

An enhanced cooperation between multi-search strategy and parallel differential evolution for optimal design problems

*Tan-Tien Nguyen¹⁾, The-Viet Ha²⁾, Quoc-Hung Nguyen³⁾

^{1), 2), 3)} Faculty of Engineering, Vietnamese-German University, Vietnam

¹⁾ tien.nt@vgu.edu.vn

ABSTRACT

The increasing complexity of modern structural design problems requires optimization algorithms to have an acceptable completion time regarding the huge number of design variables. This paper proposes a parallel differential evolution with cooperative multi-search strategies (PDECMS) and the implementation with Compute Unified Device Architecture (CUDA) for improving execution time by leveraging the Graphical Processing Unit (GPU). Three sub-populations with dedicated mutation schemes are used to establish island models, which start searching at distinct initial points. As the evolution process begins, the exchange of knowledge between islands is synchronously conducted via the migration of elite individuals. The PDECMS is used to solve five discrete sizing optimization problems of a truss structure to demonstrate the achieved solution quality, convergence speed, and scalability. It has been found that the computing time of PDECMS was at least two times faster than its serial implementation for the large population size and the attained solution quality was generally agreeable with other methods despite the sacrifice for the enhancement of performance. Numerical results reveal that the accomplishment of optimal solutions with fewer iterations and a shorter time comes from the cooperative multi-search strategy and the use of GPU. This outcome, therefore, shows that the PDECMS is capable of optimally solving multi-variable problems with a large search space.

1. INTRODUCTION

Structural optimization has received great attention in recent decades. The main topics of this class of optimization include minimization of the structure weight, a discovery of the best possible geometry, and the connection between structural members. In many practical engineering problems, the design variables (i.e., cross-sectional areas or thickness), which are discrete values, are usually from a standard set provided by

^{1), 3)} Professor

²⁾ Master

material manufacturers. Numerous optimization techniques have been developed and successfully employed for many structural problems in the past decades. Some of them are optimality criterion (Fung-Tai & Bo Ping, 1991), force method (Behrooz & Ali, 2010), and surrogate model (Tang, Tong, & Gu, 2005). In addition, a variety of metaheuristic algorithms and their variants are known as direct searching methods, have been presented including ant colony optimization (ACO) (Dorigo, Maniezzo, & Colormi, 1996), genetic algorithm (GA) (Goldberg, 1989), differential evolution (DE) (Storn & Price, 1997), harmony search (HS) (Kang Seok, Zong Woo, Sang-ho, & Kyu-woong, 2005) and artificial bee colony (ABC) (Sonmez, 2011). GA, and DE are found to be the most popular ones. Several problems were analyzed by employing those algorithms. For instance, Ho-Huu et al. (Ho-Huu, Nguyen-Thoi, Vo-Duy, & Nguyen-Trang, 2016) developed adaptive elitist differential evolution (aeDE) to optimize the discrete value of truss structure, Li et al. (Li, Huang, & Liu, 2009) conducted the sizing optimization for truss structures with discrete variables using heuristic particle swarm optimizer (HPSO), and Jawad et al. (Jawad, Mahmood, Wang, AL-Azzawi, & Al-jamely, 2021) used dragonfly algorithm (DA) for the optimal design of truss structure with discrete variables.

Although the enhancement of solution quality of these aforementioned methods and the acceleration of convergence rate were realized, researchers have not considered computing time due to a small number of papers reporting the algorithm's run time. Since there have been numerous researches and works on the parallelization of metaheuristic algorithms, the applications for structural optimization are still limited. Although the execution time was shortened with the help of GPU for computational parallelization the solution quality and robustness of the DE were not impressive.

In this study, the parallel DE (de P. Veronese & Krohling, 2010) is primarily adopted to solve truss optimization problems with discrete design variables. Then, three major changes are introduced for the algorithmic parallelization (Schryen, 2020), to create a so-called parallel differential evolution with a cooperative multi-search strategy.

