Hindawi Publishing Corporation
Modelling and Simulation in Engineering
Volume 2009, Article ID 615162, 28 pages
doi:10.1155/2009/615162

Research Article

A New Genetic Algorithm Methodology for Design Optimization of Truss Structures: Bipopulation-Based Genetic Algorithm with Enhanced Interval Search

Tugrul Talaslioglu

Technical Training College of Kadirli, Osmaniye Korkut Ata University, Osmaniye, Turkey

Correspondence should be addressed to Tugrul Talaslioglu, talaslioglu@cu.edu.tr

Received 24 January 2008; Revised 29 September 2008; Accepted 26 January 2009

Recommended by Waleed Smari

A new genetic algorithm (GA) methodology, Bipopulation-Based Genetic Algorithm with Enhanced Interval Search (BGAwEIS), is introduced and used to optimize the design of truss structures with various complexities. The results of BGAwEIS are compared with those obtained by the sequential genetic algorithm (SGA) utilizing a single population, a multipopulation-based genetic algorithm (MPGA) proposed for this study and other existing approaches presented in literature. This study has two goals: outlining BGAwEIS's fundamentals and evaluating the performances of BGAwEIS and MPGA. Consequently, it is demonstrated that MPGA shows a better performance than SGA taking advantage of multiple populations, but BGAwEIS explores promising solution regions more efficiently than MPGA by exploiting the feasible solutions. The performance of BGAwEIS is confirmed by better quality degree of its optimal designations compared to algorithms proposed here and described in literature.

1. Introduction

The steel structures consist of hot-rolled steel profiles with different cross-sectional properties. The optimum design of steel structures is considered as a constrained optimization problem. Modern optimization methods used in the design of steel structure as well as in a number of engineering design problems are inspired by natural phenomenon, such as survival of the fittest, immune system, swarm intelligence, simulating annealing, and ant colony (Saka [1]). These methods explore the problem space utilizing the global or local search-based algorithms. Moreover, it is also possible both to incorporate a local search algorithm into a global search, namely, hybridization of algorithms (memetic algorithms) and to run them in parallel (Moscato [2], Radcliffe and Surry [3], Cantú-Paz [4]).

The powerful member of these algorithms is evolutionary algorithms (EAs). EAs mimic the process of natural evolution. The evolutionary computation is achieved by either simultaneously examining and manipulating a set of possible candidate individuals or using a special individual along with its neighbors in the generation of new individuals.

Genetic algorithm (GA), a member of EAs, is a population-based global search technique based on the Darwinian evolutionary theory (Holland [5], Goldberg [6]). The preliminary approach of GAs is SGA (see a pseudocode in Algorithm 1). SGA guides the evolutionary search by a single population P_i. The size of P_i is denoted by SP. Individuals are encoded in a string scheme associated with one of the codes of the binary, integer, and real. In the evolutionary search, the promising individuals $P_{i-\text{sel}}$ and $P_{i+1-\text{sel}}$ are chosen from the population by a selection operation (roulette wheel, stochastic universal sampling, ranking, truncation, etc.). Then, the individuals chosen are applied to recombination and mutation operation (one or multipoint crossover and mutation, uniform crossover, etc.). These evolutionary operations (mutation mut, crossover cr, and selection sel) are governed by their related evolutionary parameters Par (mutation and recombination probability rates, selection pressure, etc.). The population P_{new} evolved by the application of these evolutionary operators is decoded. Then, the fitness values are computed by use of this population. The evolutionary search is executed to transmit (migration) the individuals (emigrant and immigrants)

SGA (P_i, NG, SP, F_i, Par_{sel}, Par_{mut}, Par_{cr})
If $[P_i] = []$, Initialize (P_i, SP, NDV)
for $i = 1$: NG
$[P_i^d] = P_i$
If required, $[P_i^d]$ = Decoding (P_i)
If $[F_i] = []$, $[F_i]$ = Fitness_Calculation (P_i^d)
$[P_{i-\text{sel}}]$ = Selection(P_i, F_i, Par_{sel})
$[P_{i+1-\text{sel}}]$ = Selection(P_i, F_i, Par_{sel})
$[P_{\text{new}}] = P_{\text{new}}$ U Crossover($P_{i-\text{sel}}$, $P_{i+1-sel}$, Par_{cr})
U Mutation($P_{i-\text{sel}}$, $P_{i+1-\text{sel}}$, Par_{mut})
$[P_i] = [P_{\text{new}}]$
end

Algorithm 1: Pseudocode for SGA.

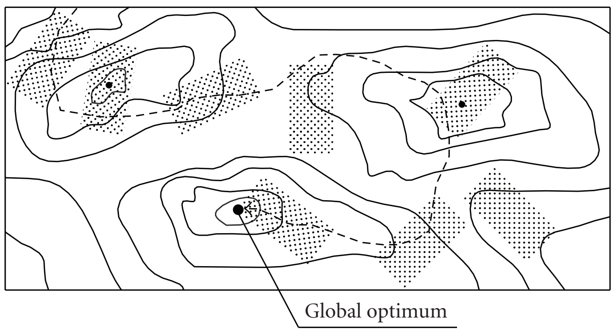


Figure 1: General visualisation of the populations randomly scattered in search topography.

to the next populations until satisfying a predetermined stopping criteria (e.g., completion of a generation number NG).

SGA is more flexible optimization tools. Therefore, it is possible to achieve a balance between two main genetic features: exploration of promising locations in the search space and exploitation of best solutions obtained. The accuracy of this balance has a big effect in the determination of SGAs' performance associating with the quality of solution, speed of convergence and generation of feasible solutions, and so forth. If this balance is not appropriately achieved throughout the generations, a stagnation problem in the progression of evolutionary search is occurred after an equilibrium state. This equilibrium state is called immature convergence. Figure 1 (depicted the local maxima by +'s) can be used for highlighting the reason of immature convergence. As seen in Figure 1, exploration of the trajectory to the global maxima is provided only by maintaining the migration of the populations. That is, if keeping the variations among the populations and the diversity within populations, then the quality of migration will increase, and so, the exploration of the global maxima embedded in one of the subregions will be more powerful in attributes of the geographically nearby populations. In this regard, the task about how the computational cost required for this balance to be minimized leads to the emergence of new GAs.

In this study, a bipopulation-based genetic algorithm methodology named BGAwEIS, whose crucial elements are developed by utilizing the fundamentals of SGA, is applied for the design optimization of truss structures. BGAwEIS utilizes feasible solutions to collect valuable genetic heredity from potential ancestors and to transmit it to offspring. For this purpose, two populations are employed for the transmission process. An intensive search of subregions of entire solution region is provided by gradual exploration strategy developed for BGAwEIS. Moreover, the dominance of similar feasible solutions in next generations is prevented by recreation of the populations at certain generation numbers. In order to asses the quality of optimal designations generated by BGAwEIS, optimal design results obtained by both SGA and existing approaches outlined in the literature are considered. Furthermore, a multipopulation-based genetic algorithm (MPGA) approach is proposed to investigate the effect of usage of multiple and single populations on the quality degree of optimal designations. For this purpose, an optimization tool called GEATbx coded in MATLAB is utilized to compute the evolutionary processes of MPGA (Pohlheim [7]).

This paper is organized as follows. The next section presents a background concerned existing design optimization approaches; Sections 3 and 4 contain the optimum design problem and main elements of BGAwEIS including the basic principles of MPGA associated with GEATbx; design details and examples are provided in Sections 5 and 6, sequentially; conclusion is presented in last section following Section 7 that summarizes the discussion of results.

2. The Review of Major Design Optimization Approaches Used in the Design of Steel Structures

A summary of major optimization approaches and their applications to the design optimization of steel structures are presented by a brief introduction. In this regard, the first part reviews the preliminary studies. Second part evaluates the evolutionary algorithms including their hybrid and parallel models. In this summary, it is intended to present the most representative works in a chronological order.

2.1. Preliminary Studies. The preliminary studies on the design optimization of steel structures are based on gradient-based mathematical programming techniques. Linear programming approach was widely utilized for weight minimization of truss structures, considering structural responses under both elastic and plastic behaviors (Cornell [8]), Bigelow and Gaylord [9]). Nonlinear programming was usedas an alternative method to linear programming. Majid and Elliott [10] applied nonlinear programming to optimize the weight of a two-bay four storey frame. Afterwards, sequential linear and quadratic programming techniques (SLP and SQP) were widely used for the design optimization of steel structures. Vanderplaats and Sugimoto [11] developed a design technique "automated design synthesis" utilizing the approaches of SLP and SQP. Automated design synthesis method was proposed for minimizing the weight of frames with various bays and stories under static and seismic loadings by Karihaloo and Kanagasundaram [12], and Gülay and Boduroğlu [13]. The optimization techniques based on nonlinear programming were used for generation of optimal designations for steel structures under different loading conditions and design requirements (Lassen [14], Wang and Grandhi [15], Salajegheh [16], Hernández [17]).

Optimality criteria method (OCM) is another challenging method. Hybridizing the nonlinear mathematical programming with Lagrange multipliers for inclusion of constraints forms the basis of OCM (Arora [18], Cameron et al. [19]). Saka and Hayalioglu [20] used OCM for optimization of geometrically nonlinear steel structures made of elastoplastic material. Hayalioglu and Saka [21] proposed OCM for design optimization of frames with nonuniform cross-sections. Chan et al. [22, 23] carried out the weigh minimization of three-dimensional steel structures with discrete cross-sections by OCM. They devised a transformation process for continuous and discrete design variables. Saka [24] optimized a frame design with tapered members thereby firstly computing the member responses under external static loadings and combining them by Lagrange multipliers to generate depth variables. Saka and Kameshki [25] used OCM for design optimization of unbraced rigid frames considering constraints imposed by sway deflections and member stresses.

2.2. Evolutionary-Based Optimization Studies. Evolutionary computation based on simulation of natural evolutionary is a new approach used in the design optimization of steel structures. Due to being appropriate for both traditional and novel computation applications in the field of structural engineering, evolutionary approaches whose major members are GAs by Holland [5], evolutionary programming (GP) by Fogel et al. [26] and evolutionary strategies (ES) by Rochenberg [27] have been improved by new implementations, such as hybrid and parallel searches. Therefore, research developments on three major EAs are firstly reviewed. Then, subsection provides an overview of recent developments concerned the issues of parallel and hybrid implementations.

GP is managed by programs defined by point-labeled parse trees used to describe the node and elements in the steel structure. The most important step in GP is the determination of the size and shape of parse trees for a design problem (Keijzer and Bobovic [28, 29]). Cevik [30] used a GP methodology, namely, a gene-expression programming, for determining rotation capacity of wide flange beams.

ES uses a population of tentative design solutions and generates the populations using several genetic operators with self-adaptive parameters (Back and Schwefel [31, 32]). Cai and Thierauf [33] proposed an evolutionary strategy without self-adaptive parameters for the design optimization of steel structures. Similar approaches were also utilized for the design optimization, such as ES with self-adaptive parameters for discrete and continuous design variables (Ebenau et al. [34], Rajasekaran [35], and Baumann and Kost [36]).

One evolutionary algorithm approach is the SGA. Due to its flexible structure, its genetic components have been improved. Taking into account the usage of genetic components, the studies are grouped into two general categories.

(i) Genetic Operators with Adjustable Parameters and Representation of Design Variables. Hajela [37] introduced a representation technique for discrete design variables. Thus, the lower and upper bounds of continues design variables were used to compute the values of discrete design variables with any determined precision. He discussed the negative effect of higher precision values leading to a larger-length binary representation. Adeli and Cheng [38] presented a decoding technique in order to use binary coded strings for continues design variables. Chen [39] discussed the lack of proportional selection operation leading to stagnation problem in evolutionary search. He showed that usage of both scaled fitness values which are computed considering certain statistical quantities of fitness values and crossover operator which was applied at different rates to the same individual improved the quality degree of optimal designations. Yang and Soh [40] pointed out that the tournament selection method was more efficient than the existing selection methods considering quality degree of optimal designations. The use of graph theory for GAs is one

of the new developments in structural optimization problem. Wang and Tai [41] devised a graph representation for the topology-related design variables in structural optimization problems. Kaveh and Kalatjari [42] utilized the graph theory for representation of the size-related design variables and force method for structural analysis. Thus, grouping of truss design variables according to magnitude and sign (compression and tension) of stress becomes a new challenging approach in the design optimization of steel structure (Saka [43], Saka et al. [44], Toğan and Daloğlu [45]).

(ii) Constraint Handling for Evaluation of Fitness Values. Camp et al. [46] devised a penalty function with several variables for design optimization of two-dimensional steel structures. The negative effect of this approach leading to an inaccurate penalization was shown by Rasheed [47]. In order to cope with this task, Rasheed [47] utilized an adaptive approach for handling the constraints. For this purpose, a penalty function was used to compute the penalty values based on an adaptation of penalty coefficients with respect to the penalty degree. Le Riche et al. [48] divided the population into small groups and applied a penalty function with different variable coefficients for each group. Coello [49] used two populations for the generation of a new population thereby comparing the penalized values. Nanakorn and Meesomklin [50] improved the penalty function by an adaptive procedure.

One of the alternative approaches to the penalty function is artificial immune-inspired model (Garrett [51]). Firstly, Yoo and Hajela [52] utilized this approach for solving design optimization problem. They employed two populations: one population was used to compute penalty values, while the other one measured the hamming distance between penalized fitness values. Coello and Cruz Cortés [53] improved Yoo and Hajela's technique by devising an adaptive evolutionary mechanism against the necessity for a penalty function.

Hybrid and Parallel Search-Based Evolutionary Algorithm Approaches. In order to improve the flexibility and efficiency of evolutionary algorithms, utilizing the hybrid or parallel models of evolutionary search algorithms is one of the important attempts.

The hybridization concept is emerged by use of local search methods for evolutionary algorithms as a complementary tool. Local search methods are proposed to propagate the genetic information obtained throughout evolutionary process into the next generations. One powerful hybridization model is memetic algorithm. This biological learning mechanism is associated with Dawkins' notion of a meme defined as a unit of cultural evolution (Dawkins [54]). Two important approaches, namely, Lamarckian and Baldwinian approaches make use of this learning mechanism for their evolutionary processes. Whereas the structure of chromosome and its fitness value are changed in the Lamarckian' approach, Baldwinian' learning mechanism only affects the fitness value of chromosome without any change in its structure. In the hybridization of evolutionary algorithms with local search method, the converging speed to the global optima may be lower than the case of using a pure evolutionary search algorithm without any hybrid implementation (Ong and Keane [55]). In order to increase the converging speed, a new memetic algorithm, namely coevolving memetic algorithm, is developed. Its fundamentals as well as a comprehensive review of the basic approaches based on this algorithm are presented by Smith [56].

The concept of parallel search is introduced to evolutionary algorithms thereby employing a number of computer processes with distributed or shared memories for a global population or a divided global population into small populations (subpopulations) (Cantú-Paz and Goldberg [57],Cantú-Paz [58, 59]). Parallel systems not only preserve diversity within the current populations, but also ensure a perpetual novelty for populations to be generated in a way of disseminating the different characteristic features embedded in the chromosomes to next populations. Among evolutionary algorithms the GAs are preferably chosen for the parallel applications. The basic genetic models utilized in parallel search are grouped into four main classes (Cantú-Paz [4]).

(a) *Master-slave GA.* It uses a single population, while master processor is employed to collect valuable genetic information, slave processors service are responsible to compute the fitness values for a certain number of individuals (Grefenstette [60], Robbins [61] and Levine [62]).

(b) *Fine grained GA or cellular GA.* The larger number of processors is assigned for fitness evaluation of subpopulations. Due to the higher number of processors, one of difficulties is encountered during the decision about how the computer processors to be designed and arranged (Baluja [63]).

(c) *Coarse grained GA or distributed GA.* It has several similar properties of fine grained GA (migration implementation and multiple population usage) with an exception of using smaller number of processors. The ease of designing and arranging the computer processors makes this model more attractive compared to cellular GA (Herrera et al. [64]). The distributed GA is performed depending on migration related parameters (policy, topology, frequency etc.) (Cohoon et al. [65], Alba and Troya [66], Skolicki and De Jong [67]). Moreover, the other important issues for migration policy are the number and frequency of migration, replacement of immigrants, size of populations and migration topology (Tanese [68, 69]). The migration related processes are directly responsible for determining the excitation order of computer processors. Moreover, if any computer processor waits to run the migration process for exchanging the individuals, then this parallel search is called as "synchronous," otherwise "asynchronous" (Alba and Troya [70]).

One of the basic models utilized by the distributed GA is island model. Several distinct subpopulations are isolated with each other, but communicated by a migration process. Evolutionary operators are applied to each subpopulation. If the parameter values of evolutionary operators are adjustable for each subpopulation being important for an independent exploration of different region of search space, then island model of this type is named as homogenous and nonhomogeneous distributed GA (Alba and Troya [70]).

