneous media using dynamic XFEM, *Comput Meth Appl Mech Eng* 2013; **259**: 50-63.
32. Jung J, Taciroglu E. Modeling and identification of an arbitrarily shaped scatterer using dynamic XFEM with cubic splines, *Comput Meth Appl Mech Eng* 2014; **278**: 101-18.
33. Jung J, Taciroglu E. A divide-alternate-and-conquer approach for localization and shape identification of multiple scatterers in heterogeneous media using dynamic XFEM, *Inverse Probl Imag* **2016**; **10**(1): 165-93.
34. Nanthakumar S, Lahmer T, Rabczuk T. Detection of multiple flaws in piezoelectric structures using XFEM and level sets, *Comput Meth Appl Mech Eng* 2014; **275**: 98-112.
35. Zhang C, Wang C, Lahmer T, He P, Rabczuk T. A dynamic XFEM formulation for crack identification, *Int J Mech Mater Des* 2016; **12**(4): 427-48.
36. Kelley CT. *Iterative Methods for Optimization* 1999; **18**, Siam.
37. Nelder JA, Mead R. A simplex method for function minimization, *The Comput J* 1965; **7**(4): 308-13.
38. Livani M, Khaji N, Zakian P. Identification of multiple flaws in 2D structures using dynamic extended spectral finite element method with a universally enhanced meta-heuristic optimizer, *Struct Multidisc Optim* 2018; **57**(2): 605-23.
39. Khatir Samir, and Magd Abdel Wahab. A computational approach for crack identification in plate structures using XFEM, XIGA, PSO and Jaya algorithm, *Theoret Appl Fract Mech* 2019; **103**: 102240.
40. Fathi H, Hoseini Vaez SR, Zhang Q, Alavi AH. A new approach for crack detection in plate structures using an integrated extended finite element and enhanced vibrating particles system optimization methods, *Struct* 2021; **29**: 638-651.
41. Kaveh A, Farhoudi N. A new optimization method: Dolphin echolocation, *Adv Eng Softw* 2013; **59**: 53-70.

42. Kaveh A, Vaez SH, Hosseini P. Modified dolphin monitoring operator for weight optimization of frame structures, *Period Polytech Civil Eng* 2017; **61**(4): 770-9.
43. Kaveh A, Hoseini Vaez SR, Hosseini P. MATLAB code for an enhanced vibrating particles system algorithm, *Int J Optim in Civil Eng* 2018; **8**(3): 401-14.
44. Newmark NM. A method of computation for structural dynamics, *J Eng Mech Div* 1959; **85**(3): 67-94.
45. Fleming M, Chu YA, Moran B, Belytschko T. Enriched element-free Galerkin methods for crack tip fields, *Int J Numer Meth Eng* 1997; **40**(8): 1483-1504.
46. Kennedy J. *Particle Swarm Optimization,* In *Encyclopedia of Machine Learning,* 2011, Springer, pp. 760-766.
47. Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *IEEE Transact Evolut Comput* 2002; **6**(1): 58-73.
48. Kaveh A, Ilchi Ghazaan M. Vibrating particles system algorithm for truss optimization with multiple natural frequency constraints, *Acta Mech* 2017; **228.1**: 307-22.
49. Kaveh A, Zolghadr A. An improved CSS for damage detection of truss structures using changes in natural frequencies and mode shapes, *Adv Eng Softw* 2015; **80**: 93-100.