







the next flying cycle (i.e. iteration) to guide their flight. Additionally, the updates to the repository are performed considering a geographically-based adaptive grids to preserve the diversity of solutions (i.e., diversity of objective function values associated with each particle).

Pareto dominance is a concept implemented in optimization practices to determine whether a certain condition or set of conditions is more desirable in current than in the previous iteration. If the current set of conditions does not worsen any of the outcomes and also improves at least one of the outcomes, then it is said to dominate the previous conditions. This same process leads to the creation of a Pareto front, a set of solutions to an optimization problem where no individual solution is entirely better than any other of the solutions in the set [18]. In MOPSO, nondominated outcomes of each iteration are stored in the repository and are compared to subsequent iterations' outcomes [13]. Because MOPSO is a process meant to optimize several interrelated, and often conflicting, variables, the concept of Pareto dominance is very important.

Although there are many ways to adapt MOPSO [18], the logic of the algorithm typically includes the following elements:

1. Problem definition:

- a. Define input (decision) variables.
- b. Define the size of the search space (upper and lower bounds for the input variables).
- c. Define objective (cost) functions; costs must be the function of input variables.

2. MOPSO input parameters selection:

- a. Set the number of particles in the swarm (i.e., population size),  $nPop$ .
- b. Define the repository size (i.e., number of non-dominated particles that can be stored),  $nRep$ .
- c. Set the maximum number of flying cycles (i.e., number of iterations),  $nIter$ .
- d. Set parameters that define the criteria for the selection of the new position. These include: inertia weight ( $w$ ), which keeps particles from traveling too far from one iteration to the next; personal learning coefficient ( $c_1$ ), which determines how each particle's velocity will be affected by its position; and global learning coefficient ( $c_2$ ), which determines how each particle's velocity will be affected by the position of the leader.

3. Problem initialization:

- a. Initialize the population by arranging the particles randomly within a search space. The particle  $i$  is defined with a position vector  $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ , where the dimension,  $D$ , of the position vector represents the total number of input variables (i.e., decision variables).
- b. Initialize the speed of each particle by setting them to zero,  $v_i = (v_{i1}, v_{i2}, \dots, v_{iD}) = 0$ .
- c. Evaluate the costs (objective function values) for each of the particles in the swarm.
- d. Initialize repository and store the positions of particles that represent nondominated vector solutions in the repository.
- e. Generate hypercubes (or grids) of the search space and locate particles using these hypercubes as a coordinate system where each particle's coordinates are defined based on their costs (objective function values).
- f. For each particle initialize the best personal position,  $Pbest_i$ , by assigning the current position,  $x_i$ , to it (i.e.,  $Pbest_i = x_i$ ). This info will be stored in the repository and used to guide the travel of particles through the search space.

4. MOPSO main loop: For each iteration (i.e., flying cycle) and each particle in the swarm do the following:











The last MOPSO parameter explored herein is the number of iterations (i.e., flying cycles),  $nIter$ . It is altered to take the following values: 1, 3, and 5. This test uses knowledge from the two prior tests; therefore  $nGrid$  is set to 15 and  $nPop$  is set to 30. Fig. 6 shows that each explored case provides a diverse decision variable outcome. However, as the cost functions get optimized with each iteration, the results show the migration of non-dominated particles in repositories towards the upper-right corner of the search space. In this study, 5 iterations achieved satisfactory MOPSO solutions and are used in the subsequent sections of this paper.

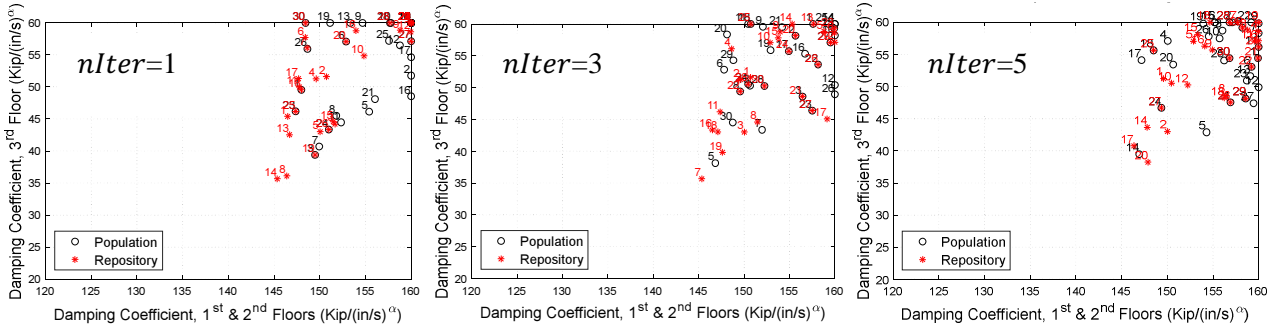


Fig. 6 - MOPSO decision variable outcomes for different number of considered iterations,  $nIter$  ( $nPop=30$ ,  $nGrid=15$ )

### 3.4 Comparison of solution outcomes when “maximum” or “average” of an EDP is used as a cost function

Building on the results of sensitivity studies presented in Sections 3.2 and 3.3, this section shows the effect of the choice of response measure (“maximum” or “average”) in the cost function on the MOPSO solution set. Meanings of “maximum” and “average” responses are next explained with an example. For instance, “maximum” ISD is the maximum of peak ISDs from all stories, where peak ISD of one story represents maximum ISD (in the absolute sense) attained at that story during the considered earthquake. Similarly, “average” ISD is the average of peak ISDs.