(d) *Hybrid GA.* Distributed GA can be straightforwardly implemented on the parallel systems consisted of a number of computer processors providing a considerable profit to the evolutionary search in a way of decreasing the computing time. Therefore, by itself, distributed GA is hybridized with existing search methods at different hierarchical levels. The hierarchical level is determined according to the use of evolutionary tools and operators for the structured population. In the determination of any hierarchical scheme, one of the most important steps is how the population to be structured according to a lattice-like topology.

The cellular GA is also successfully used in the hybridization models (Martin et al. [71]). However, the complexity degree of cellular GA is higher than distributed GA due to the increased size of both its subpopulations and underlying grid system consisted of computer processors. Alba and Troya [70] compared the cellular, distributed GAs and their hybrid models. They showed that numeric efficiency and resistance to scalability was increased by the distributed versions of cellular GA. In order to improve the exploitation capability of the cellular GA in a way of increasing the converging rate, it was hybridized with a local search method, named as cellular memetic algorithm. (Folino et al. [72]). Afterwards, cellular memetic algorithm was enhanced by an implementation of an adaptive mechanism. (Krasnogor and Smith [73]), Neri et al. [74], Caponio et al. [75], and Quang et al. [76]).

Sakamoto and Oda [77] hybridized GAs with optimality criteria method for topology and size optimization of truss structures. While the topology of truss structure was evolved through GA, optimality criteria methods determine the cross-sectional areas of truss bars. Soh and Yang [78] devised a GA approach that is managed by the fuzzy-logic-based rules. This approach was applied for weight minimization of structures and achieved to obtain more optimal designations compared to SGA's.

Adeli and Cheng [79] were developed a parallel GA called "concurrent GA." They utilized a number of computer processors in parallel for the design optimization of truss and frame structures. Topping and Leite [80] utilized this parallel GA for the design optimization of a bridge, considering a number of constraints. Adeli and Kumar [81] used a network consisted of computer processors for optimization of large-spaced steel structures. Sarma and Adeli [82] hybridized the coarse grained GAs with the fuzzy logic search method, for design optimization of three-dimensional frame structures.

Tanimura et al. [83] proposed an island model for design optimization of truss structures taking into account several constraints. They utilized a new penalty function and compared their optimal designations with SGA. They showed that their island model was more efficient than SGA. Kicinger et al. [84] utilized the island models for both topology and size optimization of tall buildings made up with steel profiles. They used two migration topologies (ring and fully-connected topology) with various migration strategies for the design optimization of two-dimensional frame with the bracing elements of various types. They showed that the quality degree of optimal designations is improved when island models were executed by using higher number of subpopulations. Then, Kicinger and Arciszewski [85] made use of MAs in the design optimization of same steel structure. Examining various genetic operators and their related parameters, they showed that the MAs were more successful than GAs.

Kaveh and Shahrouzi [86] proposed implementing the graph theory for MAs. Lamarckian and Baldwinian approaches were adapted to optimize the frame bracing layouts of steel frames. Moreover, the application of these approaches is illustrated for a two-dimensional steel frame. They compared their optimal results with SGAs and displayed that whereas Lamarckian approach reduces the topological variance with a more converging rate, the better results are obtained by an incorporation of a dynamic mutation band control to the Baldwinian approach.

Karakasis et al. [87] devised a radial basis function network for the distributed GA and applied it to an aerodynamic shape optimization problem. They compared four variants of GA and concluded that their distributed versions offer an additional advantage in the exploration of the interconnected processor network. Then, in order to carry out the shape optimization of same design problem, they devised a hierarchical distributed evolutionary scheme thereby adapting both the aerodynamic design formulation and a navier-stoke equation solver into a radial basis unction network (Karakasis and et al. [88]). Liakopoulos et al. [89] utilized a grid system consisted of a number of computer processors for performing the hybridization of hierarchical and distributed algorithms.

3. Optimum Design Problem

In this study, the weight of steel structure is minimized by taking the constraints of maximum allowable stresses and displacements into account. The evolutionary operations are operated on a population of tentative designations with binary, integer, and real codes which contains the design variables of discrete and continues types. Genetic operators are carried out by use of either phenotypic or genotypic representations of design variables. The representations of design variables encoded in genotype level are either kept in all levels of evolutionary computation or decoded for fitness evaluation in phenotypic level. The fitness values of tentative design solutions are adjusted according to the violation of constraints. In case of constrain violation; the penalized value is included into fitness value by a penalty function.

The weight of truss system and constraints are formulated as

$$W = \sum_{i=1}^{n} \rho * L_i * A_i, \tag{1}$$

subject to

$$\begin{aligned} \sigma_i &\leq \sigma_{\max} \quad i = 1, \ldots, n, \\ U_k &\leq U_{\max} \quad k = 1, \ldots, m. \end{aligned} \tag{2}$$

Here, W represents weight of the truss system. ρ is the density of steel, L_i and A_i are the length and cross-sectional area of ith member, respectively; n is the total number of members in the truss system. σ_i and $\sigma_{\max}$ symbolize the stress and the maximum allowable stress for ith member. U_k is the displacement at kth degree of freedom while m is the total degree of freedom of nodes. $U_{\max}$ represents the maximum allowable displacement for kth degree of freedom. Constraints g_s and g_d controls the joint displacements and element stresses, considering the allowable displacement and stress values. The number of constraints is determined by $s_{\max}$ and $d_{\max}$ which indicates the number of joints and displacements to be constrained.

The violation of constraint is penalized. The penalization process is used to obtain a penalty value. Thus, the fitness value F is obtained by the sum of weight of the truss system W and penalty value P. F is used in weight minimization of truss system. The minimization process is formulated as

$$\operatorname{Min} F = W + P, \tag{3}$$

where the term "W" is given in (1) and P is

$$P = (r_o * t)^{\varphi} * \left(\sum_{i=1}^{n} g_i + \sum_{k=1}^{m} g_k \right) * f. \tag{4}$$

In (4), the stress constraint is expressed as

$$g_s = \begin{cases} \dfrac{\sigma_i}{\sigma_{\max}} - 1: & \sigma_i \leq \sigma_{\max} \\ 0: & \sigma_{\max} > \sigma_i \end{cases} \qquad i = 1, \ldots, n,\ s = 1, \ldots, s_{\max}, \tag{5}$$

and displacement constraint as

$$g_d = \begin{cases} \dfrac{U_k}{U_{\max}} - 1: & \sigma_k \leq {}_{\max} \\ 0: & U_{\max} > \sigma_k \end{cases} \qquad k = 1, \ldots, m,\ d = 1, \ldots, d_{\max}. \tag{6}$$

The values of the constants in the calculation of the penalty value P are taken as $r_0 = 0.50$, $\varphi = 2$, $f = 10$, and t = current generation number as given in Hasançebi and Erbatur [90].

4. An Introduction of BGAwEIS and Multipopulation-Based Genetic Algorithm (MPGA)

The main features of BGAwEIS are similar to the island models with respect to the usage of multiple populations and static parameters in the evolutionary operators. In order to asses the effect of multiple populations on the quality of optimal designations, MPGA is proposed. It is able to perform the evolutionary processes with one processor and also capable of performing the evolutionary operations with static parameters on multiple populations. The fundamentals of BGAwEIS and MPGA are summarized in the following sections.

4.1. BGAwEIS. Parallel GAs are perfect evolutionary tools due to its flexibility structure which is adaptable to various environmental conditions. They utilize a number of processors and populations simultaneously. Considering the elevated number of interacting characteristics, it is said that parallel GAs have "complex mechanisms." While using smaller number of populations decreases this complexity, the quality of optimal solutions drops due to insufficient exploitation of genetic heredity. On the other hand, with increasing number of populations the adjustment of the values of related evolutionary parameters becomes increasingly difficult and cause a slow down in the variation among populations. This effect prevents the exploration of promising solution regions (Cantú-Paz [4]). Therefore, a balance between exploitation of genetic heredity and exploration of promising solution regions should be achieved. For this purpose, an appropriate number of populations must be used for transmitting of the genetic heredity extracted from high-quality solutions. In this regard, a new GA, namely, bipopulation-based Genetic Algorithm with Enhanced Interval Search (BGAwEIS), is developed. The basic features of BGAwEIS are itemized as follows.

(i) The design constraints may increase the complexity of the search in the solution region (Eiben and Ruttkay [91]). The largeness of the solution region affects the exploration efficiency of the GA. If the feasible solutions are utilized in the exploration of the solution region, then more promising individuals located in some other regions may be obtained. Therefore, BGAwEIS utilizes feasible solutions in order to compose the genetic heredity. The valuable genetic heredity obtained is adapted to current populations by transmission processes called "extraction" and "insertion-based transmission processes."

(ii) Two populations called "outward" and "inward" within a core population are used in transmission processes in order to investigate the unknown subsolution regions and use the genetic information obtained from previously visited candidates to explore better candidates. Transmission process is achieved by regenerating a population through migration among the feasible solutions taking into account of gradual exploration strategy developed for utilizing the promising subsolution regions of the entire solution region. Because, the exploration capacity is increased by dividing the entire solution region into subsolution regions. As a result, promising feasible solutions are used to explore more promising solution regions.

(iii) The similar feasible solutions which may be dominated in the search or feasible solutions obtained may be not enough to explore the entire solution region. Therefore, the core population is recreated at certain generation numbers.

(iv) The evolutionary processes are governed by four parameters depending on the number of design

BGAwEIS (*SubPopNum* = 1, *SubPopIndNum* = SP, SSR,NDV,NG, NGGES, NSBS, Par_{mut}, Par_{cr}, Par_{sel})
Initialize (P_i, *SupPopNum*, *SupPopIndNum*)
$[P_{\text{cor}}] = P_i$
$[\text{Par}^d_{\text{cor}}] = P_i$
If required, $[\text{Par}^d_{\text{cor}}]$ = Decoding (P_{cor}),
$[F_{\text{cor}}$, FeasPool] = Fitness_Calculation ($\text{Par}^d_{\text{cor}}$)
for i = 1: NG
$[P_{\text{inw}}, P_{\text{outw}}]$ = Extraction_based_transmission(F_{cor}, P_{cor}, FeasPool)
$[F_{\text{inw}}, F_{\text{outw}}$, FeasPool] = Fitness_Calculation(P_{inw}, P_{outw})
$[P_{\text{cor}}]$ = Insertion_based_transmission(F_{inw}, P_{outw}, P_{inw}, P_{outw}, FeasPool)
$[P_{\text{cor}}$, NSBS] = Re-creation_Population(F_{inw}, F_{outw}, P_{inw}, P_{outw}, SSR, NSBS, FeasPool)
$[P_{\text{cor}}]$ = SGA(P_{cor}, NG = 1, SP, F_{cor}, Par_{sel}, Par_{mut}, Par_{cr})
$[\text{Par}^d_{\text{cor}}] = P_{\text{cor}}$
If required, $[\text{Par}^d_{\text{cor}}]$ = Decoding (P_{cor}),
$[F_{\text{cor}}$, FeasPool] = Fitness_Calculation ($\text{Par}^d_{\text{cor}}$)
end

Algorithm 2: Pseudocode for BGAwEIS.

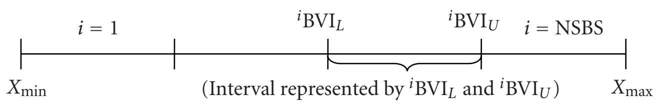


Figure 2: Visualization of the one-dimensional subsolution regions.

variables, size of solution region, and SGA-related mutation, crossover, selection parameters (Par_{mut}, Par_{cr}, Par_{sel}): size of populations, number of generations, number of subsolution before search and number of generation for exploration strategy.

The main elements of BGAwEIS are described in the following sections. An example which clarifies how BGAwEIS works is also included.

Main Elements of BGAwEIS. BGAwEIS completes one generation after five interdependent procedures with two populations within a core population (see the pseudocode in Algorithm 2). The number of subpopulations *SubpopNum* indicated by the number of populations which is obtained by dividing the global population into small populations is taken as 1; the number of individuals contained in each subpopulation *SubpopNum* is equal to the value of parameter *SP*.

The populations are called inward population P_{inw}, outward population P_{outw} within core population P_{cor}. These populations have the same total number of individuals and every individual in each population has the same number of design variables assigned to it. The interdependent procedures are extraction-based transmission, fitness calculation, insertion-based transmission, recreation of the population and application of SGA operators. In addition, gradual exploration strategy is applied for utilizing promising solution regions.

Several parameters are specified prior to the evolutionary process of BGAwEIS considering size of solution region (SSR) and number of design variables (NDV). These parameters are number of generations (NG), size of population (SP), number of generations for gradual exploration strategy (NGGES), and number of subsolution regions before search (NSBS). The data outcome after the completion of search is number of feasible solution (NFS) and number of subsolution regions after the search (NSAS).

The solution regions are composed of a design vector $X = (x_1, x_2, \ldots, x_n)$ which consists of n design variables indicating the cross-sectional areas of the truss members. The design variable has an upper bound $X_{\max}$ and lower bound $X_{\min}$. The value of any discrete design variable in one-dimensional solution region will be between $X_{\min} = 1$ and $X_{\max}$. $X_{\max}$ defines the total number of different cross-sectional areas in the discrete design variables set.

BGAwEIS works on a multidimensional solution region which is divided into one-dimensional subsolution regions and accomplishes the search within these solution regions, simultaneously. In this regard, one-dimensional subsolution region bounded by $X_{\max}$ and $X_{\min}$ is divided into equal segments as shown in Figure 2. The number of segments is denoted by NSBS. The value of NSBS is proportional to SSR. The bounds of each segment are ${}^i\text{BVI}_L$ and ${}^i\text{BVI}_U$ ($i = 1, \ldots$, NSBS). If desired, NSBS can be changed.

The boundaries of subsolution regions are gradually enlarged. This approach is called “gradual exploration strategy” and activated by NGGES. The value of NGGES is specified by the ratio of NG to NSBS. NSBS is proportional to parameter SSR. After the current generation number becomes equal to the value of NGGES, the value of NSBS is decreased. Thus, the bounds of subsolution regions are enlarged.

4.1.1. Extraction Based Transmission. In this process, the individuals that come from the core population are regenerated in order to generate inward ${}^{SP}P_{\text{inw}}$ and outward ${}^{SP}P_{\text{outw}}$ populations. The number of individuals located in these populations is limited by SP. In the construction of the inward population, the individuals taken from core ${}^{SP}P_{\text{cor}}$ population are regenerated by converging them to the best solution of the feasible pool X_{BF}.

In the generation of the outward population, the individuals taken from core population are regenerated by diverging them to the bounds $X_{\min}$ and $X_{\max}$ of the design variable. Furthermore, in the generation of the outward and inward populations, the one-dimensional solution region is divided into equal segments. While these segments are used to generate the outward population, the position of the best feasible solution with respect to these segments is used to generate inward population. In order to regenerate the individuals of ${}^{SP}P_{\text{inw}}$ and ${}^{SP}P_{\text{outw}}$, the corresponding segment ${}^{i}\text{BVI}_L$ and ${}^{i}\text{BVI}_U$ used for the individuals located in ${}^{SP}P_{\text{cor}}$ is determined. This is followed by finding out the position of the best feasible X_{BF} solution in the feasible solution pool to the corresponding segment ${}^{i}\text{BVI}_L$ and ${}^{i}\text{BVI}_U$. There are three possible locations for X_{BF} relative to the corresponding segment (Figure 3): (i) below, (ii) above, and (iii) within the corresponding segment.

