©Civil-Comp Press, 2005.
Proceedings of the Eighth International Conference
on the Application of Artificial Intelligence
to Civil, Structural and Environmental Engineering,
B.H.V. Topping (Editor),
Civil-Comp Press, Stirling, Scotland.

Paper 25

Coarse-Grain Parallel Meta-Genetic Algorithms in the Optimization of Truss-Structure Design

V. Esfahanian, A. Khajavi Rad and F. Torabi
Department of Mechanical Engineering
University of Tehran, Iran

Abstract

In this study, a coarse-grain parallel meta genetic algorithm (GA) with dynamic connection scheme has been designed. In order to show the efficiency and robustness of this model, it is implemented on the problem of finding optimal cross-sectional size, topology and configuration of 2D and 3D trusses to achieve minimum weight. Stress, deflection and kinematic stability are regarded as constraints. The results show that this method finds trusses which have smaller weight and better configuration than those, reported in the literature. Finally, the speedup and performance of a number of coarse-grain GAs with various migration methods and connectivity schemes are investigated and compared to show the capability of this new approach.

Keywords: meta-GAs, coarse-grain GAs, parallel processing, premature convergence, migration, truss-structure design.

1 Introduction

Genetic algorithms (GAs) refer to a family of computational models based on the supposed functioning of living; consequently, their evolutionary nature makes them more practical to solve complex nonlinear problems than other classical methods. However, because of their stochastic nature, there are a number of limiting factors which cause no guarantee for successful performance of such algorithms. First, implementation of the GA depends very much on the problem being solved. Moreover, even in a specific problem, the efficiency of GAs is highly dependent on many factors such as genetic operators and their corresponding parameter values, and in general it is not practical to tune these initial settings manually. The other two problems associated with GAs are their high computational cost and propensity to converge prematurely. To cope with the first problem several methods have been introduced to automate parameter tuning (Back, 1992; Lis, 1996; Eiben and

Hinterding, 1999; Lobo, 2000; Espinoza and Goldberg, 2001; Kee and Airey, 2001). Hierarchical GA or meta-GA can be considered as a simple genetic algorithm (SGA) whose individual genomes consist of a set of parameters (i.e. operator types and their corresponding parameter settings) for another GA. The fitness calculation phase involves running its sub-GAs, in order to derive a set of good initial parameters for them (Gerefenstette, 1986; Freisleben and Hartfelder, 1993; Petrovski and McCall, 1998). However, the main concern that has been observed while dealing with meta-GAs, is the high computational cost due to the large number of total evaluations. For this reason, the model can be parallelized by running sub-GAs on various nodes separately and allowing data exchange by migration of individuals between sub-populations asynchronously (Goodman and Punch, 1997). Coarse grain parallel GAs have major advantages over the serial models including rapid exploration resulted from distributed computing and reduction total amount of evaluations due to increasing the search quality. Such a conflation leads to the claim that multi-population GAs have superlinear speedup. Therefore, the global optimum is reached in a reasonable computational time and the premature convergence problem is inhibited.

In this study, a coarse-grain parallel meta-GA with dynamic topology connectivity scheme has been designed. In order to show the efficiency of this model it is implemented on the problem of finding optimal cross-sectional size, topology and configuration of 2D and 3D trusses to achieve minimum weight. In addition, the performance and speedup of this parallel GA model is compared with that of conventional ones.

2 The Coarse-Grain Meta-GA Model

The proposed meta-GA model is a SGA with tournament selection, one-point crossover and bitwise mutation operators. Its chromosome includes: 1) *Selection type*, which includes tournament, roulette wheel and linear ranking selection, 2) *Crossover type* which includes one-point, two-point, uniform and cyclic crossover, 3) *Migration rate* which defines the number of individuals passing to other islands during each migration time, 4) *Crossover rate* and 5) *Mutation rate*. Parallelizing has been achieved through a Beowulf system which contains a cluster of PCs with distributed memory. MPI library has been used for passing messages between various nodes. Lower level GAs run in parallel and after a predefined number of generations, a number of individuals, selected upon the proposed migration strategy, are sent to the other islands asynchronously. That is, during the migration time each node sends immigrants to its neighbours and receive individuals from other islands whenever they were passed to it (if no individuals were passed, the node does not wait but continues its GA search). This kind of communication was achieved through the use of *Remote Memory Access (RMA)* which is supported by *MPI2*. Due to the fact that sub-GAs have different initial parameter settings imposed by the higher level one, this model can be classified as a heterogonous parallel GA. In addition, although applying asynchronous communication in a distributed environment may lead to a higher speedup, this may also result in insertion of high-fitness individuals from a fast-evolution processor into a low fitness population in a slow-evolution one. Consequently, by applying a static connection scheme, the migration of individuals to other islands may not be effective and they may be ignored or dominate the sub-populations. To cope with this problem, a dynamic connection model is proposed which is mutable with time. According to this method, called *distance connection topology*, during each migration time the node's neighbours can be determined based on similarity, and therefore sub-populations with closest *hamming distance* exchange data with each other (Figure 1).