Fig. 7 shows decision variable outcomes across the search space for the two considered choices of response measure in the cost function, “maximum” (left figure) and “average” (right figure). In both cases, the decision variable outcomes fall within the same area of the search space following a similar pattern; however, the outcomes associated with “maximum” response measures are less dense. Furthermore, Fig. 8 compares Pareto fronts of FAs and ISDs for the two choices of response measures by illustrating “minimum”, “average”, and “maximum” response for each particle in the repository. Both of the choices result in the same pattern of Pareto fronts, where the choice of the “maximum” response measure in cost function generates a wider range of solutions as a result of a higher diversity of decision variable outcomes.

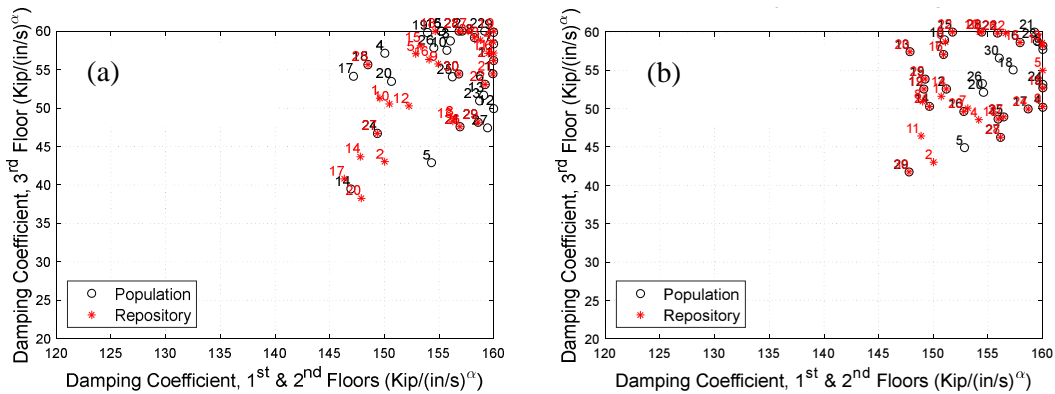


Fig. 7 - MOPSO decision variable outcomes when: (a) “maximum” of an EDP is used as the cost function, (b) “average” of an EDP is used as the cost function.







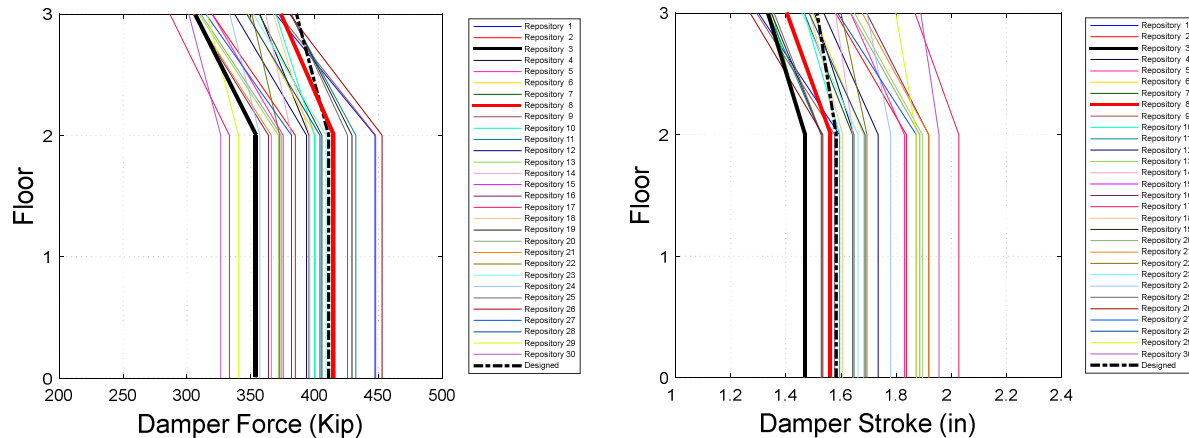


Fig. 11 - Dampers' forces and strokes for MOPSO (original VDMF design, black dashed line; repository 2, tick red line).

## 5. Summary and Conclusions

This study focuses on the optimization of viscous dampers for seismic applications utilizing multi-objective particle swarm optimization (MOPSO) algorithm. MOPSO with its inherent metaheuristic approach and geographically-based adaptive grids avoids getting trapped in the local optimal solutions and effectively discovers diverse non-convex solutions. To further improve the efficiency of the search considering an engineering application, MOPSO is expended by applying constraints on objective functions and by allowing for parallel computing.