${}^{SP}P_{\text{outw}}$ is regenerated by taking into account the segment ${}^{i}\text{BVI}_L$ and ${}^{i}\text{BVI}_U$ used for the individuals of ${}^{SP}P_{\text{cor}}$. The individuals of ${}^{SP}P_{\text{outw}}$ are forced to simultaneously diverge to both $X_{\min}$ and $X_{\max}$ by taking into account X_{BF}, ${}^{i}\text{BVI}_L$, ${}^{i}\text{BVI}_U$, and ${}^{SP}P_{\text{cor}}$. An algorithm based on the possibilities shown in Figure 3 is developed for the regeneration of ${}^{SP}P_{\text{outw}}$ from ${}^{SP}P_{\text{cor}}$ (see (7)). In order to explore entire solution region, two individuals are simultaneously produced for the regeneration of ${}^{SP}P_{\text{outw}}$,

$$
{}^{SP}P_{\text{outw}}^{\text{NDV}} = \begin{cases}
\left.\begin{array}{ll}
X_{\min} + \text{rand}({}^{i}\text{BVI}_L - X_{\min}) : (\text{Indv. 1}) & \text{if } {}^{i}\text{BVI}_L < X_{\text{BF}} < {}^{k}P_{\text{cor}}^{\text{NDV}} \\
{}^{k}P_{\text{cor}}^{\text{NDV}} + \text{rand}(X_{\max} - {}^{k}P_{\text{cor}}^{\text{NDV}}) : (\text{Indv. 2}) & \\
X_{\min} + \text{rand}({}^{k}P_{\text{cor}}^{\text{NDV}} - X_{\min}) : (\text{Indv. 1}) & \text{else} \\
{}^{i}\text{BVI}_U + \text{rand}(X_{\max} - {}^{i}\text{BVI}_U) : (\text{Indv. 2}) &
\end{array}\right\} & \begin{cases}\text{if } {}^{i}\text{BVI}_L < X_{\text{BF}} < {}^{i}\text{BVI}_U \\ (1.\text{possibility})\end{cases} \\
\begin{array}{l}
X_{\min} + \text{rand}({}^{i}\text{BVI}_L - X_{\min}) : (\text{Indv. 1}) \\
X_{\text{BF}} + \text{rand}(X_{\max} - X_{\text{BF}}) : (\text{Indv. 2})
\end{array} & \begin{cases}\text{elseif } X_{\text{BF}} > {}^{i}\text{BVI}_U \\ (2.\text{possibility})\end{cases} \quad (k = 1, \ldots, \text{SP}) \\
\begin{array}{l}
X_{\min} + \text{rand}(X_{\text{BF}} - X_{\min}) : (\text{Indv. 1}) \\
{}^{i}\text{BVI}_U + \text{rand}(X_{\max} - {}^{i}\text{BVI}_U) : (\text{Indv. 2})
\end{array} & \begin{cases}\text{else} \\ (3.\text{possibility})\end{cases} \quad (i = 1, \ldots, \text{Number of Current Intervals})
\end{cases} \tag{7}
$$

${}^{SP}P_{\text{inw}}$ is regenerated by taking into account the segment ${}^{i}\text{BVI}_L$ and ${}^{i}\text{BVI}_U$ used for the individuals of the ${}^{SP}P_{\text{cor}}$. The individuals of ${}^{SP}P_{\text{inw}}$ are forced to converge to the X_{BF} by taking the ${}^{i}\text{BVI}_L$, ${}^{i}\text{BVI}_U$ and ${}^{SP}P_{\text{cor}}$ into consideration. An algorithm based on the possibilities shown in Figure 3 is developed for the regeneration of ${}^{SP}P_{\text{inw}}$ from ${}^{SP}P_{\text{cor}}$:

$$
{}^{SP}P_{\text{inw}}^{\text{NDV}} = \begin{cases}
\begin{array}{ll}
{}^{i}\text{BVI}_L + \text{rand}({}^{SP}P_{\text{cor}}^{\text{NDV}} - {}^{i}\text{BVI}_L) & \text{if } {}^{i}\text{BVI}_L < X_{\text{BF}} < {}^{SP}P_{\text{cor}}^{\text{NDV}} \\
{}^{SP}P_{\text{cor}}^{\text{NDV}} + \text{rand}({}^{i}\text{BVI}_U - {}^{SP}P_{\text{cor}}^{\text{NDV}}) & \text{else } {}^{i}\text{BVI}_L
\end{array} & \begin{cases}\text{if } {}^{i}\text{BVI}_L < X_{\text{BF}} < {}^{i}\text{BVI}_U \\ (2.\text{possibility})\end{cases} \\
{}^{i}\text{BVI}_L + \text{rand}(X_{\text{BF}} - {}^{i}\text{BVI}_L) & \begin{cases}\text{if } X_{\text{BF}} > {}^{i}\text{BVI}_U \\ (3.\text{possibility})\end{cases} \\
X_{\text{BF}} + \text{rand}({}^{i}\text{BVI}_U - X_{\text{BF}}) & \begin{cases}\text{else } (i = 1, \ldots, \text{Number of Current Intervals}) \\ (1.\text{possibility})\end{cases}
\end{cases} \tag{8}
$$

An application of extraction-based transmission on two-dimensional solution region represented by two design variables is graphically shown in Figure 4. Only design variable 1 that is bounded by ($X_{\min}$ and $X_{\max}$) is visualized. The solution region has three optimum points. One of these three optimum points is a global optimum symbolized by "•". The remaining feasible solutions are also indicated by "•". The individuals from the core population, which are indicated as "+" and enclosed in a thin dashed closed curve, are used to build the inward and outward populations (Figure 4). In the regeneration of the inward and outward populations, the bounds of segment corresponding to the individual of the core population are in the range of (${}^{i}\text{BVI}_L$ and ${}^{i}\text{BVI}_U$). Outward population is symbolized by "∗" and enclosed in a rectangle with dashed edges (Figure 4). The individuals of inward population is symbolized as "x" and enclosed in a rectangle with a thick continuous edge (Figure 4).

4.1.2. Fitness Calculation. The module of fitness calculation computes the fitness values of individuals in each population

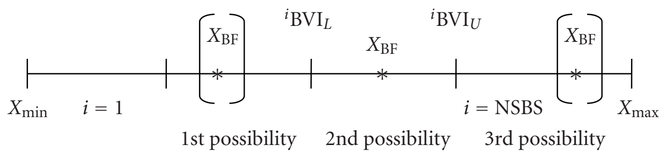


FIGURE 3: Graphical depiction of the possibilities for the location of best feasible solution (X_{BF}) into the corresponding segment in extraction-based transmission.

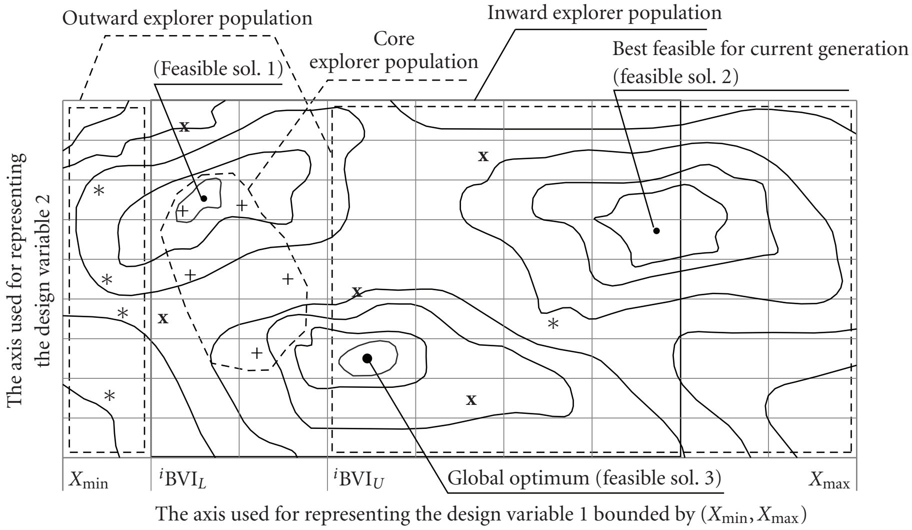


FIGURE 4: Display of the extraction-based transmission.

by taking the constraint violations into account. Thus, fitness values corresponding to populations P_{inw}, P_{outw} and P_{cor} are collected in the matrices F_{inw}, F_{outw}, and F_{cor}. At the same time, feasible solutions obtained are chosen to locate in feasible solution pool when its fitness values are lowest compared to other feasible fitness values found in the feasible solution pool.

4.1.3. Insertion-Based Transmission. This process involves construction of the core population with the individuals coming from inward and outward populations. The core, inward and outward populations all have the same number of individuals. Since one population is generated from two populations, it is necessary to eliminate a certain number of individuals. This is carried out by prioritizing certain individuals according to their feasibility and fitness. All the feasible solutions located in feasible solution pool are used in constructing the core population. In order to adapt the feasible solution pool to the core population, the inward or outward populations is divided into two equal parts. The algorithm developed in this regard is managed by four cases based on the position of the number of feasible solutions with respect to the inward or outward populations with same number of individuals, as depicted in Figure 5. These cases are as follows.

Case 1 (C1) The number of feasible solutions in inward population N_{I1} is more than SP/2.

Case 2 (C2) The number of feasible solutions in inward population N_{I2} is less than SP/2.

Case 3 (C3) The number of feasible solutions in outward population N_{O1} is more than SP/2.

Case 4 (C4) The number of feasible solutions in outward population N_{O2} is less than SP/2.

The core population from the inward and outward populations is constructed from the combination of these four cases. These combinations are C1+C3, C1+C4, C2+C3, and C2+C4. They are explained as

(i) collect $(N_{I1} + N_{O1})$, $(N_{I1} + N_{O2})$, $(N_{I2} + N_{O1})$, and $(N_{I2}+N_{O2})$ feasible solutions from inward or outward populations corresponding to the combinations of "C1+C3," "C1+C4," "C2+C3," and "C2+C4," respectively;

(ii) rank the collected feasible solutions in a descending order of their fitness values, and then store it in a dummy column matrix;

(iii) if the number of individuals in the combination of "C1+C3," "C1+C4," "C2+C3," and "C2+C4" is greater than SP, $((N_{I1}+N_{O1})-\text{SP})$, $((N_{I1}+N_{O2})-\text{SP})$, $((N_{I2} + N_{O1}) - \text{SP})$ and $((N_{I2} + N_{O2}) - \text{SP})$ feasible solutions with least fitness are discarded from the dummy column matrix. Otherwise, $(\text{SP} - (N_{I1} + N_{O1}))$, $(\text{SP} - (N_{I1} + N_{O2}))$, $(\text{SP} - (N_{I2} + N_{O1}))$ and $(\text{SP} - (N_{I2}+N_{O2}))$ feasible solutions with least fitness

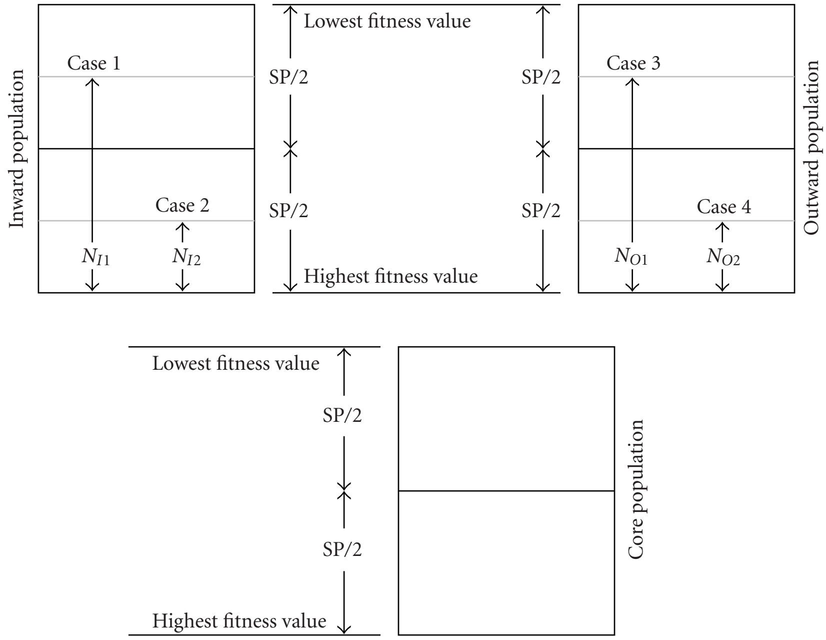


Figure 5: Insertion-based transmission.

is discarded from the dummy matrix, respectively. If the number of individuals in dummy matrix does not reach SP, some individuals are borrowed from the inward and outward populations in the descending order of their fitness values.

4.1.4. Recreation of the Core Population. An algorithm is developed for the recreation of core ${}^{SP}P_{cor}$ population. This algorithm is managed by three possibilities regarding the position of the boundaries, namely $X_{F\min}$ and $X_{F\max}$, of the feasible solution pool with respect to the center (cent) of the interval representing the one-dimensional solution region. These possibilities are graphically depicted in Figure 6. The recreation process is activated depending on NGGES and the enlargeable bounds of subsolution regions.

The algorithm for the recreation of the core population ${}^{SP}P_{cor}$ based on the possibilities given in Figure 6 is formulated by (9),

$$
{}^{SP}P_{cor}^{NDV} = \begin{cases}
\left.\begin{array}{l}\text{A Case: } 1 \le (\text{SP}^*(30/100)) \\ \text{B Case: } (\text{SP}^*(30/100)) < k \le (\text{SP}^*(50/100)) \\ \text{C Case: } (\text{SP}^*(30/100)) < k \le \text{SP}\end{array}\right. & \text{if cent} > X_{F\max} \\[2ex]
\left.\begin{array}{l}\text{A Case: } 1 \le (\text{SP}^*(40/100)) \\ \text{B Case: } (\text{SP}^*(40/100)) < k \le (\text{SP}^*(60/100)) \\ \text{C Case: } (\text{SP}^*(60/100)) < k \le \text{SP}\end{array}\right. & \text{elseif } X_{F\min} \le \text{cent} \le X_{F\max} \quad (k = 1, \dots, \text{SP}) \\[2ex]
\left.\begin{array}{l}\text{A Case: } 1 \le (\text{SP}^*(50/100)) \\ \text{B Case: } (\text{SP}^*(50/100)) < k \le (\text{SP}^*(70/100)) \\ \text{C Case: } (\text{SP}^*(70/100)) < k \le \text{SP}\end{array}\right. & \text{else}
\end{cases} \tag{9}
$$

where A Case : $X_{\min} + \text{rand}(X_{F\min} - X_{\min})$; B Case : $X_{F\min} + \text{rand}(X_{F\max} - X_{F\min})$; B Case : $X_{F\max} + \text{rand}(X_{\max} - X_{F\max})$.

4.1.5. Application of SGA Operators. SGA operators are used to regenerate the core population in order to provide a variation for the next generations. These are one-point crossover, mutation, and roulette wheel selection operators. Also, some optional operators exist for the search including what follows (Eiben and Ruttkay [91]).

(i) Multipoint mutation and crossover operators, and the other selection operators (stochastic universal sampling and stochastic remainder sampling),

(ii) the generation gap against genetic drift problem,

Table 1: A preliminary demonstration of GAwEIS to a planar truss with two-bars.

	Indiv.	Section properities	NSBS	$X_{\min}$	$X_{\max}$	DVN	VNDV	SN	BVI_L	BVI_U
Continous design variables	Indiv. (1)	[0.32 0.56]	4	0.1	1	1	0.32	1	0.100	0.325
						2	0.56	3	0.551	0.775
	Indiv. (2)	[0.82 0.24]	4	0.1	1	1	0.82	4	0.776	1.000
						2	0.24	1	0.100	0.325
Discrete design variables	Indiv. (1)	[001011]	4	1	5	1	1	1	0.100	0.325
						2	3	3	0.551	0.775
	Indiv. (2)	[100001]	4	1	5	1	4	4	0.776	1.000
						2	1	1	0.100	0.325

NSBS: number of subsolution regions (the bumber of segment), VNDV: value of each design variable, DVN: design variable number, SN: subsolution region (segment) number.

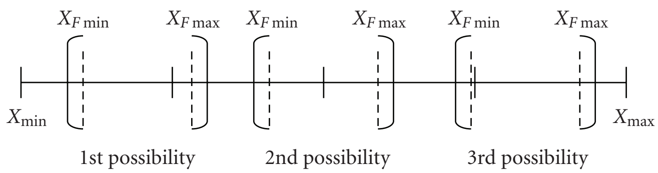


Figure 6: Graphical depiction of possibilities for the location of the interval $(X_{F\min}, X_{F\max})$ and cent = $(X_{F\min} + X_{F\max})/2$ in the recreation of the core population.

(iii) different fitness scaling methods such as linear normalization, baseline windowing, sigma truncating, linear scaling, and adaptive windowing are employed along with the elitist selection scheme against the loosing of the valuable genetic heredity.

The five interdependent procedures mentioned above are processed in one of the real, integer, and binary coding schemes. For this reason, recoding of the design variables is required before these processes are applied.

In order to depict the gradual exploration strategy, let us consider a planar truss with two bars as an example and construct a core population with two individuals. The upper and lower bounds of the design variables are given for continuous or discrete set of design variables (see Table 1).

The solution region of each design variable is divided into 4 segments which are used for subsolution regions obtained by dividing entire solution region into small ones. Numerical values of design variables vary within the interval (0.1, 1) for continuous type and discrete type of design variables. The one-dimensional solution region is divided into one-dimensional subsolution regions. One-dimensional subsolution regions are represented by intervals (0.100, 0.325), (0.326, 0.550), (0.551, 0.775), and (0.776, 1.000). The discrete design variables are coded by using three-binary-digits. Therefore, the total number of digits is equal to 6. The continuous design variables are used to find the corresponding intervals whereas the values of discrete design variables represent the segment numbers.

4.2. MPGA. MPGA makes use of a migration process with several parameters in order to provide a control for multiple populations and a communication between them. The evolutionary processes of MPGA are carried out by using GEATbx (Polheim [7]). GEATbx is the ability of running with multiple populations which is systematically structured for adequately the execution of various evolutionary processes and rich in options regarding different genetic operators and their related parameters for real-valued variables. Considering the main elements of BGAwEIS, some elements of GEATbx are appropriately activated in the implementation of MPGA procedure. The crucial evolutionary operators of MPGA are presented via a pseudocode in Algorithm 3.