Step1. Initialize individuals in sub-populations in the higher level GA.

Step2. Apply Meta-GA operators (Tournament Selection, One-point Crossover and Bitwise Mutation) in each island.

Step3. Calculate fitness value for Meta-GA individuals.

- **3-1**. Pass the Meta-GA individuals as initial settings to the lower-level GA.
- **3-2**. Initialize population in agents in the lower level GA.
- **3-3.** Apply sub-GA operators (the proper operators are selected as imposed by the higher level GA) in each agent.
- **3-4.** Calculate the fitness value for sub-GAs (which in this model includes calling FEM subroutine and passing the truss weight as the fitness value) in each Sub-population.
- **3-5.** Apply migration to the islands which their generation numbers are equal to the migration interval.
- **3-6.** Go to 3-3 while the generation number is less than a predefined value.
- **3-7.** Set the Meta-GA fitness value equal to the best fitness found in the lower level GA for each island.

Step4. Go to step 2 while the Meta-GA generation number is less than a predefined limit.

Figure 1: The coarse-grain meta-GA algorithm.

3 Truss-Structure Design

3.1 Background

Structural optimization has always attracted the attention of many researchers. Numerous classical and heuristic techniques have been developed in order to automate structural design. The main studies in optimal truss-structure design can be classified in three main categories: sizing, topology and configuration optimization. In sizing optimization, only cross sectional area of members are taken as variables and their connectivity and nodal coordinates remain fixed (Rajeev and Krishnamoorthy, 1992; Coello, 1994). In topology optimization, connectivity of members is to be determined (Krish, 1989; Ringertz, 1985) and in optimizing the configuration of trusses joints coordinates are kept as variables (Imai and Schmit, 1981). However, the most efficient way to achieve the optimal truss is to consider these three categories simultaneously (Rajan, 1997; Deb, 2001). In this paper, the problem of finding the optimal trusses for sizing, topology and shape optimization to achieve minimum weight has been solved using the proposed parallel meta-GA model.

3.2 Proposed Methodology

This paper presents simultaneous sizing, topology and shape optimization of truss-structures for achieving minimum weight by applying the proposed parallel meta-GA scheme. Stress, deflection and kinematic stability are treated as constraints using the exterior penalty method. In sizing optimization, the cross section of each member is taken as a variable which can get any value in a specified range. Topology optimization has been implemented by introducing concepts of ground structure and basic and non-basic nodes [1]. In order to optimize the truss configuration, nodes coordinates must be located in predefined limited range. Thus, the final goal is to find optimal non-basic nodes and their coordinates, required members and their cross-sectional areas to achieve trusses with minimum weight while satisfying the imposed constraints. The NLP nature of the problem and the large number of total variables and constraints results in multimodality of objective function and occurrence of many local optima. However, by using a two level GA and parallelizing it, efficiency of the heuristic model increases and both of the workload and execution time are reduced considerably.

4 Results

In all the following simulations, the maximum generation number, population size, crossover rate and mutation rate of the higher level GA are set to 15, 20, 0.30 and 0.90, respectively. The meta-GA chromosome is used to specify which set of parameters is used for each sub-population (Table 1).

Gene Function	Possible Items
Selection Type	Tournament, Roulette wheel and linear ranking selection
Crossover Type	One-Point, Two-Point, Cyclic and Uniform crossover
Migration Rate (m_{rate})	1.0%-20.0%
Crossover Rate (P_C)	0.60-0.90
Mutation Rate (P_M)	0.001-0.10

Table 1: The Meta-GA Chromosome.

The *migration interval* is set to be 50 and during the migration time, each node compares its best individual with that of others and sends immigrants to those which have a hamming distance less than 50 lb. *Migration strategy* consists of the *selection phase* and the *replacement* one. The selected immigrants fall into two groups: the first half are the best individuals (the ones with smallest weights) and the remaining ones are selected randomly. According to the replacement scheme, worst individuals are replaced with the best ones and the randomly selected immigrants are replaced by those which are selected randomly, too.

4.1 Two-Tier Truss

The coarse-grain meta-GA model has been implemented on the problem of optimizing the two-tier truss [1]. The 39-member, 12-node ground structure (Fig.1), was optimized in the following ways:

1. Sizing and topology optimization.
2. Sizing, topology and shape optimization.

The objective is minimizing the truss weight considering the stress, deflection and kinematic stability as constraints. Material properties and design parameters are to be set as given in Table 2.

Young's Modulus	10^4 ksi
Density	0.10 lb/in^3
Allowable stress	20.0 ksi
Allowable deflection	2.0 in
Maximum area	2.25 in^2

Table 2: Material Properties and Design Parameters for Two-Tier Truss.