The study provides recommendations on how to adequately use MOPSO to generate reliable solution set is seismic applications. Special emphasis is placed on the definition of objective functions, set up of MOPSO input parameters, and parallel computing utilizing high-performance computing resources. The presented study reveals that cost functions that only contain EDPs generate locally optimized solutions, while consideration of EDPs along with dampers' forces improves the solution set. Furthermore, the study highlights the importance of proper set up of MOPSO parameters (density of the search space, population size, and the number of iterations) and demonstrates their effect on the solution outcomes through sensitivity studies. The proposed MOPSO adoption for design of VDMF is verified through a comparison of the MOPSO solution set with an engineered design solution. It was found out that the MOPSO solution set contains outcomes that reduce damper demands (force and stroke) by 10-20% relative to the engineered solution, while maintaining EDP responses similar. This translates into construction savings as a result of reduced manufacturing cost of dampers. Additionally, MOPSO solution set contains the outcomes that reduce "maximum" ISDs by 15%, while generating similar demands on dampers as the engineered solution. This implies better seismic performance at the same construction cost, demonstrating the power and efficiency of MOPSO in seismic applications.

## 6. Acknowledgments

This work was supported by the National Science Foundation, Award No. CMMI-1563428. Any opinions, findings, and conclusions expressed herein are those of the authors and do not necessarily reflect those of the sponsors.

## 7. References

- [1] Terzic V, Mahin SA (2017): Using PBEE to Assess and Improve Performance of Different Structural Systems for Low-rise Steel Buildings. *International Journal of Safety and Security Engineering*, 7(4), 532-544.



- [2] Wang S, Mahin SA (2018): Seismic Upgrade of an Existing Tall Building Using Different Supplemental Energy Dissipation Devices. *Journal of Structural Engineering*, **144**(7), 04018091.
- [3] Del Gobbo GM, Williams MS, Blakeborough A (2018): Comparing fluid viscous damper placement methods considering total-building seismic performance. *Earthquake Engineering & Structural Dynamics*, **47**(14), 2864–2886.
- [4] Terzic V, Mahin SA, Comerio MC (2014): Comparative life-cycle cost and performance analysis of structural systems for buildings. *Proceedings of the 10th National Conference in Earthquake Engineering*, Earthquake Engineering Research Institute, Anchorage, AK.
- [5] Holland JH (1975): *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan; re-issued by MIT Press (1992).
- [6] Kennedy J, Eberhart R (1995): Particle swarm optimization. *Proc. IEEE International Conference on Neural Networks*, Perth, Australia.
- [7] Knowles JD, Corne DW (2000): Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary computation*, **8**(2), 149–172.
- [8] Corne DW, Jerram NR, Knowles JD, Oates MJ (2001): PESA-II: Region-based selection in evolutionary multiobjective optimization. *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, Morgan Kaufmann Publishers Inc.
- [9] Fonseca CM, Fleming PJ (1993): Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. *Proceedings of the 5th International Conference on Genetic Algorithms*, San Mateo, CA.
- [10] Deb K, Jain H (2013): An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE transactions on evolutionary computation*, **18**(4), 577–601.
- [11] Vedarajan G, Chan LC, Goldberg DE (1997): Investment portfolio optimization using genetic algorithms. *Late breaking papers at the genetic programming conference*, Stanford University, California: Stanford Bookstore.
- [12] Zhang Q, Li H (2007): MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, **11**(6), 712–731.
- [13] Coello Coello C, Lechuga MS (2002): MOPSO: A proposal for multiple objective particle swarm optimization. *Proceedings of the Congress on Evolutionary Computation*, IEEE.
- [14] Castaldo P, De Iuliis M (2014): Optimal integrated seismic design of structural and viscoelastic bracing-damper systems. *Earthquake Engineering & Structural Dynamics*, **43**(12), 1809–1827.
- [15] Pollini N, Lavan O, Amir O (2016): Towards realistic minimum-cost optimization of viscous fluid dampers for seismic retrofitting. *Bulletin of Earthquake Engineering*, **14**(3), 971–998.
- [16] Altieri D, Tubaldi E, De Angelis M, Patelli E, Dall’Asta A (2018): Reliability-based optimal design of nonlinear viscous dampers for the seismic protection of structural systems. *Bulletin of Earthquake Engineering*, **16**(2), 963–982.
- [17] Wang S, Mahin SA (2018): High-performance computer-aided optimization of viscous dampers for improving the seismic performance of a tall building. *Soil Dynamics and Earthquake Engineering*, **113**, 454–461.
- [18] Reyes-Sierra M, Coello Coello C (2006): Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*, **2**(3), 287–308.
- [19] Clerc M, Kennedy J (2002): The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, **3**(1), 58–73.
- [20] Yarpiz, <https://yarpiz.com/59/ypea121-mopso>
- [21] McKenna F, Fenves GL (2004): Open System for Earthquake Engineering Simulation (OpenSees). *PEER*, University of California, Berkeley, CA.
- [22] Baei M (2019): Optimization of dampers in viscously damped moment frames utilizing MOPSO. *Master Thesis*, California State University, Long Beach.
- [23] Thomas M (2018): Computational efficiency of multi-objective particle swarm optimization considering a single degree of freedom system under earthquake loading. *Thesis*, California State University, Long Beach.