The first step is the initialization of *SubPopNum* subpopulations. In the beginning stage, a single population with (*SubPopIndNum* $*$ *SubPopNum*) individuals to be settled to the subpopulations is created. Following this step, the fitness values are calculated by using the fitness functions. The fitness values are penalized if they violate the constraints (see (3) and (4)). The penalization values are added to the fitness values.

Ranking process governed by ranking parameter Par_{rank} (see related parameters in Section 5) is based on the redistribution of fitness values where artificial values are used instead of the actual fitness values. In this way, dominance of the best solutions on the other solutions is weakened. Ranking process is carried out in two separate stages (Bäck and Hoffmeister [92]). In the first stage, the fitness values are recreated by the linear or nonlinear scaling functions. In the second stage, scaled fitness values are redistributed depending on the quality of actual fitness values. Furthermore, the ranking share procedure is applied where fitness values are rescaled according to their rank (Goldberg and Richardson [93]).

Following the ranking process, evolutionary approaches are repeatedly executed in a loop until a predefined loop number *epoch* is reached. The first inner loop is regarded

```
MPGA (SubPopNum, SubPopIndNum, SSR, NDV, NG = 1, Epoch, Par_rank, Par_mig, Par_comp,
Par_mut, Par_cr, Par_sel)
Initialize (P_i, SubPopNum, SubPopIndNum)
[P_i^d] = Decoding (P_i)
[F_i] = Fitness_Calculation (P_i^d)
[F_i] = Ranking(F_i, SubPopNum, SubPopIndNum, Par_rank)
for i = 1: Epoch
[P_i] = Subpopulation_Order_Evaluation(P_i, F_i, SubPopNum, SubPopIndNum)
        for j = 1: SubPopNum
[P_i] = SGA (P_i, NG, SubPopIndNum, F_i, Par_sel, Par_mut, Par_cr)
        End
[P_i] = Control (P_i, SSR, NDV)
[P_i^d] = Decoding (P_i)
[F_i] = Fitness_Calculation (P_i^d)
[F_i] = Ranking(P_i, SubPopNum, SubPopIndNum, Par_rank)
[P_i] = Competion_Process (P_i, Par_comp)
[P_i] = Migration_Process (P_i, Par_mig)
end
```

Algorithm 3: A Pseudocode for MPGA.

with determining the order of population. The subpopulations are ordered with respect to the fitness values of the individuals. For this purpose, a simple competition process based on ranking procedure is utilized (Polheim [7]). The rank of subpopulations has a big impact on the migration process because a communication network comprised of subpopulations is used for transmission of emigrants and immigrants.

After ordering subpopulation by taking into account the fitness values, SGA operators (selection *sel*, mutation *mut,* and crossover cr operators) and their related parameters Par_{sel}, Par_{mut}, and Par_{cr} (see Section 5) are activated. These three operators are separately executed for each subpopulation. The subsequent process is activated when the values of design variable exceed the ranges of SSR. If this occurs, then related individuals are repaired.

The competition process aims to move the robust individuals to other subpopulations that exhibit relatively poor performance (Schlierkamp-Voosen and Mühlenbein [94]). The competition of subpopulations is governed by parameter Par_{comp} (see Section 5) and carried out in three steps: (i) determination of the capacity of each subpopulations for taking emigrant and sending immigrant individuals, (ii) picking robust individuals according to their fitness values, and (iii) the adjustment of the subpopulations size for the settlement of the robust individuals (Polheim [7]).

The transmission of immigrants to the other subpopulations is accomplished by a migration process. The migration process is regulated with parameter Par_{mig} which indicates the several parameters, such as migration rate, interval, and topology (see Section 5).

5. Design Details

Due to the differentiation in parameters of the proposed algorithms, a number of parameter sets have to be tested to determine those with higher performance. The best way to accomplish this is to focus on their basic operators. In order to make an unbiased comparison among these proposed algorithms, the values of common operator parameters Par_{sel}, Par_{mut}, Par_{cr} and some evolutionary parameters SP and GN are kept constant for all algorithms. Operators of these algorithms and their related parameter values for each example problem are tabulated on Table 2. According to Table 2, while crossover rates indicate the number of individuals to be combined, mutation rates and ranges are used to define the number of variables per individuals to be mutated and the range of mutation steps for each variable, respectively. In addition, the selection operator, namely stochastic universal sampling, is able to run with any ranking method, namely, linear and nonlinear ranking using a ranking-related parameters Par_{rank}. In the approach of MPGA, the fitness assignment provided by the linear or nonlinear ranking method is assumed according to a certain value of its parameter, namely, selection pressure. Furthermore, the competition of subpopulations is governed by the parameters, competition interval and rate denoted by Par_{comp}. While competition interval determines the frequency of competition process, the number of migrated individuals with lower performance to be removed from the subpopulations is determined by the competition rate.

In the arrangement of operators, various parameter sets are proposed for each algorithm. BGAwEIS uses two basic parameters, namely, NGGES and NSBS. In order to investigate the relation between two parameters, the first parameter values are specified as "50, 20, 20, and 25," while the values of second parameter are fixed by "20, 50, 20, and 15." Thus, four parameter sets, namely (50, 20), (20, 50), (20, 20), and (25, 15), are devised for design tests.

MPGA is governed basically by migration related parameters Par_{mig} such as migration topology (MT), interval (MI), and rate (MR). The individuals with higher quality are migrated into five populations. In order to determine

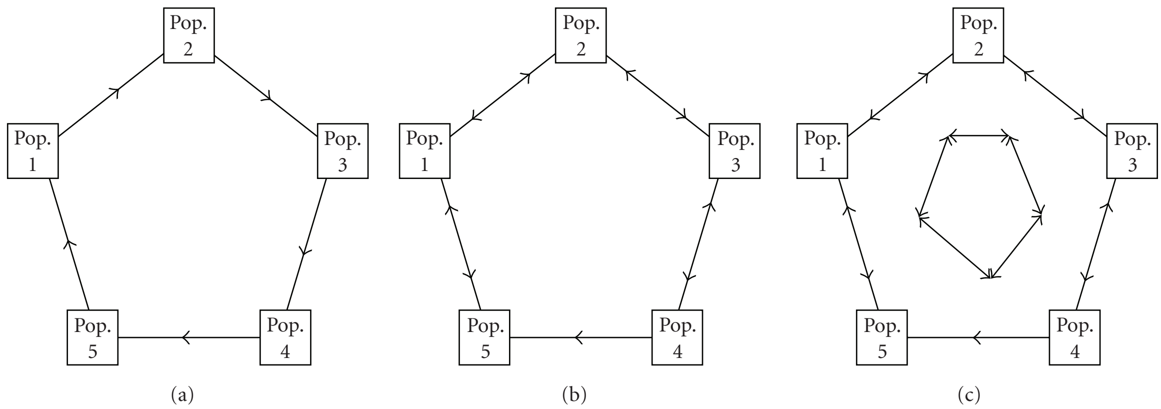


Figure 7: Introductions of the migration topologies for MPGA with five populations (Pop.) (a) ring-shaped topology, (b) neighborhood topology, and (c) unrestricted topology.

Table 2: Parameter set and related values proposed for SGA, BGAwEIS, and MPGA.

Algorithm name				SGA	BGAwEIS	MPGA				
Population no.				1	1[(b)]	1	2	3	4	5
Population size		Example 1		300	300	60	60	60	60	60
		Example 2		500	500	100	100	100	100	100
		Example 3		150	150	40	40	40	40	40
Operations	Variable type	Operator[(a)] name	Operator parameter name							
Selection	All variables	Stochastic universal sampling	Insertion rate	0.50	0.50	0.80	0.60	0.50	0.40	0.30
			Pressure	—	—	1.90	1.70	1.50	1.30	1.10
			Ranking method	—	—	NL[(c)]	NL	NL	L[(c)]	L
Crossover	Discrete variables	Single-point	Rate	0.80	0.80	0.90	0.70	0.50	0.30	0.10
	Continuous variables	Real type	Rate	0.80	0.80	0.90	0.70	0.50	0.30	0.10
Mutation	Discrete variables	Single-point	Rate	0.70	0.70	1.00	0.80	0.60	0.40	0.20
	Continuous variables	Real type	Rate	0.7	0.70	0.100	0.80	0.60	0.40	0.20
			Range	0.50	0.50	0.80	0.40	0.20	0.08	0.01
Competition	All variables	Competition of subpop.	Interval	—	—	20	20	20	20	20
			Rate	—	—	0.05	0.06	0.07	0.08	0.10
Generation gap	All variables	—	—	0.70	0.70	1.90	1.70	1.50	1.30	1.10

[(a)] See details in Polheim [7].
[(b)] Altough existing two populations within a core population, genetic operators are applied to the core population.
[(c)] NL: Nonlinear L: Linear.

the highest qualified parameter set through examining the parameter values, MI and MR are taken as “2, 10, 15, 5” and “0.10, 0.01, 0.10, 0.40”, while migration topologies are chosen either as unrestricted (denoted by 0) or of neighborhood type (denoted by 1) or ring shaped (denoted by 2) (see the depiction of proposed migration topologies for five populations (Pop) in Figure 7). Thus, a total of 48 parameter sets are tackled to assess the performance of MPGA.

6. Design Examples

The design examples are presented in the increasing order of complexity degree indicated by the number of truss bars and nodes. Three design examples with 24, 72, and 200 bars with one or two loading cases are employed for application of SGA, BGAwEIS, and MPGA. BGAwEIS and MPGA are compared considering their optimal designations obtained by using different parameter sets. The performance of SGA is evaluated with respect to its optimal designation generated by using one parameters set (see Table 2).

The dominant evaluation criteria will not only be the feasible solutions that form the basis of BGAwEIS’ control mechanism but some statistical measures are also included into the performance assessment. Besides, two interacting features of genetic search, exploration and exploitation, are utilized for the evaluation of proposed algorithms.

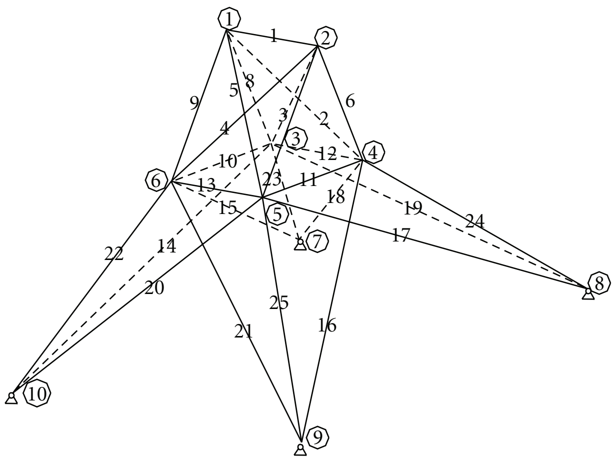


Figure 8: Geometry of spatial truss with 25-bars.

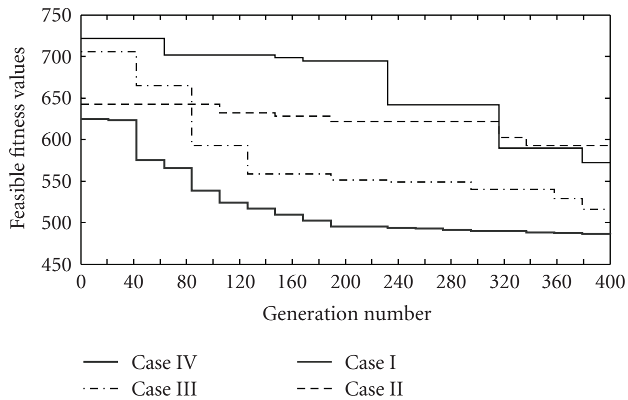


Figure 9: Convergence history of feasible solutions obtained by use of parameters sets proposed for BGAwEIS (spatial truss with 25-bars).

Exploration causes a random moving on the solution space, but exploitation involves an intensive search of promising solution region explored previously. In this regard, while exploration leads to a lower increase in fitness values, exploitation is responsible for a higher increase.

BGAwEIS is initially applied to observe the interdependence of its parameters with the output associated with three ratios:

Ratio 1 $R1$: (Size of Solution Region SSR)/(Number of Design Variables NDV),

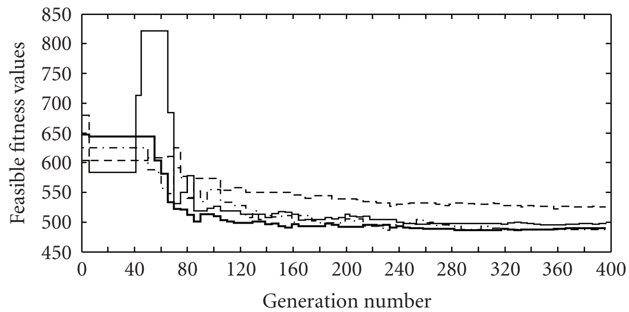
Ratio 2 $R2$: (number of generation *NG*)/(Number of Subsolution Regions before Search *NSBS*),

Ratio 3 $R3$: (Number of Generation *NG*)/(Number of Feasible Solution *NFS*).

While $R1$ is indicative about the quantities of feasible solutions, $R2$ and $R3$ are used to measure the performance of gradual exploration strategy. Moreover, $R1$ is important for specifying NG and SP. Optimal designations generated by use of four parameters sets are both tabulated for their output including corresponding statistical data and displayed for their convergence histories. Statistical data are computed by use of feasible solutions deserved to collect in feasible solution pool.

The optimal designations generated by MPGA are reported considering 48 parameter sets. The output data is both listed and displayed associated with feasible solutions obtained. For this purpose, the standard deviation and mean values of feasible solutions are computed. In order to comparatively present the designations, the parameter sets which achieve to obtain lower and higher quality of optimal designations are chosen among 48 parameter sets. Besides, activated frequencies of these parameter sets are also presented.

6.1. Design Example 1 (25-Bars Space Truss). This design problem is widely employed for the evaluation of various optimization methods (Figure 8). The members of space



	MT	MR	MI	Best	Mean	Std	Feas. num.
——	1	0.1	10	486.295	535.339	49.407	71
-·-	0	0.1	10	487.05	571.735	73.567	72
——	2	0.1	10	495.555	581.442	79.229	70
- - -	1	0.01	10	522.968	607.755	67.601	69

Figure 10: Convergence history of feasible solutions obtained by use of parameters sets proposed for MPGA (spatial truss with 25-bars).

truss linked in 8 groups are selected from a discrete set of 30 available sections (Table 3).

The design and evolutionary data for BGAwEIS (as an input and output obtained by use of four parameter sets) are listed on Table 3. The variation in feasible solutions corresponding to design variables is presented on Table 4. The convergence history of feasible fitness values are displayed in Figure 9.

From the tests of MPGA, the optimal designations obtained by use of 48 parameter sets are listed including their statistical analysis results (mean and standard deviations of feasible fitness values) in Table 5. The high and low values of these quantities are indicated by shaded boxes. Considering the parameter sets chosen, convergence history of feasible fitness values and activated frequencies of their populations are presented in Figures 10 and 11.

Table 3: Design and evolutionary data for BGAwEIS (spatial truss with 25-bars).

Design data					
Material properties					
Modulus of elasticity: 10^4 ksi					
Density of material: 0.1 lb/in.^{3}					
Loading data					
Case number	Joint number	X (kips)	Y (kips)	Z (kips)	
1	1	1	−10	−10	
	2	0	−10	−10	
	3	0.5	0	0	
	6	0.6	0	0	
Constraint data					
Displacement constraints: $U_k \leq 0.35$ inc $(k = 1, 2)$ for X and Y directions					
Stress constraints: $-40 \leq \sigma_i \leq 40$ ksi $(I = 1, \ldots, 25)$					
Elements of discrete sets and their position number for A_i $(I = 1, \ldots, 25)$					
0.1(1),0.2(2),0.3(3),0.4(4),0.5(5),0.6(6),0.7(7),0.8(8),0.9(9),1.0(10),1.1(11),1.2(12),1.3(13),1.4(14),1.5(15),1.6(16),1.7(17), 1.8(18),1.9(19),2.0(20),2.1(21),2.2(22),2.3(23),2.4(24),2.5(25),2.6(26),2.8(27), 3.0(28),3.2(29),3.4(30)					
Evolutionary data					
Input					
Number of design variables: 8					
Size of solution region: 30					
Number of generation: 400					
Size of inward population: 300					
Size of outward population: 300					
Size of core population: 300					
		Cases			
		Case I	Case II	Case III	Case IV
	NGGES	50	20	20	25
	NSBS	20	50	20	15
Output					
	NSAS	13	34	1	1
	NFS	7	6	10	18
	Ratio 1 *R*1	3.75	3.75	3.75	3.75
	Ratio 2 *R*2	20	8	20	27
	Ratio 3 *R*3	57	67	40	22
Best feasible fitness value		571.618	592.656	515.845	485.90
Mean of feasible fitness values		659.771	619.972	587.133	521.678
Standard deviation of feasible fitness values		59.591	18.856	67.096	45.880

The optimal designations obtained by proposed algorithms and existing ones outlined in literature are presented in Table 6 including the critical values of stress and displacement corresponding to the optimal designations.