In order to provide equal chance for a member's existence in the truss, the area range is set to $-2.25\,\text{in}^2 < A < 2.25\,\text{in}^2$. Each member will be present in the structure if its cross section is greater than a critical value, set optionally $0.05\,\text{in}^2$ here.

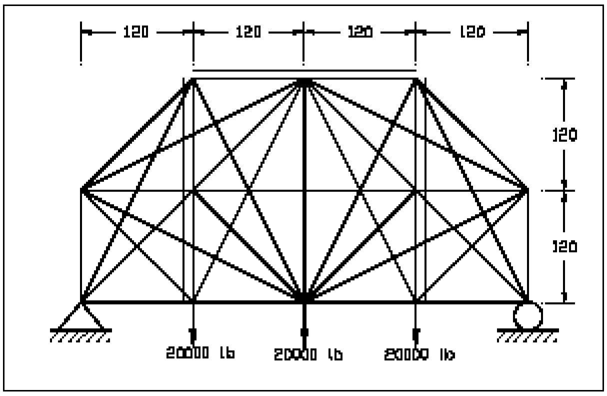
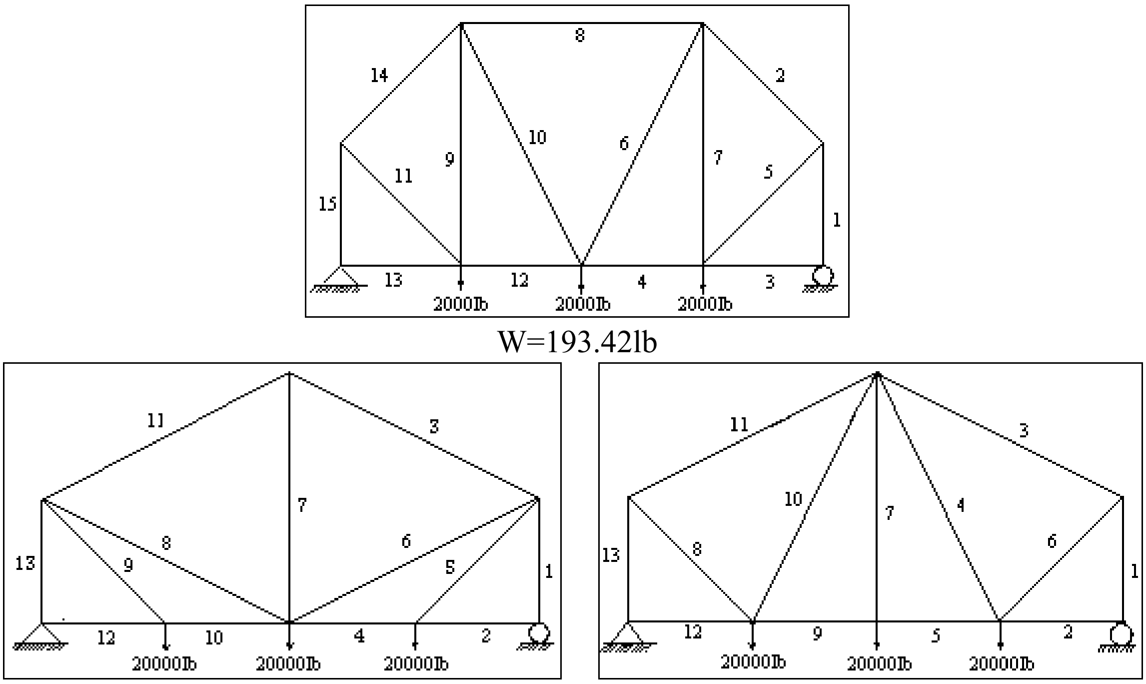


Figure 2: Two-tier, 39-member, 12-node ground structure.

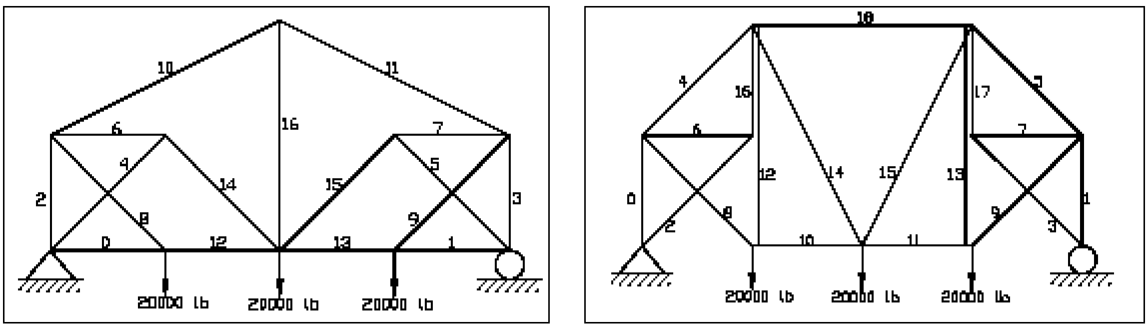
4.1.1 Sizing and Topology Optimization

Sizing and topology optimization has been implemented on the two-tier truss, with total population and maximum generation number equal to 800 and 400, respectively. Four processors were used as subpopulations whose population sizes were equal to the population number applied to the same problem with serial-GA divided by the number of the nodes. After 12 meta-GA generations, optimized trusses were found with lower weight and better topology compared to those reported in the literature (Figs. 3, 4). Cross sectional areas and stresses of the optimized trusses are listed in Table 3. The optimized truss with 196.8 lb resulted from the corresponding serial code is found after 20 generations which reveals the capability of parallel processing in both decreasing the computational time and work of GAs, which leads to a superlinear speedup equal to 5.28.