2. TRUSS OPTIMIZATION WITH DISCRETE DESIGN VARIABLES

The major task in sizing optimization problems with discrete design variable is the selection of optimal cross-sectional area for each structural element in accordance with the standard list. The problem is constructed to find the minimal weight of the structure while satisfy some assigned constraints concerning the structural behaviors and limitations of design variables. The description can be found below

Minimize

$$weight(A) = \sum_{i=1}^e \rho_i l_i A_i, i = 1, \dots, e \quad (1)$$

$$\begin{aligned} \delta_{lower} &\leq \delta_j \leq \delta_{upper}, j=1, 2, \dots, nj \\ \sigma_i^{lower} &\leq \sigma_i \leq \sigma_i^{upper}, i=1, 2, \dots, ne \\ \text{if } \sigma_i &\leq 0, \sigma_i^b \geq |\sigma_i|, i=1, 2, \dots, nc \\ A \in S &= \{A_1, A_2, \dots, A_{ng}\} \end{aligned} \quad (2)$$

where A is the design variable group including the cross-sectional area A_i of the bars; $weight(A)$ stands for the total weight of the structure; ρ_i and l_i are respectively the material density and the length of the i^{th} bar; e denotes the total number of bars in the truss structure; n_j and ne are the number of nodes and the number of elements in the given problem, respectively; δ_i is the so-called nodal displacement and σ_i denotes the normal stress and σ_i^b is the allowable buckling stress for which i^{th} element is under compression. S is set of discrete standard value of areas.

3. THE DIFFERENTIAL EVOLUTION ALGORITHM

3.1 Initialization

First of all, an initial population containing NP individuals, is randomly generated from a given search space. Each individual is described by a vector $X_i = (X_{1,i}, X_{2,i}, \dots, X_{D,i})$ with D members as known as number of dimensions in solution space. Each vector is created by

$$X_{j,i} = X_j^{lower} + rand[0,1] \times (X_j^{upper} - X_j^{lower}), \quad i=1,2,\dots, NP; \quad j=1, 2, \dots, D \quad (3)$$

3.2 Mutation

Secondly, DE obtains the mutant vector V_i^G with respect to each individual X_i^G via its mutation strategies at iteration G . Several common mutation strategies are used in numerous DE variants, are listed below

$$\text{DE/rand/1: } V_i^G = X_{r1}^G + F \times (X_{r2}^G + X_{r3}^G) \quad (4)$$

$$\text{DE/rand/2: } V_i^G = X_{r1}^G + F \times (X_{r2}^G + X_{r3}^G) + F \times (X_{r4}^G + X_{r5}^G) \quad (5)$$

$$\text{DE/best/1: } V_i^G = X_{best}^G + F \times (X_{r1}^G - X_{r2}^G) \quad (6)$$

$$\text{DE/best/2: } V_i^G = X_{best}^G + F \times (X_{r1}^G - X_{r2}^G) + F \times (X_{r3}^G - X_{r4}^G) \quad (7)$$

$$\text{DE/current-to-best/1: } V_i^G = X_i^G + F \times (X_{best}^G - X_i^G) + F \times (X_{r1}^G - X_{r2}^G) \quad (8)$$

To keep the j th component V_{ji} of mutant vector V_i within the allowable region, each component which violated the boundary condition, is handled as follows

$$V_{ji} = \begin{cases} 2X_i^{lower} - V_{ji} & \text{if } V_{ji} < X_i^{lower} \\ 2X_i^{upper} - V_{ji} & \text{if } V_{ji} > X_i^{upper} \\ V_{ji} & \text{otherwise} \end{cases} \quad (9)$$

3.3 Crossover

After the mutation phase, a binomial crossover operator is applied on the target vector X_i^G to obtain a trial vector U_i^G . This is done by simply replacing some components of target vector by some components of trial vector

$$V_{ji} = \begin{cases} V_{ji} & \text{if } rand[0,1] \leq CR \text{ or } j = j_{rand} \\ X_{ji} & \text{otherwise} \end{cases} \quad (10)$$

3.4 Selection

In this phase, the comparison between the trial vector U_i and the target vector X_i is made based on fitness value from the objective function. According to the result of the comparison, the better one which has lower value, will retain for the next generation,

$$X_i^{G+1} = \begin{cases} U_i^G & \text{if } f(U_i^G) \leq f(X_i^G) \\ X_i & \text{otherwise} \end{cases} \quad (11)$$