6.2. Design Example 2 (72-Bars Spatial Truss). The transmission tower with 72 members is also used by many researchers as a benchmark problem. This steel structure has 16 independent design variables and subjected to two different loading conditions (Figure 12).

The design and evolutionary data for BGAwEIS (as an input and output obtained by use of four parameter sets) are listed on Table 7. The variation of feasible solutions values through generation number are displayed in Figure 13.

Optimal designations generated by use of 48 parameter set for MPGA are tabulated including their statistical analysis results (mean and standard deviations of feasible fitness values) (Table 8). High and low values of these quantities are indicated by shaded boxes. The convergence history of feasible fitness values corresponding to these parameter sets and activated frequencies of their populations are shown in Figures 14 and 15. The optimal designations obtained by proposed algorithms and existing methods outlined in the literature are reported in Table 9 including the critical values of stress and displacement corresponding to the optimal designations.

Table 4: Variation of feasible fitness values according to design variables for spatial truss with 25-bars.

Fitness values	Design variable groups						
	1	2–5	6–9	10-11	12-13	14–17	18–21	22–25
	Feasible design variable values obtained from different generations							
624.71	7	29	20	5	5	24	1	29
623.60	28	20	29	26	10	12	11	26
575.49	27	14	28	7	18	15	5	29
565.82	22	17	26	20	17	10	7	29
538.74	24	9	30	2	21	13	4	29
523.87	1	1	29	5	17	10	12	30
516.84	1	10	29	3	12	10	7	30
509.60	1	10	29	3	12	9	7	30
502.30	1	10	29	3	12	9	7	30
495.11	1	10	29	3	12	9	5	30
493.80	1	12	29	1	11	9	4	30
492.63	1	2	30	2	19	10	7	30
491.13	1	2	30	2	18	10	7	30
489.63	1	2	30	2	17	10	7	30
488.13	1	2	30	2	16	10	7	30
487.41	1	1	30	1	20	10	7	30
486.63	1	2	30	1	16	10	7	30
485.9	**1**	**1**	**30**	**1**	**19**	**10**	**7**	**30**

Table 5: Statistical analysis results of feasible fitness values obtained by use of parameter sets proposed for MPGA (spatial truss with 25-bars).

Parameter set	Best	Mean	Std	Parameter set	Best	Mean	Std
MT=0, MR=0.01, MI=2	502,104	577,038	67,707	MT=1, MR=0.10, MI=2	497,679	553,204	52,961
MT=0, MR=0.01, MI=10	491,290	572,765	73,076	**MT=1, MR=0.10, MI=10**	**486,295**	**535,339**	**49,407**
MT=0, MR=0.01, MI=15	510,594	594,444	68,822	MT=1, MR=0.10, MI=15	492,001	564,707	61,902
MT=0, MR=0.01, MI=5	501,384	589,606	68,165	MT=1, MR=0.10, MI=5	490,972	582,356	73,748
MT=0, MR=0.05, MI=2	493,103	553,969	57,707	MT=1, MR=0.40, MI=2	493,464	559,504	54,353
MT=0, MR=0.05, MI=10	495,631	564,716	59,424	MT=1, MR=0.40, MI=10	494,279	568,882	63,133
MT=0, MR=0.05, MI=15	493,015	558,278	48,711	MT=1, MR=0.40, MI=15	492,607	564,306	71,255
MT=0, MR=0.05, MI=5	492,631	559,883	58,218	MT=1, MR=0.40, MI=5	488,096	561,872	54,620
MT=0, MR=0.10, MI=2	495,242	579,444	67,091	MT=2, MR=0.01, MI=2	510,554	602,508	77,442
MT=0, MR=0.10, MI=10	**487,050**	**571,735**	**73,567**	MT=2, MR=0.01, MI=10	488,320	574,907	72,809
MT=0, MR=0.10, MI=15	504,839	585,483	63,716	MT=2, MR=0.01, MI=15	505,944	589,977	68,911
MT=0, MR=0.10, MI=5	494,356	581,330	64,624	MT=2, MR=0.01, MI=5	512,587	578,087	59,429
MT=0, MR=0.40, MI=2	491,887	546,261	54,684	MT=2, MR=0.05, MI=2	493,299	574,085	44,792
MT=0, MR=0.40, MI=10	496,481	570,750	54,856	MT=2, MR=0.05, MI=10	489,609	551,748	63,178
MT=0, MR=0.40, MI=15	511,631	583,177	65,891	MT=2, MR=0.05, MI=15	493,907	558,576	53,586
MT=0, MR=0.40, MI=5	488,049	549,874	61,681	MT=2, MR=0.05, MI=5	493,523	563,007	60,136
MT=1, MR=0.01, MI=2	520,044	583,861	70,871	MT=2, MR=0.10, MI=2	491,125	572,506	60,291
MT=1, MR=0.01, MI=10	**522,371**	**607,755**	**67,601**	**MT=2, MR=0.10, MI=10**	**495,555**	**581,442**	**79,229**
MT=1, MR=0.01, MI=15	506,457	582,052	57,692	MT=2, MR=0.10, MI=15	492,714	570,818	68,788
MT=1, MR=0.01, MI=5	506,226	602,789	60,236	MT=2, MR=0.10, MI=5	511,718	584,622	60,603
MT=1, MR=0.05, MI=2	489,218	580,434	75,958	MT=2, MR=0.40, MI=2	498,955	561,760	50,194
MT=1, MR=0.05, MI=10	507,234	556,624	53,246	MT=2, MR=0.40, MI=10	492,850	574,766	63,040
MT=1, MR=0.05, MI=15	500,155	562,187	53,653	MT=2, MR=0.40, MI=15	491,468	566,396	69,870
MT=1, MR=0.05, MI=5	491,397	561,556	56,276	MT=2, MR=0.40, MI=5	501,608	577,688	51,228

Table 6: Comparison of optimum designs, critical deflection, and stress values for BGAwEIS (spatial truss with 25-bars).

Ref.	Best weight	Design variable groups							
		1	2–5	6–9	10-11	12-13	14–17	18–21	22–25
Rajeev and Krishnam. [95]	546.01	0.10	1.80	2.30	0.20	0.10	0.80	1.80	3.00
Zhu [96]	562.93	0.10	1.90	2.60	0.10	0.10	0.80	2.10	2.60
Erbatur et al. [97]	493.80	0.10	1.20	3.20	0.10	1.10	0.90	0.40	3.40
**Coello et al. [98]	493.94	—	—	—	—	—	—	—	—
**Wu and Chow [99]	491.72	—	—	—	—	—	—	—	—
SGA	814.64	0.10	3.00	2.80	2.40	2.20	1.90	2.80	2.40
MPGA	486.29	0.10	0.50	3.40	0.10	1.50	0.90	0.60	3.40
BGAwEIS	**485.90**	**0.10**	**0.10**	**3.40**	**0.10**	**1.90**	**1.00**	**0.70**	**3.40**
Max. displacement. in x, y and z directions: 0.1206, 0.3498, 0.0462 at node1									
Max. element stress: 20.2311 at element 25									

** Design variable groups are not presented in the references.

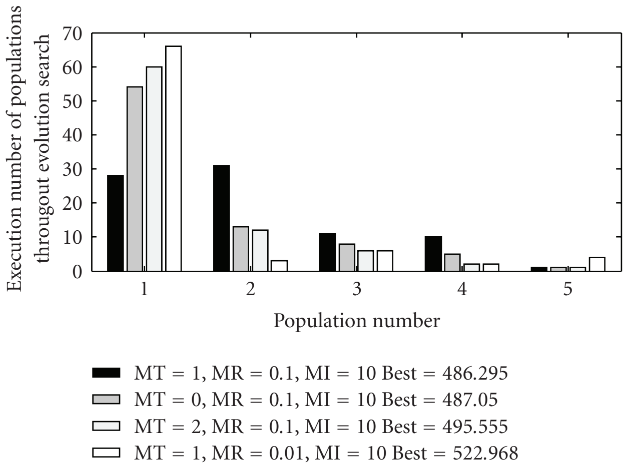


Figure 11: Activated numbers of each population obtained by MPGA (spatial truss with 25-bars).

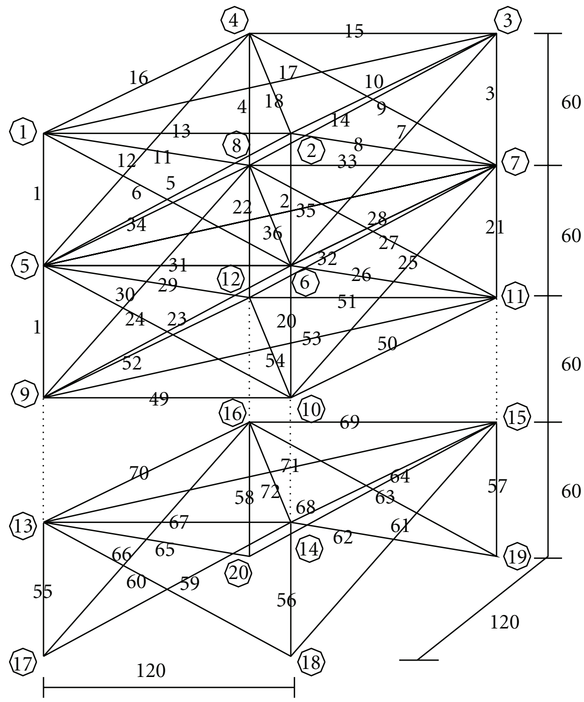


Figure 12: Geometry of the spatial truss with 72-bars.

6.3. Design Example 3 (200-Bars Planar Truss). The plane truss shown in Figure 8 involves both continuous as well as discrete design variables (Ponterosso and Fox [104]). It has 200 independent design variables (Figure 16).

The design and evolutionary data for BGAwEIS (as an input and output obtained by four parameter sets) are listed on Table 10. The variation of feasible fitness values through generation number are shown in Figure 17.

Optimal designations obtained by MPGA, considering 48 parameter sets are summarized including statistical analysis results (mean and standard deviations of feasible fitness values) (Table 11). The convergence history of feasible fitness values obtained by use of these parameter sets chosen and activated frequencies of their populations are displayed in Figures 18 and 19. The optimal designations with higher performance are presented for proposed algorithms and existing approaches outlined in literature in Table 12 including the critical values of stress and displacement corresponding to the optimal designations. Design variables that belong to optimal designation obtained by BGAwEIS are presented in the appendix.

7. Discussion

In this section, BGAwEIS, MPGA, and SGA are evaluated, considering the effect of different parameter sets on the quality degree of optimal designations and then their performance is investigated taking into account the exploration and exploitation features of genetic search. However, due to fact that evolutionary parameters of SGA are fixed for design examples, the evaluation of SGA is skipped here.

Table 7: Design and evolutionary data for BGAwEIS (spatial truss with 72-bars).

Design data					
Material properties					
Modulus of elasticity: 10^4 ksi					
Density of material: 0.1 lb/in.^{3}					
Loading data					
Case number	Joint number	X (kips)	Y (kips)	Z (kips)	
1	1	5	5	−5	
2	1	0	0	−5	
	2	0	0	−5	
	3	0	0	−5	
	4	0	0	−5	
Constraint data					
Displacement constraints: $U_k \leq 0.25$ inc ($k = 1, \ldots, 20$) for X and Y directions					
Stress constraints: $-25 \leq \sigma_i \leq 25$ ksi ($i = 1, \ldots, 72$)					
Elements of continuous sets					
$0.1 \leq A_i \leq 2$ ($i = 1, \ldots, 72$)					
Evolutionary data					
Input					
Number of design variables: 16					
Size of solution region: ∞					
Number of generation: 600					
Size of inward population: 500					
Size of outward population: 500					
Size of core population: 500					
		Cases			
		Case I	Case II	Case III	Case IV
	NGGES	50	20	40	15
	NSBS	20	50	15	50
Output					
	NSAS	8	35	2	23
	NFS	8	7	11	12
	Ratio 1 $R1$	∞	∞	∞	∞
	Ratio 2 $R2$	30	12	40	12
	Ratio 3 $R3$	75	85	55	50
Best feasible fitness value		427.753	452.286	380.784	381.08
Mean of feasible fitness values		694.574	842.722	891.234	882.993
Standard deviation of feasible fitness values		290.294	329.609	409.814	351.210

For ease of presentation, the values of the parameters discussed below are presented using a vector-like notation like (•, •, •) where the 1st, 2nd, and 3rd value within the parentheses correspond to examples 1, 2, and 3, respectively.

7.1. Consideration of Variation in the Values of Evolutionary Parameters of GAwEIS and MPGA

Regarding with BGAwEIS. (i) Generation number and population size are proportional to $R1$. In this work, $R1$ has the values (3.75, ∞, 0.15) (Tables 3, 7, and 10). Corresponding to $R1$ value set, generation number is specified as (400, 600, 200), and population size as (300, 500, 150).

(ii) There is a direct proportionality between R1 and output NFS computed using the feasible solution pool (Tables 3, 7, and 10). For example, considering the parameter sets with higher performance, the values of $R1$ corresponding to the value of the output NFS set of (18, 11, 19) are equal to (3.75, ∞, 0.15).

(iii) The best optimal designations are obtained when using parameters NGGES and NSBS set (25, 15), (40, 15), and (40, 5) for each design example.

The gradual exploration strategy is activated by parameter NGGES. The parameters NSBS and NSAS are indicative of the activated frequency of gradual exploration strategy (Tables 3, 7, and 10). Considering parameter value sets

Table 8: Statistical analysis results of feasible fitness values obtained by use of parameter sets proposed for MPGA (spatial truss with 72-bars).

Parameter set	Best	Mean	Std	Parameter set	Best	Mean	Std
MT=0, MR=0.01, MI=2	640,229	907,646	147,346	MT=1, MR=0.10, MI=2	683,537	912,827	109,804
MT=0, MR=0.01, MI=10	734,629	913,598	113,454	**MT=1, MR=0.10, MI=10**	**594,811**	**951,129**	**119,593**
MT=0, MR=0.01, MI=15	711,954	913,401	126,805	MT=1, MR=0.10, MI=15	598,955	952,984	124,680
MT=0, MR=0.01, MI=5	695,537	922,956	113,759	MT=1, MR=0.10 MI=5	753,755	943,440	110,412
MT=0, MR=0.05, MI=2	**767,134**	**951,538**	**123,862**	MT=1, MR=0.40, MI=2	618,419	890,768	140,968
MT=0, MR=0.05, MI=10	760,033	917,419	107,376	MT=1, MR=0.40, MI=10	693,387	935,066	116,045
MT=0, MR=0.05, MI=15	633,658	935,527	136,514	MT=1, MR=0.40, MI=15	603,390	871,091	137,827
MT=0, MR=0.05, MI=5	723,858	898,141	114,732	MT=1, MR=0.40, MI=5	717,659	948,091	118,160
MT=0, MR=0.10, MI=2	615,721	926,908	105,704	MT=2, MR=0.01, MI=2	620,216	893,702	146,986
MT=0, MR=0.10, MI=10	689,895	891,536	148,994	MT=2, MR=0.01, MI=10	622,736	902,485	148,753
MT=0, MR=0.10, MI=15	732,047	934,492	113,616	MT=2, MR=0.01, MI=15	644,256	929,181	129,546
MT=0, MR=0.10, MI=5	696,988	904,547	115,401	MT=2, MR=0.01, MI=5	656,687	921,091	119,474
MT=0, MR=0.40, MI=2	704,639	915,372	111,372	MT=2, MR=0.05, MI=2	731,287	922,980	121,985
MT=0, MR=0.40, MI=10	673,424	874,583	125,405	MT=2, MR=0.05, MI=10	729,340	941,674	116,160
MT=0, MR=0.40, MI=15	678,748	890,891	127,018	MT=2, MR=0.05, MI=15	618,836	877,379	138,408
MT=0, MR=0.40, MI=5	682,176	916,081	127,354	**MT=2, MR=0.05, MI=5**	**625,641**	**885,707**	**151,493**
MT=1, MR=0.01, MI=2	669,352	943,534	122,819	MT=2, MR=0.10, MI=2	690,889	867,153	149,447
MT=1, MR=0.01, MI=10	604,188	917,024	136,592	**MT=2, MR=0.10, MI=10**	**695,986**	**920,233**	**100,479**
MT=1, MR=0.01, MI=15	621,338	882,202	134,963	MT=2, MR=0.10, MI=15	697,875	909,407	108,374
MT=1, MR=0.01, MI=5	667,580	895,065	133,153	MT=2, MR=0.10, MI=5	638,491	870,591	133,380
MT=1, MR=0.05, MI=2	735,599	945,279	113,778	MT=2, MR=0.40, MI=2	706,990	943,478	133,583
MT=1, MR=0.05, MI=10	625,002	902,000	121,989	MT=2, MR=0.40, MI=10	718,084	939,940	130,951
MT=1, MR=0.05, MI=15	673,220	923,570	114,286	MT=2, MR=0.40, MI=15	698,415	906,278	118,242
MT=1, MR=0.05, MI=5	679,263	932,444	122,463	MT=2, MR=0.40, MI=5	695,139	944,974	125,174

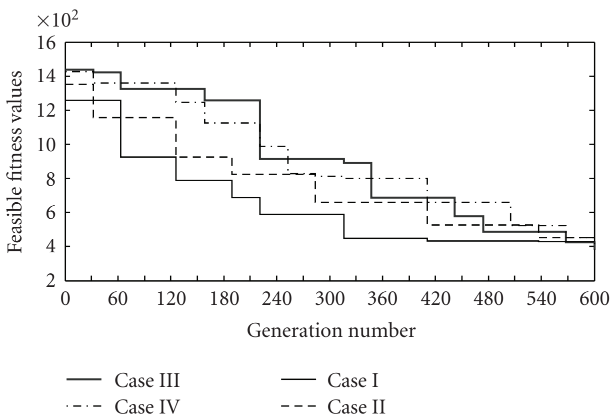
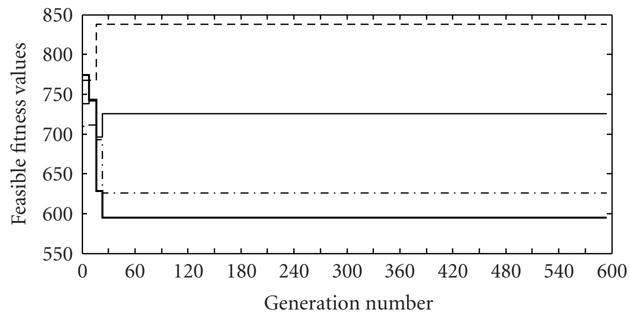


Figure 13: Convergence history of feasible solutions obtained by use of parameters sets proposed for BGAwEIS (spatial truss with 72-Bars).