W=196.04lb W=196.20lb

Figure 3: Optimized trusses for sizing and topology consideration (proposed method).



W=198lb W=196.54lb

Figure 4: Optimized trusses for sizing and topology consideration (Deb [1]).

Member	1, 15	2, 14	3, 13	4, 12	5, 11	6, 10	7, 9	8
Area (in^2)	1.500	1.061	.051	.750	1.065	.560	.250	1.00
Stress (ksi)	-20.00	-19.994	0.0	20.00	19.918	19.965	20.00	-20.00

Table 3a: W=193.42lb

Member	1, 13	2, 12	3, 11	4, 10	5, 9	6, 8	7
Area (in^2)	1.50	.050	1.120	1.00	1.415	.051	1.00
Stress (ksi)	-20.00	0.0	-19.965	20.00	19.989	0.0	20.00

Table 3b: W=196.04lb

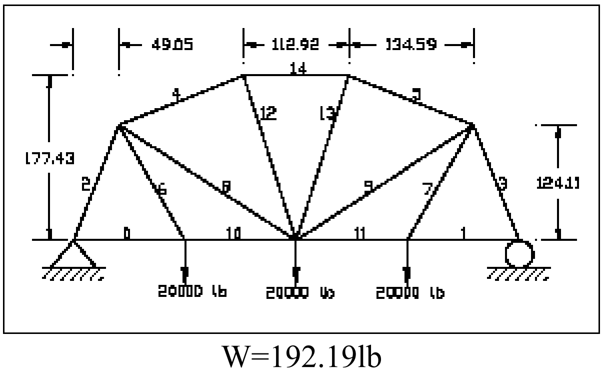
Member	1, 13	2, 12	3, 11	4, 10	5, 9	6, 8	7
Area (in^2)	1.50	.050	1.120	.050	1.00	1.420	1.00
Stress (ksi)	-20.00	0.0	-19.965	0.0	20.00	19.920	20.00

Table 3c: W=196.20lb

Table 3: Member areas and stresses for optimized truss structure in the case of sizing and topology optimization.

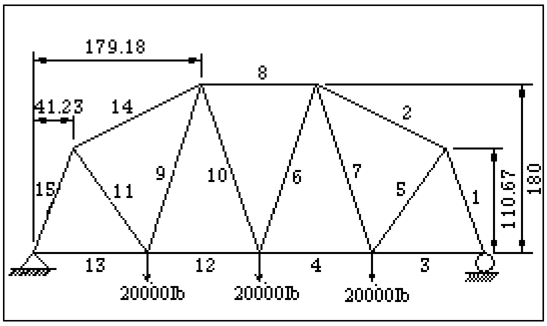
4.1.2 Sizing, Topology and Shape Optimization

In simultaneous sizing, topology and shape optimization, the nodal coordinates of the members are also regarded as variable. These new variables assumed to vary within (-120, 120) in. The meta-GA model has been implemented on this problem with the maximum generation number and total population size of 400 and 1600 in its lower level GAs, respectively. The total population is divided into 8 sub-populations, which run in parallel each with the population size equal to 200. After 15 generations of the meta-GA, trusses with smaller weight compared to those found in the literature were achieved (Figs. 3, 4). The member areas and stresses of the optimized trusses are listed in Table 4. The speedup and the other considerations associated with this parallel scheme will be discussed in section 5.

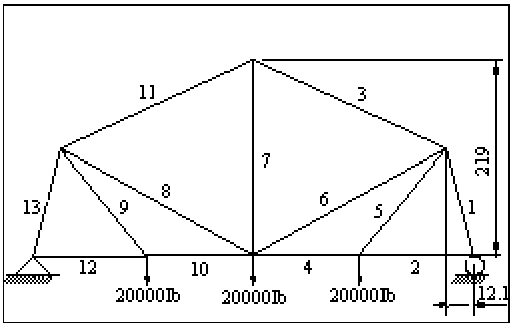


W=192.19lb

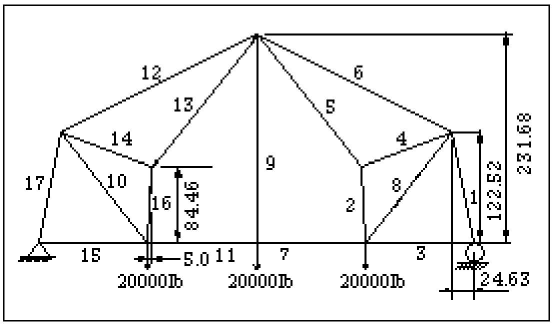
Figure 5: Optimized truss for sizing, topology and shape consideration (Deb [1]).



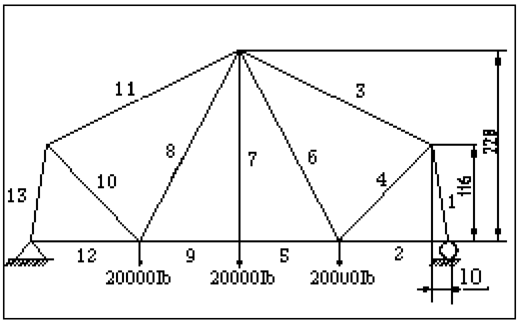
W=190.98lb



W=191.58lb



W=192.63lb



W=192.53lb

Figure 6: Optimized trusses for sizing, topology and shape considerations (Proposed method).

Member	1, 15	2, 14	3, 13	4, 12	5, 11	6, 10	7, 9	8
Area (in^2)	1.600	1.343	.559	1.165	1.102	.528	.108	1.334
Stress (ksi)	-19.996	-19.996	19.994	19.985	19.999	19.995	19.988	-19.999

Table 4a: W=190.98lb

Member	1, 13	2, 12	3, 11	4, 10	5, 9	6, 8	7
Area (in^2)	1.508	.152	1.195	1.051	1.345	.051	.955
Stress (ksi)	-19.996	19.983	-20.00	19.993	19.993	19.945	19.944

Table 4b: W=191.58lb

Member	1, 17	2, 16	3, 15	4, 14	5, 13	6, 12	7, 11	8, 10	9
Area (in^2)	1.53	.054	.304	.05	.054	1.199	1.036	1.199	1.00
Stress (ksi)	-20.0	19.94	20.00	12.78	19.88	-19.99	20.00	19.99	20.00

Table 4c: W=192.63lb

Member	1, 13	2, 12	3, 11	4, 10	5, 9	6, 8	7
Area (in)	1.506	.133	1.182	.0724	1.006	1.303	1.00
Stress (ksi)	-20.00	19.852	-19.989	19.909	19.989	19.988	20.00

Table 4d: W=192.53lb

Table 4: Member areas and stresses for optimized truss structure in the case of sizing, topology and shape optimization.

4.2 Three-Dimensional, 25-Member, 10-Node Truss

In order to apply the meta-GA model on a 3D truss, a 25-member, 10-node ground structure, taken from the literature (Haung and Arora, 1989), is considered for sizing and topology optimization (Fig. 4). Considering symmetry on both opposite sides and cross members, number of variables is reduced to 7. This reduction has been performed as shown in Table 5:

Group	Member
1	0
2	1, 2, 3, 4
3	5, 6, 7, 8
4	9, 10, 11, 12
5	13, 14, 15, 16
6	17, 18, 19, 20
7	21, 22, 23, 24

Table 5: Group membership for the 25-member space truss.

Loading has been implemented by applying four force vectors: (1000; 10000; -5000) lbs on node 1, (0; 10000; -5000) on node 2 and (500; 0; 0) on nodes 3 and 6. Young's modulus and density of the material are taken as before. However, the other settings are to be modified as given in Table 6.

Allowable Stress	40 ksi
Allowable Deflection	0.35 in
Maximum Area	$3.0\,\text{in}^2$

Table 6: Material properties and design parameters for space truss.

In order to provide equal chance for a member's existence in the truss, the area range is set to $-3.0\,\text{in}^2 < A < 3.0\,\text{in}^2$. Each member will be present in the structure if its cross section is greater than a critical value, set optionally $.005\,\text{in}^2$ here.

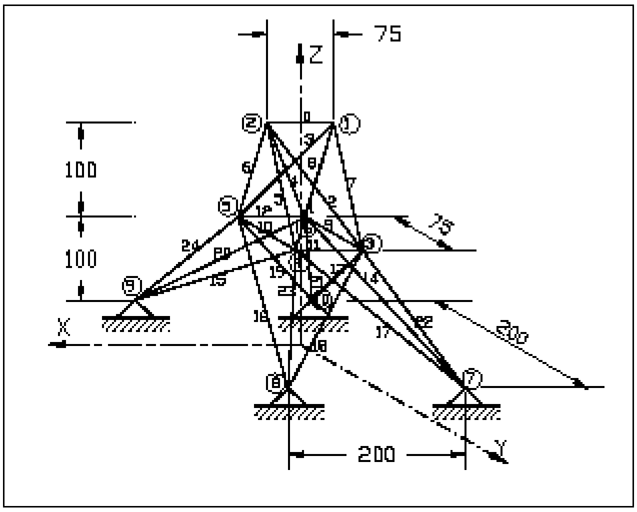


Figure 7: The ground structure for the 3D, 25-member and 10-node truss.

The proposed meta-GA model is applied with the population size and maximum generation number equal to 300 and 400 in its lower level GA, respectively. Due to the lower computational cost and less complexity of the search space for this truss compared to the two-tier one, total population has been divided into two sub-populations which run in parallel and exchange data as before. After 10 generations of the meta-GA a truss with the same topology as was reported by Deb [1], but with a smaller weight was found. The member areas are compared in Table 7.

Member (Fig. 7)	Area (in^2)	
	Proposed Method	Deb (2001)
0	—	—
1, 2, 3, 4	2.65	2.037
5, 6, 7, 8	1.96	2.969
9, 10, 11, 12	—	—
13, 14, 15, 16	.95	.699
17, 18, 19, 20	1.21	1.644
21, 22, 23, 24	2.976	2.658
Weight of Truss (lbs)	537.447	544.852

Table 7: Sizing and topology optimization results for the 3D-ground structure.

5 Parallel Model Efficiency

In order to evaluate the efficiency of our algorithm, we have carried out some experiments on the problem of sizing, topology and shape optimization of the two-tier truss mentioned in section 4.1. In the first phase, to show the improvement of exploration due to the usage of the coarse-grain model, the meta-GA maximum generation number was taken fixed. Since the most important parameters in the performance of parallel GAs are their migration method and connection scheme, five different models were considered to show the capability of our approach. The proposed models are as follow:

1. There is no migration between subpopulations. This is the simplest model of coarse-grain GAs which is often called *isolated island GA*.
2. The second model is a ring topology using synchronous communication; that is, all processors must wait till the slowest node reach to the migration time.
3. The same as model 2 but a dynamic connection scheme, as was mentioned in section 2, was implemented.
4. In this model, processors exchange data asynchronously in a ring topology.
5. The proposed model, which as was mentioned in section 2, is the same as model 4 but with *distance connection topology* scheme.

In all the above models the maximum meta-GA generation number was set to be 25. The subpopulation size is equal to the total population number (1600) divided by the number of processors (1, 2, 4 and 8) being used. The migration interval and scheme were the same as the method discussed in section 3. Each model was run 5 times using different seeds to allow comparisons. The best solution, average of best solution, best speedup and the average speedup are listed for each model (Table 8).

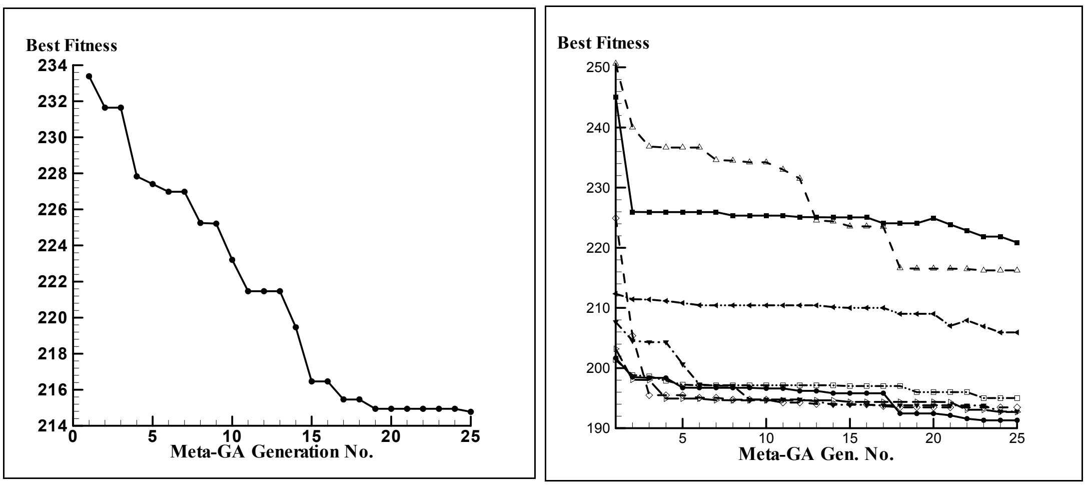
Model	Node Numbers	Best Fitness	Ave. Best	Best Speedup	Ave. Speedup
1	1	214.773	214.773	1	1
	2	194.15	202.325	1.952	1.937
	4	201.456	206.802	4.754	4.651
	8	192.657	209.506	11.039	10.382
2	1	214.773	214.773	1	1
	2	208.141	213.580	1.543	1.526
	4	196.091	211.977	3.613	3.610
	8	193.050	222.035	7.974	7.968
3	1	214.773	214.773	1	1
	2	192.558	194.206	1.512	1.492
	4	193.181	209.313	3.654	3.648
	8	191.15	201.351	7.946	7.941

4	1	214.773	214.773	1	1
	2	206.356	212.472	1.853	1.544
	4	197.832	205.798	3.840	3.412
	8	193.567	207.678	8.712	8.452
5	1	214.773	214.773	1	1
	2	194.432	196.324	1.752	1.523
	4	191.910	201.377	3.745	3.522
	8	190.98	203.546	8.669	8.316

Table 8: Comparisons of various coarse-grain parallel GA architectures.

Considering the above results, the following observations are obtained:

1. Increasing the number of processors will lead to a better exploration of search space. Different high fitness individuals are maintained in different nodes, which decrease the probability of premature convergence (Figs. 8, 9).
2. The speedup of the synchronous models is lower than that of asynchronous ones. Since in distributed workstations there are processors with different speeds; synchronization can cause some nodes to *wait* and slows down the speed of evolution to that of the slowest one.
3. The models with distance topology connection scheme outperform those with a ring topology, which is due to the heterogeneous nature of the meta-GA model. Since in the dynamic topology scheme node's neighbours are selected upon similarity, the problem of injecting incompatible individuals is inhibited.



(A) (B)

Figure 8: Best fitness vs. number of meta-GA generations, (A) Serial GA, (B) Model 5 with 8 subpopulations.

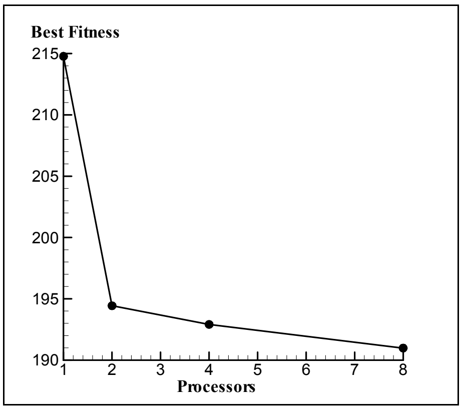


Figure 9: Best fitness vs. number of CPUs (Model 5).

In all above experiments, the speedup was evaluated for a fixed number of meta-GA generations. Consequently, results cannot show the reduction of workload due to parallelism. Thus, it is better to compute speedup based on a predefined fitness value, which in this problem was set to 192 lb (a smaller weight than those found in the literature). Since one node experiment cannot reach the desired search quality, the two-node execution time was considered as the base and the speedup was evaluated according to it. The experiments were conducted on the asynchronous meta-GA with dynamic topology scheme. The speedup and efficiency of the parallel model are shown in figures 10 and 11, respectively.

Figure 8 shows a high *superlinear* speedup which indicates that not only was there a speedup from parallel processing, but also from reduction of workload due to applying the multi-population model. This can be observed for some heuristic algorithms, which the total amount of work differs by varying the number of processors being applied.

In order to show the concept of automating parameter tuning implemented by the meta-GA model, crossover, mutation and migration rates are sketched for model 5 with 4 processors as subpopulations (Figs. 12-14). These parameters are associated with the elite individual found in each meta-GA generation. As it can be seen from these figures, by introducing the concept of meta-GA, it is possible to find a set of optimized initial settings for lower level GAs, which increase their efficiency considerably.

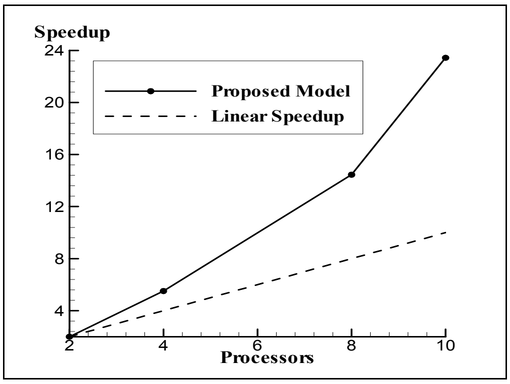


Figure 10: Speedup vs. number of CPUs.

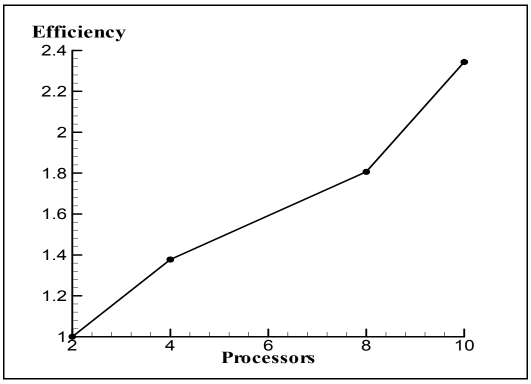


Figure 11: Efficiency vs. number of CPUs.

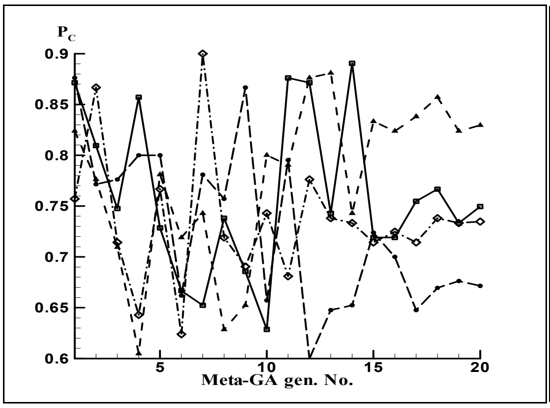


Figure 12: Crossover rate vs. meta-GA generation number.

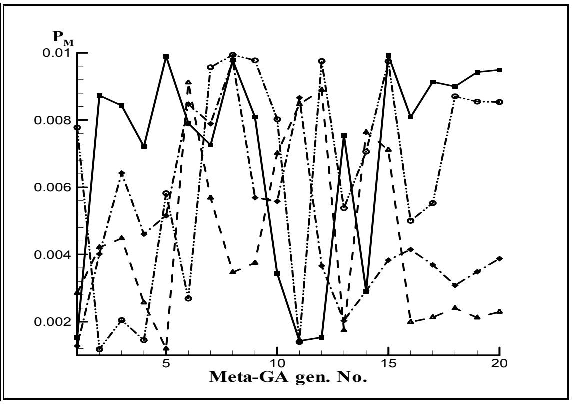


Figure 13: Mutation Rate vs. meta-GA generation number.

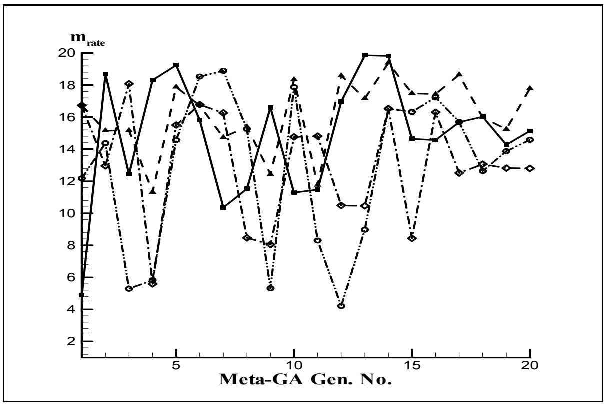


Figure 14: Migration rate vs. meta-GA generation number.

6 Conclusions

This paper presents a coarse-grain meta-GA implemented on the problem of sizing, topology and shape optimization of 2D and 3D trusses to achieve minimum weight. The proposed model was both efficient in automating parameter tuning and preventing premature convergence. The results show that this scheme finds trusses with lower weight and better configurations than those reported in the literature. In addition, by using the multi-population model, different high fitness individuals were found in various islands which reveal the capability of parallel processing in finding the local optima in multi-modal functions. The proposed model was compared with a number of coarse-grain GAs with different migration methods and connectivity schemes. It was found that this model outperforms the schemes with synchronous communication and static connection topology. Finally, by defining the speedup as the *time-to-solution,* high superlinear speedup was observed, indicating the effectiveness of distributed processing in both decreasing the execution time and workload of GAs.

Acknowledgment

The authors would like to thank ADSRC for computer resources and also the support of the Vehicle, Fuel and Environment Research Institute (VFERI) and University of Tehran. The authors would also like to acknowledge Arman Hajati for his valuable advices, which improved the quality of this paper.

References

[1] K. Deb and S. Gulati, "*Design of truss-structures for minimum weight using genetic algorithms,*" Journal of Finite Elements in Analysis and Design, 2001.

[2] K. Sarma and H. Adeli, "*Bilevel Parallel Genetic Algorithms for Optimization of Large Steel Structures*", Computer-Aided Civil and Infrastructure Engineering, 2001

[3] S. Lin, W. F. Punch, and E. D. Goodman, "*Coarse grain parallel genetic algorithms: Categorization and new approach*", The Sixth IEEE Symposium on Parallel and Distributed Processing, 1994, pp. 28-37.

[4] E. Goodman, W. F. Punch and V. Uskov, "*Optimization of a GA and within a GA for a 2-Dimensional Layout Problem",* First International Conference on Evolutionary Computation and its Applications, 1996.

[5] S. D. Rajan, "*Sizing, shape and Topology Optimization of Trusses Using Genetic Algorithm",* Computer Methods in Applied Mechanics and Engineering, 1995.

[6] S. Rajeev and C. S. Krishnamoorthy, "*Discrete Optimization of Structures Using Genetic Algorithms",* Journal of Structural Engineering, 1992.

[7] B. H. V. Topping, "*Shape Optimization of Skeletal Structures: A review",* Journal of Structural Engineering, 1983.

[8] E. J. Haung and J. S. Arora, "*Introduction to Optimal Design",* New York: McGraw Hill, 1989.

[9] M. J. Quinn, "*Parallel Programming in C with MPI and Open MP"*, McGraw Hill, Singapore, 2003.

[10] "*MPI2: Extensions to Message Passing Interface*", Message Passing Interface Forum, 1997.