4. A PARALLEL DIFFERENTIAL EVOLUTION WITH COOPERATIVE MULTI-SEARCH STRATEGY

4.1 The proposed algorithm

By integrating the above-mentioned modifications and mechanism into the parallel DE [x], a so-called parallel differential evolution with a cooperative multi-search strategy is created. The parameters for differential evolution such as F and CR are randomly generated after every iteration. The approach is described in the pseudocode as in algorithm below

Initialize algorithm parameters: N_p , $subPopSize = N_p/3$, $maxIteration$, $updateRate$, $migrationRate$

$numOfExchangedMem = N_p/migrationRate$, $F1 = F2 = F3 = 0.5$, $CR1 = CR2 = CR3 = 0.5$

Memory allocation for host and device variables

Initialize $cuRAND$ states with IVs: state1, state2, state3

Initialize population for three islands

Evaluation all individuals in each island

for $i = 0$ to $maxIteration$

 Generate indices for current-to-best mutation scheme

 Generate indices for rand/2 mutation scheme

 Generate indices for best/2 mutation scheme

 Generate crossover value $rand[j]$ for each component in individual using $curand_uniform [0,1]$

 Generate new population for the first island

 Generate new population for the second island

 Generate new population for the third island

 Perform elitist selection on each island

if $currentIteration \% updateRate = 0$

 Copy all individuals of each island to temporary buffer

 Extract elite individuals from each island through temporary buffer

 Merge all extracted elite individuals

 Extract the best fitness individuals out of the merged population

end if

$F1 = rand[0.3, 0.7]$, $CR1 = rand[0.6, 1]$

$F2 = rand[0.1, 0.5]$, $CR2 = rand[0.2, 0.7]$

$F3 = rand[0.2, 0.7]$, $CR3 = rand[0.1, 0.6]$

end

Post-process the results

4.2 Handling discrete variables

Adjusting the continuous design variables to the nearest discrete value is achieved by a rounding function. The method is described as follows (Ho-Huu, Nguyen-Thoi, Vo-Duy, & Nguyen-Trang, 2016). It has a six-fold symmetry, and is adequately defined by a tensile meridian (when the angle of similarity)

$$X_i^{discrete} = fix(X_i^{continuous}) \quad (12)$$

where $fix(x)$ is a function, which adjusts the given input to the nearest provided discrete value. The function is required before the evaluation of the fitness function happens so that each individual is mapped to the standard set of cross-sectional area.

5. NUMERICAL EXAMPLES

The first planar problem is the 200-bar truss structure shown in Fig. 1. This problem was previously solved by Ho-Huu et al. (Ho-Huu, Nguyen-Thoi, Vo-Duy, & Nguyen-Trang, 2016) using aeDE. The provided properties of this problem are the mass density of 0.285 lb/in.³ and the Young's modulus of elasticity of 30000 ksi. Stress limitations of ± 10 ksi is taken into consideration for all truss members. The 200 bars of this structure are grouped into 29 design variables (Ho-Huu, Nguyen-Thoi, Vo-Duy, & Nguyen-Trang, 2016).

The convergence progress of the PDECMS and the parallel DE is described in Fig. 2 and the execution time is plotted in Fig. 3. Adding 750 to the original of 150 *NP* consequently raised the execution time by 514.19% for the serial version and 44.19% for parallel version. The increment of 300 caused the serial and parallel variants to increase approximately by 108% and 11%. It, therefore, indicates that parallelization empowers the proposed algorithm to deal with a large population size. The PDECMS took 40209 analyses or 67 iterations to converge to the optimal solution while parallel DE was not able to.

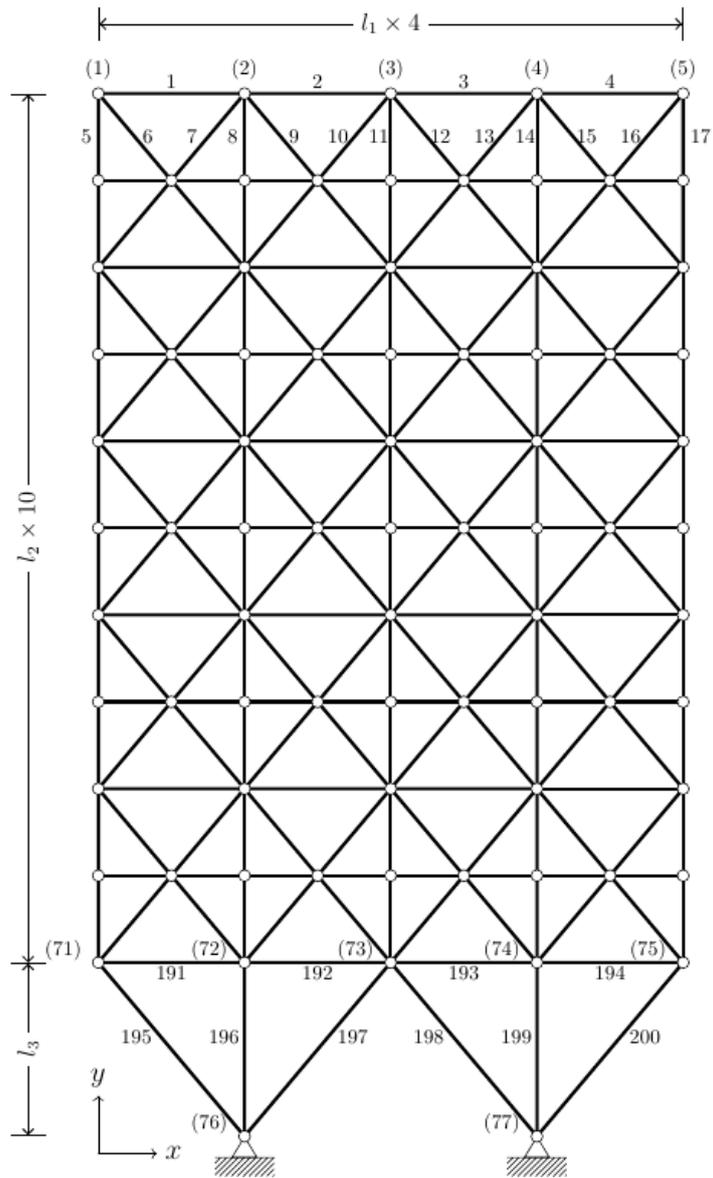


Fig. 1: A 200-bar planar truss structure

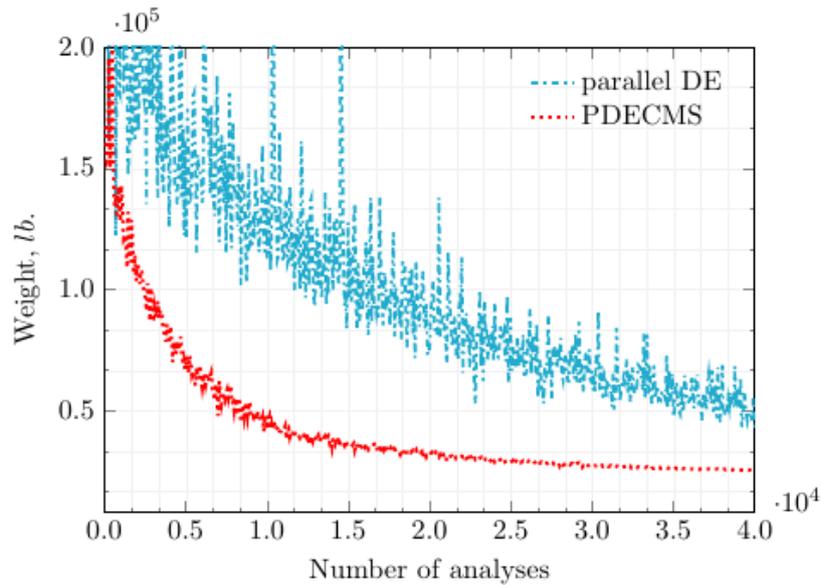


Fig. 2: Comparison of convergence of parallel DE and PDECMS for the 200-bar truss structure

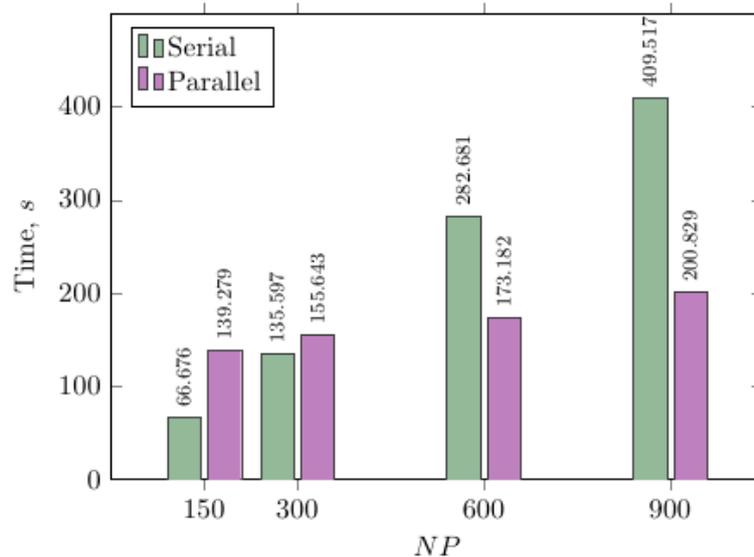


Fig. 3: Comparison of execution time between the serial and parallel versions of PDECMS with different NP values for the 200-bar truss structure

The last problem is the 160-bar truss structure shown in Fig. 4 and this was solved by Groenwold et al. (Groenwold, Stander, & Snyman, 1999) using the regional genetic algorithm (RGA), V. Ho-Huu et al. (Ho-Huu, Nguyen-Thoi, Vo-Duy, & Nguyen-Trang, 2016) using aeDE. The density of the material is 0.00785 kg/cm^3 and the modulus of elasticity is $2.047 \times 10^6 \text{ kgf/cm}^2$. The 160 members of the truss are linked to 38 independent design variables. The bars' linkage and the nodal coordinates of this problem is reused from (Ho-Huu, Nguyen-Thoi, Vo-Duy, & Nguyen-Trang, 2016). The structure is designed to withstand eight load cases provided in (Ho-Huu, Nguyen-Thoi, Vo-Duy, & Nguyen-Trang, 2016).

The best design in the literature is 1336.634 kg, and there is a difference of 4.993 kg to PDECMS's solution. This problem can be considered saturated since there are no significant gaps between solutions. For such a problem, the limitation of using float as a data type was obvious. Since the algorithm was able to converge to the optimal solution at early iteration, it could not evolve any further. The average weight and the standard deviation of PDECMS were the smallest compared to other methods. The convergence of the proposed algorithm and the parallel DE is illustrated in Fig. 5. The PDECMS achieved the optimal solution after 70599 analyses at the 117th iteration, while the parallel DE still did not. The result shows that the exchange of information between islands speeded up the convergence. The completion time of serial and parallel versions of PDECMS is shown in Fig. 6, the addition of 750 to NP increased the execution time by 29.51% and 516.95% respectively for PDECMS and its serial variant.