	MI	MR	MJ	Best	Mean	Std	Feas. num.
——	1	0.1	10	594.811	951.129	119.593	4
-·-·	2	0.05	5	625.641	885.707	151.493	4
——	2	0.1	10	695.986	920.233	100.479	4
- - -	0	0.05	2	767.134	951.538	123.862	3

Figure 14: Convergence history of feasible solutions obtained by use of parameters sets proposed for MPGA (spatial truss with 72-bars).

with better performance, the value set of output NSAS corresponding to the value set of parameter NGGES (25, 40, 40) are (1, 2, 1). The values of parameter NSBS are gradually decreased and reach the value of parameter NSAS eventually. This shows that evolutionary search is successfully completed after gradually enlarging of the bounds of subsolutions regions. Moreover, $R2$ also indicates about the activated frequency of the gradual exploration strategy. The output NFS is proportionally increased by the activation of gradual exploration strategy. In this regard, if $R2$ is close to or higher than $R3$, then the gradual exploration strategy is successfully applied, that is, a feasible solution is obtained once the

Table 9: Comparison of optimum designs, critical deflection, and stress values for BGAwEIS (spatial truss with 72-bars).

Design variables	References							
	Venkaya [100]	Gellatly and Berke [101]	Renwei and Peng [102]	Schmit and Farshi [103]	Erbatur et al. [97]	SGA	MPGA	BGAwEIS
1–4	0.161	0.1492	0.1641	0.1585	0.161	0.873	0.675	**0,156**
5–12	0.557	0.7733	0.5552	0.5936	0.544	1.681	0.253	**0,555**
13–16	0.377	0.4534	0.4187	0.3414	0.379	0.100	0.601	**0,370**
17-18	0.506	0.3417	0.5758	0.6076	0.521	1.418	0.437	**0,510**
19–22	0.611	0.5521	0.5327	0.2643	0.535	0.986	0.841	**0,620**
23–30	0.532	0.6084	0.5256	0.5480	0.535	1.530	0.861	**0,530**
31–34	0.100	0.100	0.100	0.100	0.103	1.982	0.460	**0,100**
35-36	0.100	0.100	0.100	0.1509	0.111	1.121	1.513	**0,100**
37–40	1.246	1.0235	1.2893	1.1067	1.310	1.589	1.910	**1,250**
41–48	0.524	0.5421	0.5201	0.5793	0.498	1.987	0.789	**0,523**
49–52	0.100	0.100	0.100	0.100	0.110	1.083	0.132	**0,101**
53-54	0.100	0.100	0.100	0.100	0.103	1.856	0.936	**0,105**
55–58	1.818	1.464	1.9173	2.0784	1.910	0.268	1.840	**1,860**
59–66	0.524	0.5207	0.5207	0.5034	0.525	1.473	0.899	**0,513**
67–70	0.100	0.100	0.100	0.100	0.122	0.849	0.244	**0,100**
17–72	0.100	0.100	0.100	0.100	0.103	1.469	0.183	**0,100**
Best Weight	381.28	395.97	379.66	388.65	383.120	1196.89	594.811	**380.783**
0.0091, 0.0091, 0.2391 at node1 for Case 1; Max. displacement. in *x*, *y* and *z* directions: 0.2499, 0.2499, 0.0718 at node1 for Case 2								
Max. element stress: 16.2519 at element 1 for Case 1; 24.9371 at element 1 for Case 2								

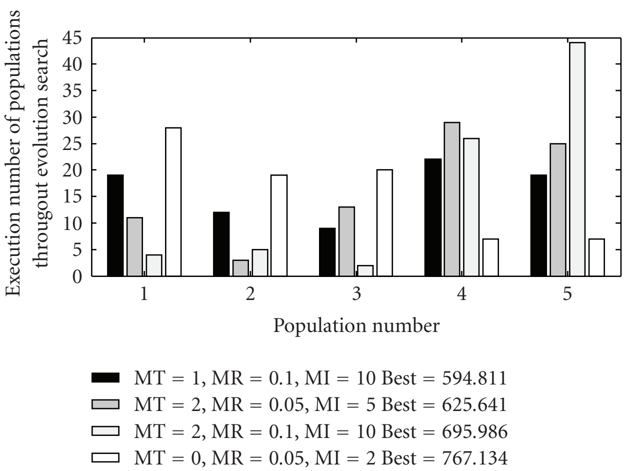


Figure 15: Activated numbers of each population obtained by MPGA (spatial truss with 72-bars).

bounds of subsolution regions are enlarged. For example, an $R2$ value set (27, 40, 40) for Cases IV, III, and III of design examples 1, 2, and 3 corresponds to an $R3$ value set (22, 55, 11). In design example 2, although the value of $R2$ is lower than $R3$, the value of output NSAS is obtained as 2. This indicates that the bounds of subsolution regions can be further enlarged.

Considering the convergence history of feasible fitness values corresponding to the four cases of each design example, the success of parameter sets is also confirmed by consistently decreased trend lines in Figures 9, 13, and 17.

Regarding with MPGA. (i) In the sensitivity analysis of basic parameters of MPGA, a total of 48 parameter sets composed of various values of parameters MT, MI, and MR are considered. The parameter values with higher performance for each design examples are indicated by a dark-shaded box and obtained as (1, 0.10, 10), (1, 0.10, 10), and (0, 0.05, 5), each of which is denoted by MT, MI, and MR, respectively (see Tables 5, 8, and 11). These results show that migration interval and rates varies proportionally with the generation numbers and population size. For example, the migration interval (10, 10, 5) increases with the generation number (400, 600, 200). This indicates that the rate of migration interval to generation numbers varies within a range of (0.015 (or 10/600)–0.025 (or 5/200)) or % (1.5–2.5). Migration rate varies within a range % (0.05–0.10) of SP. It appears that the number of migrated individuals is increased with the population size. For example, the number of migrated individuals is (30, 50, 10) where the migration

Table 10: Design and evolutionary data for BGAwEIS (planar truss with 200-bars).

The design data				
Material properties				
Modulus of elasticity: 30×10^3 ksi Density of material: 0.283 lb/in.^{3}				
Loading data				
Case number	Joint number	X (kips)	Y (kips)	Z (kips)
1	1,6,15,20,29,34,43,48,57,62,71	1	0	0
	1,2,3,4,5,6,8,10,12,14,15,16,17,18,19,71,72,73,74,75	0	−10	0
Constraint data				
Displacement constraints: $U_k \leq 0.50$ inc ($k = 1, \ldots, 77$) for X and Y directions Stress constraints: $-30 \leq \sigma_i \leq 30$ ksi ($i = 1, \ldots, 200$)				
Elements of discrete sets and their position number for A_i *(*$I = 1, \ldots, 200$*)*				
0.100(1),0.347(2),0.440(3),0.539(4),0.954(5),1.081(6),1.174(7),1.333(8),1.488(9),1.764(10),2.142(11),2.697(12),2.800(13),3.131(14),3.565(15),3.813(16),4.805(17),5.952(18),6.572(19),7.192(20),8.525(21),9.300(22),10.850(23),13.330(24),14.290(25),17.170 (26),19.180(27),23.680(28),28.080(29),33.700(30)				
Evolutionary data				
Input				
Number of design variables: 150 Size of solution region: 30 Number of generation: 200 Size of inward population: 150 Size of outward population: 150 Size of core population: 150				
	Cases			
	Case I	Case II	Case III	Case IV
NGGES	50	20	40	15
NSBS	20	50	5	15
Output				
NSAS	17	42	1	1
NFS	5	6	19	16
Ratio 1 R1	0.15	0.15	0.15	0.15
Ratio 2 R2	10	4	40	13
Ratio 3 R3	40	33	11	13
Best feasible fitness value	37659.683	39480.881	33405.949	35128.000
Mean of feasible fitness values	146652.109	130204.259	182974.671	153277.013
Standard deviation of feasible fitness values	98588.652	88555.569	87725.056	59388.735

rates are (0.10, 0.10, 0.05) and population sizes (300, 500, 200). The migration topologies referred to as unrestricted and dominantly neighborhood perform well.

(ii) Considering the lower and upper values of statistical data of results obtained by parameter sets, several parameter sets are chosen and indicated by shaded boxes (see Tables 5, 8, and 11). The convergence history of feasible fitness values corresponding to parameter sets chosen are shown in Figures 10, 14, and 18. The trend lines are consistently decreased for design example 1, but inconsistently for design example 2 and 3. Particularly, it is observed that the feasible solutions are generated in the beginning of evolutionary search. Then, generation of feasible solutions stagnates at the remaining generation numbers in design examples 2 and 3.

(iii) The activated numbers of populations obtained by use of parameter sets chosen are shown by bars in Figures 11, 15, and 19. From these figures, it is obvious that the distribution of activated numbers of populations corresponding to these parameter sets with higher performance is more regular.

7.2. Performance Investigation of BGAwEIS and MPGA in Design Optimization. Considering the various parameter

Table 11: Statistical analysis results of feasible fitness values obtained by use of parameter sets proposed for MPGA (planar truss with 200-bars).

Parameter set	Best	Mean	Std	Parameter set	Best	Mean	Std
MT=0, MR=0.01, MI=2	**72633,880**	**80630,569**	**8571,450**	MT=1, MR=0.10, MI=2	43405,949	46677,338	6914,943
MT=0, MR=0.01, MI=10	48093,574	56656,419	4216,594	MT=1, MR=0.10, MI=10	46110,341	52166,191	4860,834
MT=0, MR=0.01, MI=15	46658,763	53303,681	4797,318	MT=1, MR=0.10, MI=15	46797,205	53567,554	6498,843
MT=0, MR=0.01, MI=5	43563,417	49063,016	3737,226	MT=1, MR=0.10 MI=5	45121,767	54525,702	6203,645
MT=0, MR=0.05, MI=2	47659,683	53501,963	4633,370	MT=1, MR=0.40, MI=2	44316,220	48637,425	3729,784
MT=0, MR=0.05, MI=10	44562,844	52044,266	6328,921	MT=1, MR=0.40, MI=10	43141,914	48152,488	5033,876
MT=0, MR=0.05, MI=15	48956,630	55827,141	5594,529	MT=1, MR=0.40, MI=15	46041,416	54705,672	4620,251
MT=0, MR=0.05, MI=5	**40079,507**	**46459,547**	**5301,405**	MT=1, MR=0.40, MI=5	48661,997	54604,399	3207,818
MT=0, MR=0.10, MI=2	40355,654	46811,317	4671,882	MT=2, MR=0.01, MI=2	44005,459	52233,311	6203,747
MT=0, MR=0.10, MI=10	46765,127	55515,587	4599,817	MT=2, MR=0.01, MI=10	50810,975	57383,192	4752,537
MT=0, MR=0.10, MI=15	47207,684	52014,078	4626,456	MT=2, MR=0.01, MI=15	44926,934	52888,602	3982,791
MT=0, MR=0.10, MI=5	56928,673	60909,024	3266,486	MT=2, MR=0.01, MI=5	43645,028	50901,106	5238,565
MT=0, MR=0.40, MI=2	41540,470	46590,198	4894,845	MT=2, MR=0.05, MI=2	49959,857	55518,284	5472,687
MT=0, MR=0.40 MI=10	45410,658	53645,716	4960,128	MT=2, MR=0.05, MI=10	51736,329	62136,457	6177,860
MT=0, MR=0.40, MI=15	49511,159	53964,220	3839,889	MT=2, MR=0.05, MI=15	48048,719	57350,748	5234,649
MT=0, MR=0.40, MI=5	49983,427	53905,603	3265,335	MT=2, MR=0.05, MI=5	43096,147	50628,687	7155,731
MT=1, MR=0.01, MI=2	**63494,763**	**64043,462**	**350,856**	MT=2, MR=0.10, MI=2	70763,445	73702,854	2774,557
MT=1, MR=0.01, MI=10	58124,138	62066,360	4194,507	MT=2, MR=0.10, MI=10	46612,625	51832,507	4416,433
MT=1, MR=0.01, MI=15	53912,544	58640,574	4411,781	MT=2, MR=0.10, MI=15	45523,323	50852,676	4438,879
MT=1, MR=0.01, MI=5	43427,511	48630,306	5504,416	MT=2, MR=0.10, MI=5	52091,564	57823,182	3014,238
MT=1, MR=0.05, MI=2	49480,881	49870,624	5751,019	MT=2, MR=0.40, MI=2	48312,173	54232,383	6997,833
MT=1, MR=0.05, MI=10	41201,483	48559,551	5957,630	MT=2, MR=0.40, MI=10	46408,771	52875,303	3739,149
MT=1, MR=0.05, MI=15	44122,014	49935,079	3517,734	MT=2, MR=0.40, MI=15	46437,702	56239,893	6986,506
MT=1, MR=0.05, MI=5	43731,886	51337,650	6528,667	MT=2, MR=0.40, MI=5	43413,187	53022,954	5995,418

Table 12: Comparison of optimum designs, critical deflection, and stress values for BGAwEIS (planar truss with 200-bars).

	References			
	Ponterosso and Fox [104]	SGA	MPGA	**BGAwEIS**
Minimum weight	35394.00	122047.14	40079.507	**33405.949**
Max. displacement. in x, y and z directions: 0.4959, 0.2745, 0.4998 at node1 for Case 1; 0.0948, 0.4738, 0.0428 at node1 for Case 2				
Max. element Stress: 20.9676 at element 90 for Case 1; 6.4872 at element 185 for Case 2				

sets proposed for BGAwEIS and MPGA, a parameter set with high performance is determined for each algorithm. The results obtained with these parameter sets are to be examined according to exploration and exploitation features of genetic search discussed previously and the quality of existing optimal solutions outlined in literature.

(i) The exploration and exploitation features of genetic search cause a lower and higher increase in the fitness values, respectively. This is easily confirmed for BGAwEIS by observing the change in fitness values obtained for design example 1 (see Table 4). While the difference between the first and second feasible fitness values is equal to 1.11 (or 624.71–623.60), it increases to 27.08 (or 565.82–538.74) for the fourth and fifth fitness values. This issue is also observed in Figure 9, considering the trend lines corresponding to the best parameter set denoted by Case IV. The exploration is dominant both in the beginning and towards the end of search (after generation number 200). The exploitation becomes dominant within a certain interval between generation numbers 40 and 200. A balance between exploration and exploitation is relatively established for Case III in design example 2 (Figure 13). In Case III of design example 3, exploration is dominant in the beginning of evolutionary search, but then exploitation begins to control the evolutionary search (Figure 17). Starting with a lower fitness values for the first generation causes a decrease in the mean and standard deviation of feasible fitness values (see Figure 9, 13, and 17 along with Tables 3, 7, and 10).

MPGA is managed by a migration dominated evolutionary process. Considering the parameter sets with high

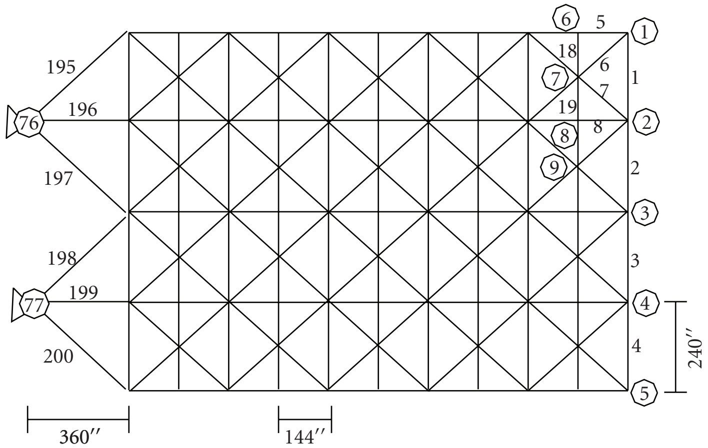


Figure 16: Geometry of planar truss with 200-bars.

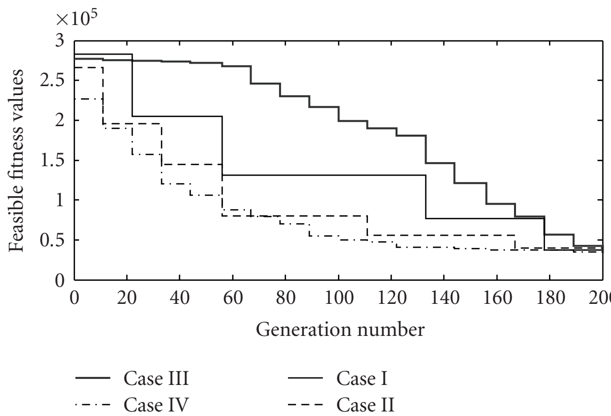


Figure 17: Convergence history of feasible solutions obtained by use of parameters sets proposed for BGAwEIS (planar truss with 200-bars).

performance, the dominancy of exploration and exploitation is consistently observed at different generation numbers in design example 1 that has a small number of bars and nodes, but variably for design examples 2 and 3 where an increased number of bars and nodes are present. Especially, evolutionary search ends up with stagnation while searching the feasible solutions in design example 2 and 3 (see Figures 10, 14, and 18). As in BGAwEIS, the initialization of evolutionary search with higher fitness values causes an increase in the mean and standard deviations of feasible fitness values (see Figures 10, 14, and 18 along with Tables 5, 8, and 11).

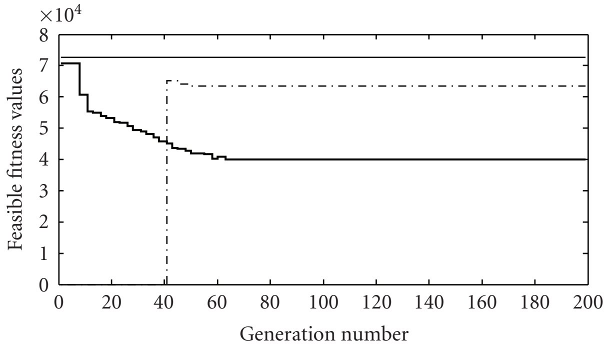
(ii) Investigating the optimal designations obtained by BGAwEIS, MPGA, SGA, and existing solution methods outlined in literature, it can be said that BGAwEIS is more efficient in improving the quality of optimal designations (see Tables 6, 9, and 12). Particularly, in order to cope with the complexity arising from the increase in the truss elements and nodes, the values of parameter NSBS associated with the value of parameters NGGES and NG are elevated. MPGA improves the quality of optimal designations using unrestricted and dominantly neighborhood migration topologies along with a migration interval about % (1.5–2.5) of NG and a migration rate about % (0.05–0.10) of SP.

8. Conculusion

In this work, a new genetic algorithm method, namely, (BGAwEIS) is presented to be used with the design optimization of pin-jointed structures. In order to evaluate the capability and efficiency of BGAwEIS, the optimal designations obtained by SGA and solution methods outlined in literature are not only examined but also an MPGA is proposed to assess the influence of multiple populations on the quality of optimal designations. The tests are performed on three design examples having 25, 72, and 200 bars. The following conclusions are drawn from the results of design examples considered.

(i) It is shown that bipopulation approach proposed by BGAwEIS achieves effective usage of exploration and exploitation features of genetic search simultaneously compared to MPGA with multiple populations. Particularly, it is shown that the gradual exploration strategy has a significant impact on BGAwEIS' performance causing an increase in the values of NSBS with respect to NGGES and NG. It is displayed that MPGA is able to improve quality of its optimal designations by use of migration topologies called unrestricted and dominantly neighborhood along with a migration interval about % (1.5–2.5) of generation numbers and a migration rate about % (0.05–0.10) of population size. Furthermore, the activated numbers of populations obtained by use of these parameter sets are shown to be more homogeneous compared to other ones.

(ii) Although it is shown that MPGA is successful in providing an equal distribution of activated frequencies for each population, it has difficulties in directing the evolutionary search for exploration of new solution regions because purely using the migration process causes the certain individuals to be dominant during evolutionary search. This negativity leads to stagnation on the generation of promising



	MI	MR	MJ	Best	Mean	Std	Feas. num.
——	0	0.05	5	40079.507	46549.547	5301.405	24
-·-·-	1	0.01	2	63494.763	64043.462	350.856	5
——	0	0.01	2	72633.88	80630.569	8571.45	1

Figure 18: Convergence history of feasible solutions obtained by use of parameters sets proposed for MPGA (planar truss with 200-bars).

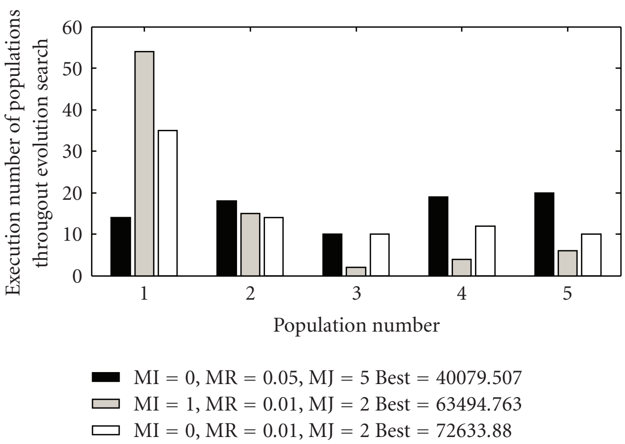


Figure 19: Activated numbers of each population obtained by MPGA (planar truss with 200-bars).

individuals. However, considering MPGA ability of using multiple populations with different parameters, it is possible to improve its performance by the implementation of genetic operations proposed by BGAwEIS.

(iii) It is demonstrated that BGAwEIS is able to obtain more convergent results compared to existing methods outlined in literature and optimal results obtained by MPGA and SGA.

(iv) The search in BGAwEIS is initiated with either a randomized or a user-defined population. Although a randomized population is used in this work, it is noted that the utilization of the user-defined population provides an advantage in the search for the offspring on the promising subsolution regions.

(v) The comparison of BGAwEIS, MPGA, and SGA is carried out by keeping several evolutionary parameters within certain limits. If population size and generation number is increased thereby assigning different values for the evolutionary parameters of these proposed algorithms, it is possible to improve quality of optimal designations.

In the future, the efficiency of BGAwEIS will be investigated thereby carrying out the several applications as follows.

(i) Statistical tests, such as parametric or nonparametric tests of hypotheses and variance analysis, will be performed for evaluation of the results generated by BGAwEIS. Thus, the best combination of parameter values will be determined considering the optimal designations with more convergent thereby including the decisions about the population distributions.

(ii) The possibilities used in extraction or insertion-based transmission and recreation of core population will be arranged for a self-adaptive usage.

(iii) MPGA will be modified to implement the main components of BGAwEIS. Moreover, the parallel and hybrid models of this improved algorithm will be also proposed to observe how the quality degree of optimal designations varies.

Appendix

The position numbers corresponding to optimal design for example 3 are [10, 14, 13, 14, 6, 2, 2, 7, 7, 13, 5, 23, 9, 23, 1, 20, 1, 11, 13, 9, 2, 2, 17, 19, 18, 9, 15, 13, 12, 5, 4, 10, 15, 14, 7, 22, 17, 16, 21, 5, 3, 11, 7, 23, 6, 4, 8, 27, 15, 15, 13, 17, 21, 9, 26, 8, 8, 8, 14, 8, 6, 4, 19, 8, 15, 14, 4, 17, 17, 15, 21, 2, 17, 13, 8, 7, 17, 9, 7, 19, 9, 10, 4, 9, 6, 8, 16, 1, 13, 5, 22, 12, 7, 7, 5, 11, 3, 2, 1, 16, 17, 24, 10, 5, 20, 17, 2, 18, 7, 7, 14, 9, 15, 8, 1, 4, 8, 5, 5, 2, 8, 27, 1, 8, 17, 8, 19, 23, 23, 4, 7, 20, 9, 8, 4, 9, 7, 7, 12, 16, 15, 6, 16, 14, 1, 14, 6, 3, 16, 12, 20, 18, 15, 7, 3, 2, 6, 11, 3, 15, 10, 22, 8, 17, 14, 19, 17, 3, 18, 11, 15, 5, 17, 8, 20, 8, 18, 8, 4, 8, 20, 21, 6, 12, 3, 19, 16, 7, 17, 15, 11, 13, 13, 11, 11, 23, 22, 10, 18, 22;]

Nomenclature

ρ: Density of steel
L: Length of member
A: Cross-sectional area
σ: Member stress
$\sigma_{\max}$: Maximum allowable stress
U: Joint displacement
$U_{\max}$: Maximum allowable displacement
F: Fitness value
P: Penalty value
W: Weight of truss system
X: Design vector
r_0, φ, f: Penalty constants
t: Current generation number
BVI: Values of interval bounds
BVI_U: Upper bound of interval
BVI_L: Lower bound of interval
$F_{\mathrm{inw}}, F_{\mathrm{outw}}, F_{\mathrm{cor}}$: Fitness values of inward, outward and core populations
$\mathrm{Par}_{\mathrm{rank}}, \mathrm{Par}_{\mathrm{mig}}, \mathrm{Par}_{\mathrm{mut}}, \mathrm{Par}_{\mathrm{cr}}, P_{\mathrm{sel}}$: Parameters regarded with ranking mutation, crossover and selection operations

P_{inw}: Inward population
P_{outw}: Outward population
P_{cor}: Core population
$X_{F\,\min}$: Lower bound of feasible solution pool
$X_{F\,\max}$: Upper bound of feasible solution pool
$X_{\max}$: Upper bound of design variable
$X_{\min}$: Lower bound of design variable
X_{BF}: Best feasible design variable
FeasPool: Feasible solution pool used to collect feasible solutions
DVN: Design variable number
NDV: Number of design variables
NFS: Number of feasible solution collected in feasible solution pool
NGGES: Number of generations for gradual exploration strategy
NG: Number of generations
NSAS: Number of subsolution regions after search
NSBS: Number of subsolution regions (number of segment) before search
SP: Size of population
SSR: Size of solution region
SubPopNum: Number of subpopulations
SubPopIndNum: Number of individuals contained each subpopulation
VNDV: Value of each design variable
SN: Subsolution region (segment) number.

References

[1] M. P. Saka, "Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review," in *Civil Engineering Computations: Tools and Techniques*, B. H. V. Topping, Ed., pp. 105–147, Saxe-Coburg, Stirlingshire, UK, 2007.

[2] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms," Tech. Rep. 826, Caltech Concurrent Computation Program, California Institute of Technology, Pasadena, Calif, USA, 1989.

[3] N. Radcliffe and P. Surry, "Formal memetic algorithms," in *Proceedings of the AISB Workshop on Evolutionary Computing*, T. C. Fogarty, Ed., vol. 865 of *Lecture Notes in Computer Science*, pp. 1–16, Leeds, UK, April 1994.

[4] E. Cantu-Paz, "A survey of parallel genetic algorithms. Calculateurs paralleles," *Reseaux et Systems Repartis*, vol. 10, no. 2, pp. 141–171, 1998.

[5] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.

[6] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.

[7] H. Pohlheim, "GEATbx: The Genetic and Evolutionary Algorithm Toolbox for Matlab," (1995–2002), http://www.geatbx.com/.

[8] C. A. Cornell, "Examples of optimization in structural design," Tech. Rep. R65-26, University of Waterloo, Waterloo, Canada, 1966.

[9] R. H. Bigelow and E. H. Gaylord, "Design of steel frames for minimum weight," *Journal of The Structural Division*, vol. 93, no. 6, pp. 109–1, 1967.

[10] K. I. Majid and D. W. Elliott, "Optimum design of frames with deflection constraints by nonlinear programming," *The Structural Engineer*, vol. 49, no. 4, pp. 179–188, 1971.

[11] G. N. Vanderplaats and H. Sugimoto, "A general-purpose optimization program for engineering design," *Computers & Structures*, vol. 24, no. 1, pp. 13–21, 1986.

[12] B. L. Karihaloo and S. Kanagasundaram, "Minimum-weight design of structural frames," *Computers & Structures*, vol. 31, no. 5, pp. 647–655, 1989.

[13] G. Gülay and H. Boduroğlu, "An algorithm for the optimum design of braced and unbraced steel frames under earthquake loading," *Earthquake Engineering & Structural Dynamics*, vol. 18, no. 1, pp. 121–128, 1989.

[14] T. Lassen, "Optimum design of three-dimensional framework structures," *Journal of The Structural Engineering*, vol. 119, no. 3, pp. 713–727, 1993.

[15] L. Wang and R. Grandhi, "Optimal design of frame structures using multi-point spline approximation," *AIAA Journal*, vol. 32, no. 10, pp. 2090–2098, 1994.

[16] E. Salajegheh, "Optimum design of structures with reference to space structures," *International Journal of Space Structures*, vol. 10, no. 4, pp. 205–214, 1995.

[17] S. Hernández, "Optimum design of steel structures," *Journal of Constructional Steel Research*, vol. 46, no. 1–3, pp. 374–378, 1998.

[18] J. S. Arora, *Introduction to Optimum Design*, McGraw-Hill, New York, NY, USA, 1989.

[19] G. E. Cameron, L. Xu, and D. E. Grierson, "Discrete optimal design of 3D frameworks," in *Proceedings of the 10th ASCE Conference on Electronic Computation*, pp. 181–188, Indianapolis, Ind, USA, April-May 1991.

[20] M. P. Saka and M. S. Hayalioglu, "Optimum design of geometrically nonlinear elastic-plastic steel frames," *Computers & Structures*, vol. 38, no. 3, pp. 329–344, 1991.

[21] M. S. Hayalioglu and M. P. Saka, "Optimum design of geometrically nonlinear elastic-plastic steel frames with tapered members," *Computers & Structures*, vol. 44, no. 4, pp. 915–924, 1992.

[22] C.-M. Chan, D. E. Grierson, and A. N. Sherbourne, "Automatic optimal design of tall steel building frameworks," *Journal of Structural Engineering*, vol. 121, no. 5, pp. 838–847, 1995.

[23] C.-M. Chan, A. N. Sherbourne, and D. E. Grierson, "Stiffness optimization technique for 3D tall steel building frameworks under multiple lateral loadings," *Engineering Structures*, vol. 16, no. 8, pp. 570–576, 1994.

[24] M. P. Saka, "Optimum design of steel frames with tapered members," *Computers & Structures*, vol. 63, no. 4, pp. 797–811, 1997.

[25] M. P. Saka and E. S. Kameshki, "Optimum design of unbraced rigid frames," *Computers & Structures*, vol. 69, no. 4, pp. 433–442, 1998.

[26] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence through Simulated Evolution*, John Wiley & Sons, New York, NY, USA, 1966.

[27] I. Rechenberg, *Evolutionstrategie: Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution*, Frommann-Holzboog, Stutugart, Germany, 1973.

[28] M. Keijzer and V. Bobovic, “Dimensionally aware genetic programming,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO ’99)*, pp. 1069–1076, Morgan Kaufmann, Orlando, Fla, USA, July 1999.

[29] M. Keijzer and V. Bobovic, “Genetic programming within a framework of computer-aided discovery of scientific knowledge,” in *Proceedings of the 2nd Genetic and Evolutionary Computation Conference (GECCO ’00)*, pp. 543–550, Morgan Kaufmann, Las Vegas, Nev, USA, July 2000.

[30] A. Cevik, “Genetic programming based formulation of rotation capacity of wide flange beams,” *Journal of Constructional Steel Research*, vol. 63, no. 7, pp. 884–893, 2007.

[31] T. Back and H. P. Schwefel, “Evolution strategies—I: variants and their computational implementation,” in *Genetic Algorithms in Engineering and Computer Science*, G. Winter, J. Perieaux, M. Gala, and P. Cuesta, Eds., pp. 111–126, John Wiley & Sons, Chichester, UK, 1995.