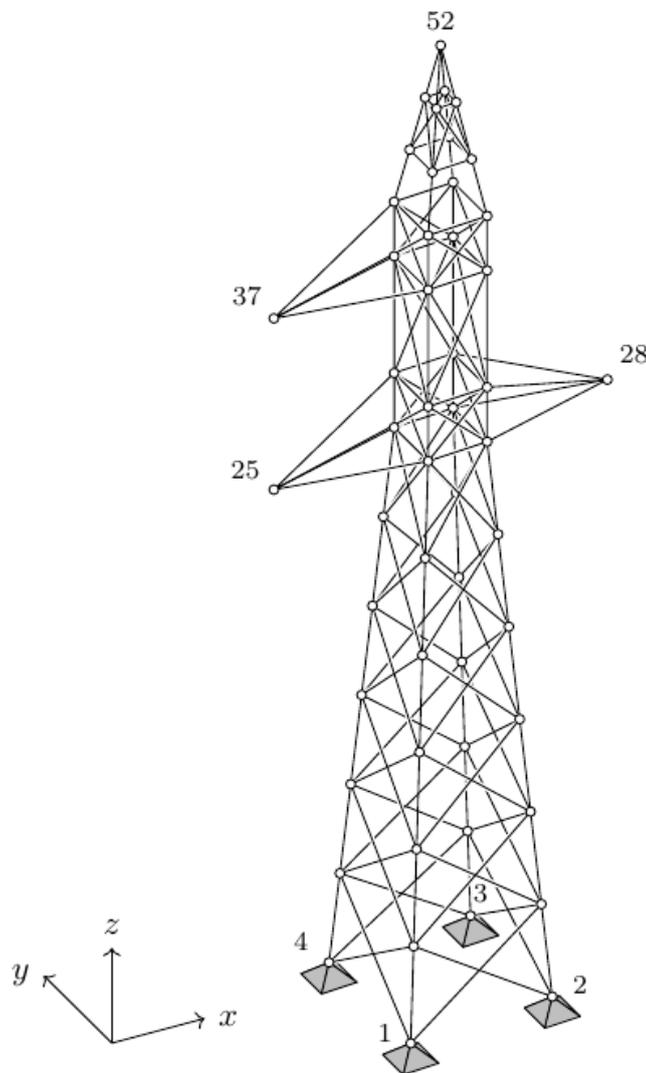


Fig. 4: A 160-bar spatial truss structure

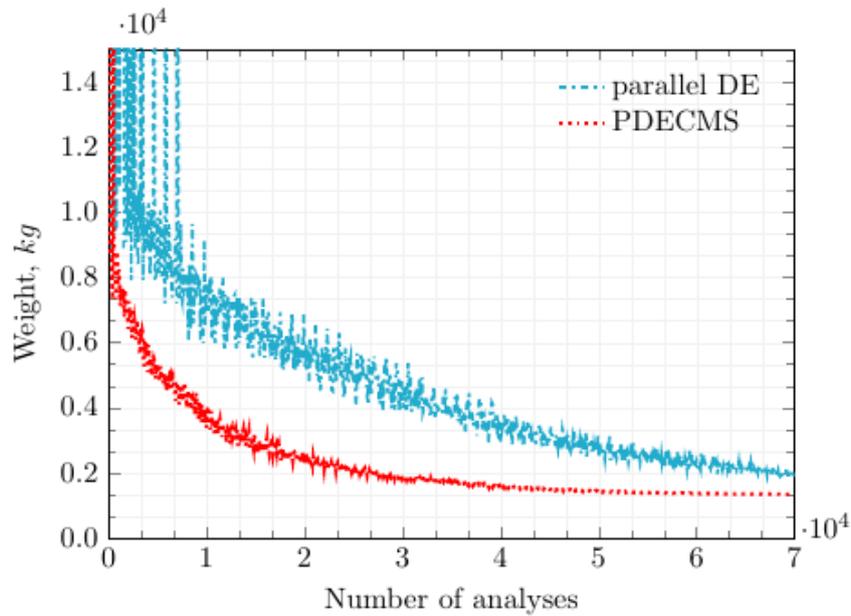


Fig. 5: Comparison of convergence of parallel DE and PDECMS for the 160-bar spatial truss structure

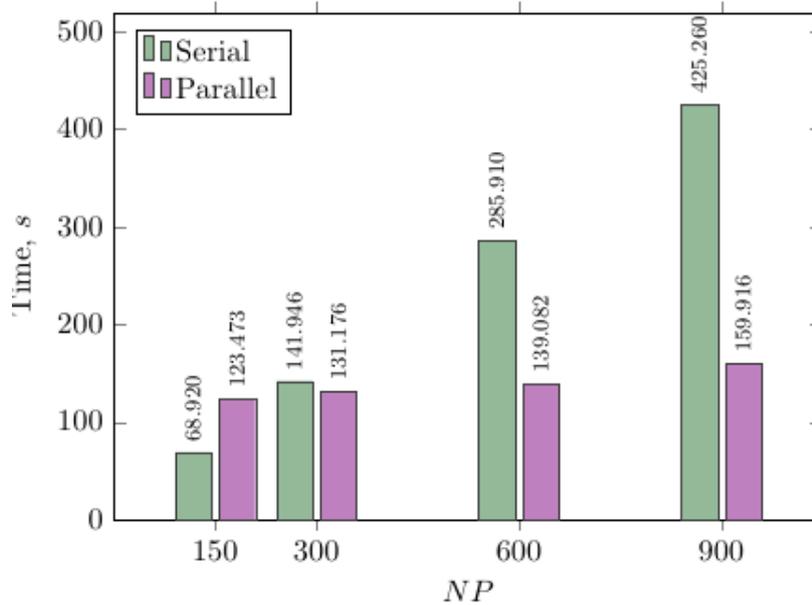


Fig. 6: Comparison of execution time between the serial and parallel versions of PDECMS with different NP values for the 160-bar spatial truss structure

6. CONCLUSIONS

In this study, the integrative framework for the parallelization of the metaheuristic algorithm is employed to accelerate the algorithmic design through the parallel differential evolution with cooperative multi-search for discrete optimization of the truss structure. Three islands that are created to store sub-populations are then initiated with three IVs to achieve the different starting points for the searching process. In addition, the

cooperative coevolution of three islands is carried out with the dedicated mutation scheme being assigned to each island. As a result, the exchanged information is utilized to construct the new elite population and introduced back to each island to boost the solution quality and the convergence.

The PDECMS is then adopted to solve different truss problems with discrete design variables. The numerical outcome shows that the achieved solutions were agreeable or better than other algorithms, although the implementation used the float data type for faster computation and less memory consumption. The coevolution of three islands with the integration of synchronous communication enables the PDECMS to converge faster and find better solutions in comparison with the parallel DE.

REFERENCES

- Behrooz, F., & Ali, A.-z. (2010). Sizing optimization of truss structures by method of centers and force formulation. *Int. J. Solids Struct.*, 2508-2524.
- de P. Veronese, L., & Krohling, R. A. (2010). Differential evolution algorithm on the GPU with C-CUDA. *IEEE Congress on Evolutionary Computation*, (pp. 1-7).
- Dorigo, M., Maniezzo, V., & Coloni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, 29-41.
- Fung-Tai, K., & Bo Ping, W. (1991). An improved method of optimality criteria for structural optimization. *Comput. Struct.*, 629-636.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc.
- Groenwold, A., Stander, N., & Snyman, J. (1999). A regional genetic algorithm for the discrete optimal design of truss structures. *Int. J. Numer. Methods Eng.*, 749-766.
- Ho-Huu, V., Nguyen-Thoi, T., Vo-Duy, T., & Nguyen-Trang, T. (2016). An adaptive elitist differential evolution for optimization of truss structures with discrete design variables. *Comput. Struct.*, 59-75.
- Jawad, F. K., Mahmood, M., Wang, D., AL-Azzawi, O., & Al-jamely, A. (2021). Heuristic dragonfly algorithm for optimal design of truss structures with discrete variables. *Structures*, 843-862.
- Kang Seok, L., Zong Woo, G., Sang-ho, L., & Kyu-woong, B. (2005). The harmony search heuristic algorithm for discrete structural optimization. *Eng. Optim.*, 663-684.
- Li, L. J., Huang, Z. B., & Liu, F. (2009). A heuristic particle swarm optimization method for truss structures with discrete variables. *Comput. Struct.*, 435-443.
- Schryen, G. (2020). Parallel computational optimization in operations research: A new integrative framework, literature review and research directions. *Eur. J. Oper. Res.*, 1-18.
- Sonmez, M. (2011). Discrete optimum design of truss structures using artificial bee colony algorithm. *Struct. Multidiscip. Optim.*, 85-97.
- Storn, R., & Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Global Optim.*, 341-359.
- Tang, W., Tong, L., & Gu, Y. (2005). Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables. *Int. J. Numer. Methods Eng.*, 1737-1762.