[32] T. Back and H. P. Schwefel, “Evolution Strategies—II: therotical aspects,” in *Genetic Algorithms in Engineering and Computer Science*, G. Winter, J. Perieaux, M. Gala, and P. Cuesta, Eds., pp. 127–140, John Wiley & Sons, Chichester, UK, 1995.

[33] J. Cai and G. Thierauf, “Discrete structural optimization using evolution strategies,” in *Neural Networks and Combinatorial Optimization in Civil and Structural Engineering*, B. H. V. Topping and A. I. Khan, Eds., pp. 95–100, Civil-Comp, Edinburgh, UK, 1993.

[34] G. Ebenau, J. Rottschäfer, and G. Thierauf, “An advanced evolutionary strategy with an adaptive penalty function for mixed-discrete structural optimisation,” *Advances in Engineering Software*, vol. 36, no. 1, pp. 29–38, 2005.

[35] S. Rajasekaran, “Optimal laminate sequence of non-prismatic thin-walled composite spatial members of generic section,” *Composite Structures*, vol. 70, no. 2, pp. 200–211, 2005.

[36] B. Baumann and B. Kost, “Structure assembling by stochastic optimization,” *Computers & Structures*, vol. 83, no. 25-26, pp. 2175–2184, 2005.

[37] P. Hajela, “Stochastic search in structural optimization: genetic algorithm and simulated annealing,” Tech. Rep. NTRS 2004-11-03, Glenn Research Center, Cleveland, Ohio, USA, 1993.

[38] H. Adeli and N.-T. Cheng, “Augmented Lagrangian genetic algorithm for structural optimization,” *Journal of Aerospace Engineering*, vol. 7, no. 1, pp. 104–118, 1994.

[39] D. Chen, *Least weight design of 2D and 3D geometrically nonlinear structures using a genetic algorithm*, PhD. dissertation, University of Memphis, Memphis, Tenn, USA, 1997.

[40] J. Yang and C. K. Soh, “Structural optimization by genetic algorithms with tournament selection,” *Journal of Computing in Civil Engineering*, vol. 11, no. 3, pp. 195–200, 1997.

[41] S. Y. Wang and K. Tai, “Graph representation for structural topology optimization using genetic algorithms,” *Computers & Structures*, vol. 82, no. 20-21, pp. 1609–1622, 2004.

[42] A. Kaveh and V. Kalatjari, “Topology optimization of trusses using genetic algorithm, force method and graph theory,” *International Journal for Numerical Methods in Engineering*, vol. 58, no. 5, pp. 771–791, 2003.

[43] M. P. Saka, “Optimum design of pin-jointed steel structures with practical applications,” *Journal of The Structural Engineering*, vol. 116, no. 10, pp. 2599–2620, 1990.

[44] M. P. Saka, A. Daloglu, and F. Malhas, “Optimum spacing design of grillage systems using genetic algorithm,” in *Innovation in Computer Methods for Civil and Structural Engineering*, B. H. V. Topping and M. B. Leeming, Eds., pp. 137–148, Civil-Comp Press, Edinburgh, UK, 1997.

[45] V. Toğan and A. T. Daloğlu, “An improved genetic algorithm with initial population strategy and self-adaptive member grouping,” *Computers & Structures*, vol. 86, no. 11-12, pp. 1204–1218, 2008.

[46] C. Camp, S. Pezeshk, and G. Cao, “Optimized design of two-dimensional structures using a genetic algorithm,” *Journal of Structural Engineering*, vol. 124, no. 5, pp. 551–559, 1998.

[47] K. M. Rasheed, *GADO: a genetic algorithm for continues design optimization*, PhD. dissertation, The State University of New Jersey, News Brunswick, NJ, USA, 1998.

[48] R. G. Le Riche, C. Knopf-Lenoir, and R. T. Haftka, “A segregated genetic algorithm for constrained structural optimization,” in *Proceedings of the 6th International Conference on Genetic Algorithms*, L. J. Eshelman, Ed., pp. 558–565, Morgan Kaufmann, Pittsburgh, Pa, USA, July 1995.

[49] C. A. Coello Coello, “Use of a self-adaptive penalty approach for engineering optimization problems,” *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.

[50] P. Nanakorn and K. Meesomklin, “An adaptive penalty function in genetic algorithms for structural design optimization,” *Computers & Structures*, vol. 79, no. 29-30, pp. 2527–2539, 2001.

[51] S. M. Garrett, “How do we evaluate artificial immune systems?” *Evolutionary Computation*, vol. 13, no. 2, pp. 145–178, 2005.

[52] J. Yoo and P. Hajela, “Immune network modeling in design optimization,” in *New Ideas in Optimization*, pp. 167–183, McGraw-Hill, London, UK, 1999.

[53] C. A. Coello Coello and N. Cruz Cortés, “Hybridizing a genetic algorithm with an artificial immune system for global optimization,” *Engineering Optimization*, vol. 36, no. 5, pp. 607–634, 2004.

[54] R. Dawkins, *The Selfish Gene*, Oxford University Press, Oxford, UK, 1976.

[55] Y. S. Ong and A. J. Keane, “Meta-Lamarckian learning in memetic algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 99–110, 2004.

[56] J. E. Smith, “Coevolving memetic algorithms: a review and progress report,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 37, no. 1, pp. 6–17, 2007.

[57] E. Cantú-Paz and D. E. Goldberg, “Modeling idealized bounding cases of paralel genetic algorithms,” in *Proceedings of the 2nd Annual Conference on Genetic Programming*, J. Koza, K. Deb, M. Dorigo, et al., Eds., pp. 353–361, Morgan Kaufmann, San Francisco, Calif, USA, July 1997.

[58] E. Cantú-Paz, “A summary of research on parallel genetic algorithms,” IlliGAL Report 95007, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana-Champaign, Ill, USA, 1995.

[59] E. Cantú-Paz, “Migration policies, selection pressure, and parallel evolutionary algorithms,” *Journal of Heuristics*, vol. 7, no. 4, pp. 311–334, 2001.

[60] J. J. Grefenstette, “Parallel adaptive algorithms for function optimization,” Tech. Rep. CS-81-19, Vanderbilt University, Nashville, Tenn, USA, 1981.

[61] G. Robbins, “EnGENEer—the evolution of solutions,” in *Proceedings of the 5th Annual Seminar on Neural Networks and Genetic Algorithms*, pp. 218–232, London, UK, March 1992.

[62] D. Levine, "Users guide to the PGAPack parallel genetic algorithm library," Tech. Rep. ANL-95/18, Argonne National Laboratory, Argonne, Ill, USA, 1996.

[63] S. Baluja, "Structure and performance of fine-grain parallelism in genetic search," in *Proceedings of the 5th International Conference on Genetic Algorithms (ICGA '93)*, pp. 155–162, Urbana-Champaign, Ill, USA, June 1993.

[64] F. Herrera, M. Lozano, and C. Moraga, "Hierarchical distributed genetic algorithms," *International Journal of Intelligent Systems*, vol. 14, no. 11, pp. 1099–1121, 1999.

[65] J. P. Cohoon, S. U. Hedge, W. N. Martin, and D. S. Richards, "Punctuated equilibria: a parallel genetic algorithm," in *Proceedings of the 2nd International Conference on Genetic Algorithms and Their Application (ICGA '87)*, J. J. Grefenstette, Ed., pp. 148–154, Cambridge, Mass, USA, July 1987.

[66] E. Alba and J. M. Troya, "An analysis of synchronous and asynchronous parallel distributed genetic algorithms with structured and panmictic islands," in *Proceedings of the 13th International Parallel Processing Symposium and the 10th Symposium on Parallel and Distributed Processing (IPPS/SPDP '99)*, vol. 1586 of *Lecture Notes in Computer Science*, pp. 248–256, San Juan, Puerto Rico, USA, April 1999.

[67] Z. Skolicki and K. A. De Jong, "Improving evolutionary algorithms with multi-representation island models," in *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature(PPSN '04)*, vol. 3242 of *Lecture Notes in Computer Science*, pp. 420–429, Springer, Birmingham, UK, September 2004.

[68] R. Tanese, "Paralel genetic algorithms for a hypercube," in *Proceedings of the 2nd International Conference on Genetic Algorithms (ICGA '87)*, J. J. Grefenstette, Ed., pp. 177–183, Lawrence Erlbaum Associates, Cambridge, Mass, USA, July 1987.

[69] R. Tanese, "Distributed genetic algorithms," in *Proceedings of the 3rd International Conference on Genetic Algorithms (ICGA '89)*, J. D. Schaffer, Ed., pp. 434–439, Morgan Kaufmann, Fairfax, Va, USA, June 1989.

[70] E. Alba and J. M. Troya, "Improving flexibility and efficiency by adding parallelism to genetic algorithms," *Statistics and Computing*, vol. 12, no. 2, pp. 91–114, 2002.

[71] W. N. Martin, J. Lienig, and J. P. Cohoon, "Island (migration) models: evolutionary algorithms based on punctuated equilibria," in *Handbook of Evolutionary Computation*, T. Bäck, D. B. Fogel, and Z. Michalewicz, Eds., pp. C6.3:1–C6.3:16, Institute of Physics, Bristol, UK, 1997.

[72] G. Folino, C. Pizzuti, and G. Spezzano, "Combining cellular genetic algorithms and local search for solving satisfiability problems," in *Proceedings of the 10th IEEE International Conference on Tools with Artificial Intelligence*, pp. 192–198, Taipei, Taiwan, November 1998.

[73] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp. 474–488, 2005.

[74] F. Neri, J. Toivanen, G. L. Cascella, and Y.-S. Ong, "An adaptive multimeme algorithm for designing HIV multidrug therapies," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 2, pp. 264–278, 2007.

[75] A. Caponio, G. L. Cascella, F. Neri, N. Salvatore, and M. Sumner, "A fast adaptive memetic algorithm for online and offline control design of PMSM drives," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 37, no. 1, pp. 28–41, 2007.

[76] Q. H. Quang, Y. S. Ong, M. H. Lim, and N. Krasnogor, "Adaptive cellular memetic algorithm," *Evolutionary Computation*, vol. 17, no. 2, pp. 257–274, 2009.

[77] J. Sakamoto and J. Oda, "A technique of optimal layout design for truss structures using genetic algorithm," in *Proceedings of the 34th AIAA/ASCE/ASME/AHS Structural Dynamics and Material Conference and AIAA/ASME Adaptive Structures Forum*, pp. 2402–2408, La Jolla, Calif, USA, April 1993.

[78] C. K. Soh and J. Yang, "Fuzzy controlled genetic algorithm search for shape optimization," *Journal of Computing in Civil Engineering*, vol. 10, no. 2, pp. 143–150, 1996.

[79] H. Adeli and N.-T. Cheng, "Concurrent genetic algorithms for optimization of large structures," *Journal of Aerospace Engineering*, vol. 7, no. 3, pp. 276–296, 1994.

[80] B. H. V. Topping and J. P. B. Leite, "Parallel genetic models for structural optimization," *Engineering Optimization*, vol. 31, no. 1, pp. 65–99, 1998.

[81] H. Adeli and S. Kumar, "Distributed genetic algorithm for structural optimization," *Journal of Aerospace Engineering*, vol. 8, no. 3, pp. 156–163, 1995.

[82] K. C. Sarma and H. Adeli, "Bilevel parallel genetic algorithms for optimization of large steel structures," *Computer-Aided Civil and Infrastructure Engineering*, vol. 16, no. 5, pp. 295–304, 2001.

[83] Y. Tanimura, T. Hiroyasu, and M. Miki, "Dicussion on didtributed genetic algorithms for designing truss structures," in *Proceedings of the 5th International Conference and Exhibition on High Performance Computing in the Asia-Pacific Region (HPC '01)*, pp. 1–13, Queensland, Australia, September 2001.

[84] R. Kicinger, T. Arciszewski, and K. A. De Jong, "Distributed evolutionary design: islandmodel based optimization of steel skeleton structures in tall buildings," in *Proceedings of the 10th International Conference on Computing in Civil and Building Engineering (ICCCBE '04)*, K. Beucke, B. Firmenich, D. Donath, R. Fruchter, and K. Roddis, Eds., p. 190, Weimar, Germany, June 2004.

[85] R. Kicinger and T. Arciszewski, "Emprical analysis of memetic algorithms for conceptual design of steel structural systems in tall buildings," in *Proceedings of International Conference on Advance in Engineering Structures, Mechanics & Construction*, vol. 140 of *Solid Mechanics and Its Applications*, pp. 277–288, Springer, Waterloo, Canada, May 2006.

[86] A. Kaveh and M. Shahrouzi, "Graph theoretical implementation of memetic algorithms in structural optimization of frame bracing layouts," *Engineering Computations*, vol. 25, no. 1, pp. 55–85, 2008.

[87] M. K. Karakasis, A. P. Giotis, and K. C. Giannakoglou, "Inexact information aided, low-cost, distributed genetic algorithms for aerodynamic shape optimization," *International Journal for Numerical Methods in Fluids*, vol. 43, no. 10-11, pp. 1149–1166, 2003.

[88] M. K. Karakasis, D. G. Koubogiannis, and K. C. Giannakoglou, "Hierarchical distributed metamodel-assisted evolutionary algorithms in shape optimization," *International Journal for Numerical Methods in Fluids*, vol. 53, no. 3, pp. 455–469, 2007.

[89] P. I. K. Liakopoulos, I. C. Kampolis, and K. C. Giannakoglou, "Grid enabled, hierarchical distributed metamodel-assisted evolutionary algorithms for aerodynamic shape optimization," *Future Generation Computer Systems*, vol. 24, no. 7, pp. 701–708, 2008.

[90] O. Hasançebi and F. Erbatur, "Constraint handling in genetic algorithm integrated structural optimization," *Acta Mechanica*, vol. 139–145, no. 1–4, pp. 15–31, 2000.

[91] A. E. Eiben and Z. Ruttkay, "Self-adaptivity for constraint satisfaction: learning penalty functions," in *Proceedings of the IEEE Conference on Evolutionary Computation (ICEC '96)*, pp. 258–261, Nagoya, Japan, May 1996.

[92] T. Bäck and F. Hoffmeister, "Extended selection mechanisms in genetic algorithms," in *Proceedings of the 4th International Conference on Genetic Algorithms (ICGA '91)*, pp. 92–99, San Diego, Calif, USA, July 1991.

[93] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proceedings of the 2nd International Conference on Genetic Algorithms (ICGA '87)*, pp. 41–49, Cambridge, Mass, USA, July 1987.

[94] D. Schlierkamp-Voosen and H. Mühlenbein, "Adaptation of population sizes by competing subpopulations," in *Proceedings of the IEEE Conference on Evolutionary Computation (ICEC '96)*, pp. 330–335, Nagoya, Japan, May 1996.

[95] S. Rajeev and C. S. Krishnamoorthy, "Genetic algorithms-based methodologies for design optimization of trusses," *Journal of Structural Engineering*, vol. 123, no. 3, pp. 350–358, 1997.

[96] D. M. Zhu, "An improved Templaman's algorithm for optimum design of trusses with discrete member sizes," *Engineering Optimization*, vol. 9, pp. 303–312, 1986.

[97] F. Erbatur, O. Hasançebi, I. Tütüncü, and H. Kılıç, "Optimal design of planar and space structures with genetic algorithms," *Computers & Structures*, vol. 75, no. 2, pp. 209–224, 2000.

[98] C. A. Coello Coello, M. Rudnick, and A. D. Christiansen, "Using genetic algorithms for optimal design of trusses," in *Proceedings of the 6th International Conference on Tools with Artificial Intelligence (TAI '94)*, pp. 88–94, New Orleans, La, USA, November 1994.

[99] S.-J. Wu and P.-T. Chow, "Genetic algorithms for solving mixed-discrete optimization problems," *Journal of the Franklin Institute*, vol. 331, no. 4, pp. 381–401, 1994.

[100] V. B. Venkayya, "Design of optimum structures," *Computers & Structures*, vol. 1, no. 1-2, pp. 265–309, 1971.

[101] R. A. Gellatly and L. Berke, "Optimal structural design," Tech. Rep. TR-70-16, Air Force Flight Dynamics Laboratory (AFFDL), Wright-Patterson Air force Base, Ohio, USA, 1971.

[102] X. Renwei and L. Peng, "Structural optimization based on second-order approximations of functions and dual theory," *Computer Methods in Applied Mechanics and Engineering*, vol. 65, no. 2, pp. 101–114, 1987.

[103] L. A. Schmit Jr. and B. Farshi, "Some approximation concepts for structural synthesis," *AIAA Journal*, vol. 12, no. 5, pp. 692–699, 1974.

[104] P. Ponterosso and D. S. J. Fox, "Heuristically seeded genetic algorithms applied to truss optimisation," *Engineering with Computers*, vol. 15, no. 4, pp. 345–355, 1999.

Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
International Journal of Distributed Sensor Networks
Advances in Civil Engineering
Journal of Control Science and Engineering
Hindawi
Submit your manuscripts at
http://www.hindawi.com
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration