Harmony Search Method for Multimodal Size, Shape, and Topology Optimization of Structural Frameworks

Kirk Martini, P.E.[1]

Abstract: This paper describes and demonstrates modifications of the harmony search method to support multimodal structural optimization. Several researchers have recognized the potential of population-based optimization methods, such as genetic algorithms and particle swarm optimization, to support multimodal optimization, that is, generating a range of good alternative solutions, rather than a single best solution. Among these population-based methods is the harmony search method, which has been demonstrated to be efficient and effective in many unimodal structural optimization problems. Toward the goal of making the harmony search method more effective in multimodal optimization, this paper describes a new strategy for generating solutions called close-harmony improvisation, and a new strategy for replacing solutions called local replacement. Examples demonstrate the effect of the two strategies used individually and in tandem. The discussion compares results with conventional harmony search and finds that close-harmony improvisation consistently improves the fitness of the search results, although the effect is sometimes mild, whereas local replacement is quite effective in increasing the diversity of the search result. **DOI: 10.1061/(ASCE)ST.1943-541X.0000378.**

CE Database subject headings: Optimization; Algorithms; Structural design.

Author keywords: Optimization algorithms; Structural design.

Introduction

Population-based metaheuristic optimization methods have been widely applied in structural optimization. These methods include genetic algorithms, particle swarm optimization, ant colony optimization, harmony search method, and several hybrids of these and other methods. Hasancebi et al. (2009) gives a thorough overview and comparison of such methods. Some researchers have noted that population-based methods have the potential to generate a range of viable alternatives, which can be more useful than developing a single best solution (Balling 2006; von Buelow 2007; Winslow 2008). One advantage is that fitness functions typically cannot account for considerations that are difficult to quantify, such as aesthetics and constructability. An optimization algorithm that produces alternatives allows a human designer to assess those alternatives with respect to external considerations. Optimization methods that identify multiple locally optimal solutions are called multimodal.

Balling et al. (2006) and von Buelow (2007) have developed multimodal genetic algorithms for structural optimization, which have proven effective but computationally intensive. Examples in the Balling et al. study used a population of 1,000 designs computed over 500 generations, meaning 500,000 model evaluations. Examples in the von Buelow (2007) study used a multilevel arrangement with a population of 20 topologies, each with 50 corresponding geometries, resulting in a total population of 1,000 that is run for hundreds of cycles, implying hundreds of thousands of model evaluations, similar to Balling et al. (2006). There is a question as to whether useful multimodal optimization can be achieved more efficiently.

Harmony search is a promising possibility because it produces one design with each cycle rather than an entire new population and therefore offers the potential to produce well-developed solutions with fewer model evaluations. This paper presents modifications to the harmony search method that address this question. These modifications concern two steps of the search process: the strategy for generating new solutions and the strategy for replacing existing solutions with new solutions. These modifications have been applied to multimodal optimization of a two-dimensional (2D) truss and a three-dimensional (3D) arch structure by using decision variables that include member cross sections (size), geometric properties of the framework (shape), and the presence or absence of members (topology). The discussion begins with a formal statement of the representation of the optimization problem followed by a review of the conventional harmony search method. This paper then describes the modifications incorporated in the proposed method followed by examples and conclusions.

Representation of the Optimization Problem

A particular design configuration is called a solution, and is represented as a vector $\mathbf{x}$ of decision variables x_i. The objective is to find a solution that minimizes a fitness function $f(\mathbf{x})$ subject to multiple constraints

$$g_j(\mathbf{x}) \leq 0 \quad (1)$$

A variable x_i may vary continuously (as for geometric dimensions), may represent discrete symbolic values from a predefined list (as for standard structural cross sections), or may be boolean (as for member removal in defining structural topology). For a continuous variable x_i, the value must lie between a specified upper bound (x_i^U) and lower bound (x_i^L).

Concerning constraints, there are three types: stiffness, strength, and stability. Stiffness constraints are formulated as shown in

[1]Associate Professor, Dept. of Architecture, Univ. of Virginia, P.O. Box 400122, Charlottesville, VA 22904. E-mail: martini@virginia.edu

Note. This manuscript was submitted on July 2, 2010; approved on January 3, 2011; published online on January 5, 2011. Discussion period open until April 1, 2012; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Structural Engineering***, Vol. 137, No. 11, November 1, 2011. ©ASCE, ISSN 0733-9445/2011/11-1332–1339/$25.00.

Eq. (2), in which Δ_j = displacement in a specified coordinate direction of a node j, and Δ_{allow} = allowable displacement in that coordinate direction

$$g_j(\mathbf{x}) = \frac{\Delta_j}{\Delta_{\text{allow}}} - 1 \tag{2}$$

There are multiple options for defining strength constraints. One of the examples discussed subsequently uses a simple allowable stress criterion for truss elements, and in that case, the strength constraints are defined as shown in Eq. (3), in which σ_k is the stress in element k, and σ_{allow} is the allowable stress

$$g_k(\mathbf{x}) = \frac{\sigma_k}{\sigma_{\text{allow}}} - 1 \tag{3}$$

Other examples use strength criteria defined in the AISC LRFD steel design code (AISC 2001). These criteria are expressed in interaction formulas for axial force and bending moment for frame elements as shown in Eqs. (4) and (5)

$$\frac{P_u}{\phi P_n} \geq 0.2: \quad g_{bk} = \frac{P_u}{\phi P_n} + \frac{8}{9}\left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right) - 1 \tag{4}$$

$$\frac{P_u}{\phi P_n} < 0.2: \quad g_{bk} = \frac{P_u}{2\phi P_n} + \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right) - 1 \tag{5}$$

in which P_u, M_{ux}, and M_{uy} = maximum axial force, strong axis bending moment, and weak axis bending moment as a result of factored loads; P_n, M_{nx}, and M_{ny} = nominal capacity of a member in axial force, strong axis bending moment, and weak axis bending moment; ϕ and ϕ_b = capacity reduction factors for axial force and bending; and g_{bk} = value of the bending and axial constraint for element k.

In addition, for AISC hollow structural sections (HSS), the strength constraints consider shear and torsion as shown in Eq. (6)

$$g_{tk} = \left(\frac{P_u}{\phi P_n} + \frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right) + \left(\frac{V_u}{\phi_v V_n} + \frac{T_u}{\phi_T T_n}\right)^2 - 1 \tag{6}$$

in which V_u and T_u = maximum shear and torsion in a member as a result of factored loads, V_n and T_n = nominal capacity of the member in shear and torsion, and ϕ_v and ϕ_T = capacity reduction factors for shear and torsion. The value of the constraint for element k, g_k, is taken as the larger of g_{bk} and g_{tk}.

The degree of constraint violation for a solution is set as the maximum value among all positive constraints rather than as a sum of the positive constraints because in problems with topological variables, some solutions have more members than others. Comparing the sum of the constraint values could be misleading in that case. In addition to stiffness constraints for joints and strength constraints for elements, the method can also account for stability constraints for the entire structure with an option to perform a large displacement analysis. If the analysis fails to converge, the solution is considered unstable and assigned a large constraint value (1.0×10^6 for the examples in this paper).

Review of the Harmony Search Method

The harmony search method was introduced by Geem et al. (2001) and is based on a musical metaphor where variable values are viewed as musical pitches: a solution vector is analogous to a musical chord. That method will be called conventional harmony search. The following is a brief summary of its essential steps:

Step 1: Initialization. Produce a collection of solutions with random values. The collection is called harmony memory, and the number of solutions is the harmony memory size (μ).
Step 2: Improvisation. Improvise (i.e., generate) a new solution derived from the solutions in harmony memory by using a stochastic procedure described subsequently.
Step 3: Replacement. If the fitness of the new solution is better than that of the worst solution in harmony memory, replace that worst solution with the new solution. This replacement strategy will be called global replacement (GR).
Step 4: Termination check. If the termination criterion is not satisfied, go to step 2. The termination criterion is typically a specified maximum number of cycles.

Step 2 generates a new solution by using three control parameters to guide the stochastic process: the harmony memory consideration rate (η), the pitch adjustment rate (ρ), and the bandwidth (β). The η and ρ parameters represent probabilities. Improvising a new solution involves the following steps: First, initialize a solution vector by assigning each variable a random value. For each variable, with a probability defined by η, set the value to that of a randomly chosen solution in harmony memory. If the value was chosen from harmony memory, then with a probability defined by ρ, modify the value by a small change, called a pitch adjustment. For continuous variables, the pitch adjustment is calculated as a uniform random number on the interval $[-1, 1]$ times the bandwidth parameter β, which is typically a small percentage of the possible range. For discrete variables, the pitch adjustment typically means a few discrete steps. The examples in this study used a bandwidth of 1% of the variable range for continuous variables, and one discrete step for discrete variables.

The success of the harmony search is typically sensitive to the values of the control parameters (Degertekin 2008). Researchers have proposed a range of adaptive strategies to set the η and ρ factors dynamically (Hasancebi et al. 2010; Mahdavi et al. 2007; Wang and Huang 2010). The examples in this study use the adaptive method proposed by Hasancebi et al. (2010). In this method, at the beginning of design cycle i, the algorithm computes values of η_i and ρ_i on the basis of past values by using Eqs. (7) and (8)

$$\eta_i = \left(1 + \frac{1-\bar{\eta}}{\bar{\eta}} e^{-\lambda N(0,1)}\right)^{-1} \tag{7}$$

$$\rho_i = \left(1 + \frac{1-\bar{\rho}}{\bar{\rho}} e^{-\lambda N(0,1)}\right)^{-1} \tag{8}$$

in which $\bar{\eta}$ and $\bar{\rho}$ = average values over all solutions in harmony memory, $N(0,1)$ = normally distributed random number with an expected value of 0 and a variance of 1, and λ = parameter set to 0.35 for the examples in this study. Hasancebi et al. (2010) provide more detail.

Concerning modifications of harmony search to support multimodal optimization, Gao et al. (2009) have proposed a notable method. When a new solution is improvised, that method determines the solutions in harmony memory within a specified distance, in which distance is some scalar measure of similarity; these solutions are said to be in the vicinity. If the fitness of the new solution is higher than the average fitness of solutions in its vicinity, and if the number of solutions in the vicinity is fewer than a user-specified limit, then the new solution may replace the worst solution in harmony memory. This method has characteristics in common with the replacement method proposed in this study, but there are significant differences explained subsequently.

Modifications for Multimodal Optimization

One characteristic of conventional harmony search is the tendency of the solutions to converge toward a common configuration as cycles progress. This characteristic supports unimodal optimization but not multimodal optimization. In terms of the music metaphor, multimodal optimization seeks to produce not a single harmonious chord, but rather a diverse collection of chords, as in an interesting piece of music. The proposed modifications to support multimodal optimization concern the improvisation and replacement steps of the algorithm, with the replacement step relying on an approach to constraint handling that avoids penalty functions. The modifications also rely on methods to measure similarity among solutions. The discussion begins with constraint handling and measures of similarity and then moves to the replacement and improvisation steps.

Constraint Handling

Published implementations of harmony search have commonly handled constraints by using penalty functions (Degertekin 2008; Hasancebi et al. 2010; Geem et al. 2001), however there are several well-known drawbacks to that approach (Deb 2000). Simple penalty methods typically require trial and error experimentation to set parameters effectively. More complex adaptive penalty methods are well suited to some types of problems, but not others. The method presented in this paper handles constraints by adapting the method proposed by Deb (2000) for genetic algorithms. This method uses the following three rules for comparing solutions:

1. Any feasible solution is preferred to any infeasible solution;
2. Among two infeasible solutions, the one having the smaller constraint violation is preferred; and
3. Among two feasible solutions, the one having the better objective function value is preferred.

When applied to harmony search, these rules create two modes of operation called culling and refinement. Culling concerns the application of rules 1 and 2 in guiding the replacement of infeasible solutions in harmony memory. Refinement concerns the application of rules 1 and 3 in guiding the replacement of feasible solutions in harmony memory. This method of constraint handling is similar to one proposed by Gao et al. (2009) for the harmony search method, although Gao does not reference the study by Deb (2000).

Measure of Similarity: Normalized Euclidean Distance

The methods proposed in this study rely on normalized Euclidean distance as a measure of similarity. The following discussion reviews this concept and explains its application in guiding the search. The definition of normalized Euclidean distance can begin with the definition of normalized value for a variable. For a solution $\mathbf{x}$, the normalized value v_i of the variable x_i is defined by Eq. (9):

$$v_i = \frac{x_i - x_i^L}{x_i^U - x_i^L} \tag{9}$$

The concept of normalized value leads to normalized difference. Considering two solutions, $\mathbf{x}^{(j)}$ and $\mathbf{x}^{(k)}$, the normalized difference $d_i^{(jk)}$ between the corresponding variables $x_i^{(j)}$ and $x_i^{(k)}$ is defined as the difference between their normalized values:

$$d_i^{(jk)} = v_i^{(j)} - v_i^{(k)} = \frac{x_i^{(j)} - x_i^{(k)}}{x_i^U - x_i^L} \tag{10}$$

For boolean variables, the normalized difference between two variables equals 0 when the values are equal, and 1 when the values are different. For section variables, which are assigned discrete symbolic values from a predefined list, the organization is similar but somewhat more complex. In addition to specifying the list of sections, the user also specifies the primary section property (e.g., area or strong axis moment of inertia) for the variable. The numeric values of that section property serve as the basis for computing normalized difference per Eq. (10). Typically, the cross section area would be the primary section property for an axial force member, whereas strong axis moment of inertia or plastic modulus would be for a section used primarily in bending (Hasancebi et al. 2010).

By using the normalized difference, the normalized Euclidean distance $D^{(jk)}$ between solutions $\mathbf{x}^{(j)}$ and $\mathbf{x}^{(k)}$ is defined as follows (Deb 2000):

$$D^{(jk)} = \sqrt{\frac{1}{N_d}\sum_{i=1}^{N_d}\left(d_i^{(jk)}\right)^2} \tag{11}$$

in which N_d = number of design variables in a solution. For the sake of brevity, the following discussion uses the term distance to mean normalized Euclidean distance unless otherwise noted.

By using these concepts, design space can be viewed as a unit hypercube with a dimension equal to the number of decision variables. One particularly useful concept is the maximum distance between two feasible solutions, which will be called the feasible diameter (D_f) of the design space. The new replacement method proposed in this paper, described subsequently, employs this concept. The algorithm maintains a lower-bound estimate of D_f at design cycle i, denoted $D_f^{(i)}$, by computing the distance between each newly generated feasible solution and all previously determined feasible solutions. The $D_f^{(i)}$ parameter is set to the maximum distance found to that point in the search and so increases monotonically as the search progresses.

Replacement Method: Local Replacement

To counter the tendency of conventional harmony search to converge, this paper proposes a new replacement method called local replacement. When a new solution is improvised, local replacement first considers the set of feasible solutions, the distance of which from that solution is less than a specified distance R_n, called the neighborhood radius; the solutions in harmony memory within that radius are called close neighbors of the new solution; the number of close neighbors N_n is called the close neighbor count (S_n). If $N_n < S_n$, the neighborhood is called uncrowded; if $N_n = S_n$, the neighborhood is called crowded; and if $N_n > S_n$, the neighborhood is called overcrowded. When the neighborhood of a new feasible solution is uncrowded, the new solution is compared with and may replace the worst solution in harmony memory. If the neighborhood is crowded or overcrowded, then the set of close neighbors is sorted by fitness and the new solution is compared with the close neighbor with fitness rank equal to S_n. If the fitness of the new solution is better than the fitness of the S_n-ranked neighbor, then the new solution replaces that neighbor. When the neighborhood is overcrowded, then the algorithm employs thinning, in which all solutions with fitness worse than that of the S_n-ranked close neighbor are reset, meaning that the constraint violation is set to a large value (1.0×10^5 for examples in this study) that marks the solution as infeasible. After a solution has been reset, it will be replaced in later design cycles by any feasible solution or by any infeasible solution with a lower constraint violation. Local replacement inhibits its convergence by limiting the number of feasible solutions within a neighborhood. When the algorithm finds a neighborhood that is crowded or overcrowded, it puts only better solutions there, not more solutions.

Local replacement depends on two parameters, S_n and R_n. The S_n parameter needs to be a fraction of the harmony memory size, μ. The examples in this study set S_n to one-fifth of the harmony memory size, by using $\mu = 75$ and $S_n = 15$. Similarly, R_n needs to be a fraction of the feasible diameter D_f; if R_n is greater than or equal to D_f, then all feasible solutions could fall in one neighborhood, which would defeat the purpose. The method presented in this paper sets R_n as a fraction of the current lower-bound estimate of the feasible diameter $D_f^{(i)}$, which means that R_n is adaptive; its value depends on the particular problem and the stage of the search. The examples in this study used $R_n = 0.25D_f^{(i)}$.

As discussed previously, local replacement shares characteristics with a replacement strategy proposed by Gao et al. (2009). That strategy is similar in that it considers the fitness of nearby neighbors, but it is different in that it replaces the worst solution in harmony memory, irrespective of whether that worst solution is nearby. Local replacement, in contrast, replaces the S_n-ranked close neighbor when the neighborhood is crowded or overcrowded. Replacing the worst solution in harmony memory is one of the clearest common characteristics of the many published versions of the harmony search method, so local replacement represents a significant departure from established practice.

Improvisation Method: Close Harmony

As described previously, when conventional harmony search improvises a variable value for a new solution it may use a value from a randomly selected solution. When a value is selected from an existing solution, all solutions in harmony memory have an equal probability of being chosen as the source of the variable value. This approach will be called full-harmony improvisation and has been proven to work well for unimodal optimization. To better support multimodal optimization, the proposed method uses a new improvisation method called close-harmony improvisation. In close-harmony improvisation, the algorithm selects a subset of closely related solutions. There are several potential approaches to implementing this concept. The one pursued in this study is to improvise a new solution by first randomly selecting a solution from harmony memory and then gathering all solutions with a distance from that solution that is less than or equal to a specified distance. The set of solutions is called the close-harmony set, and the specified distance is called the close-harmony radius (R_h). The new solution is improvised from the close-harmony set in the same way that full-harmony improvisation generates a solution from the entire harmony memory. In order to make R_h adaptive, in this study it is set equal to the average distance among all solutions in harmony memory.

Concerning control parameters in close-harmony improvisation, the algorithm modifies the adaptive strategy proposed by Hasancebi et al. (2010), shown in Eqs. (7) and (8). In that strategy, the terms $\bar{\eta}$ and $\bar{\rho}$ are averaged over all the solutions in harmony memory. In close-harmony improvisation, the modified version instead averages those values over the members of the close-harmony set. Hasancebi et al. (2010) notes that the adaptive strategy allows for different values of the control parameters during different phases of the search. The intent of the modification for close harmony is to also allow different values of the control parameters in different regions of design space.

Accounting for Topology

The proposed algorithm supports topologic optimization by using methods proposed by Balling et al. (2006) for optimization with a genetic algorithm. In particular, the specification of a member includes an option to mark it as removable. Member removal is modeled by giving the member an extremely low stiffness (the examples in this study use 1.0×10^{-6} times the specified stiffness), which models the effect of removing the member but maintains numeric stability in structural analysis. Structures that are kinematically unstable produce extremely large violations of stiffness constraints. In addition, when the maximum strength constraint value for a member is less than a specified threshold (the examples in this paper use 1.0×10^{-4}), that member is marked as spurious and the algorithm removes it. Balling et al. (2006) provides further detail.

When a member is marked as removable, the algorithm adds a boolean decision variable to the solution, called a topologic variable, in which a value of true means the member is removed. In harmony search, topologic variables introduce a new stochastic parameter, the member removal rate (α), which specifies the probability that a member will be removed when a model is generated randomly. For example, when $\alpha = 0.2$, there is a 20% chance that a randomly generated topologic variable will take the value of true. In addition, topologic variables are not considered by the pitch adjustment mechanism in improvisation, as that mechanism is intended to introduce small changes to a variable's value, and boolean variables do not allow small changes.

Examples

Overview

The following discussion applies the two new strategies described previously to two examples: a widely published planar truss problem and a three-dimensional arch bridge. Each example considers four search methods on the basis of the possible combinations of two improvisation strategies, full harmony (FH) and close harmony (CH), and two replacement strategies, global replacement (GR), and local replacement (LR). The four methods are denoted as follows: FH-GR, CH-GR, FH-LR, and CH-LR. Note that FH-GR corresponds to conventional harmony search. The primary method of interest is CH-LR, which employs both of the new strategies. The CH-GR and FH-LR methods are included to better demonstrate the effect of each new strategy acting alone. For each of the four methods, the study performed 10 optimization runs of 4,000 design cycles each.

Concerning control parameters, only a few researchers have offered guidance on setting values. Lee et al. (2005) recommended η between 0.7 and 0.95, ρ between 0.2 and 0.5, and μ between 10 and 50 on the basis of empirical observation. Degertekin (2008) noted that optimization results are sensitive to the parameter settings and recommended $\eta = 0.8$, $\rho = 0.4$, and μ in the range of 50 to 100. Wang and Huang (2010), in a study of optimization test functions, recommended large values of η, an adaptive strategy for ρ, and $\mu = 50$. For harmony memory size, $\mu = 50$ appears to have the strongest consensus for unimodal optimization. The examples presented here all use $\mu = 75$ on the basis of reasoning that multimodal optimization should use a larger population to develop a diverse range of solutions. Concerning η and ρ, as discussed previously, all examples use the adaptive strategy proposed by Hasancebi et al. (2010), with initial values of $\eta = 0.8$ and $\rho = 0.2$.

10-Bar Truss Example

Problem Definition

The 10-bar, 6-node truss shown in Fig. 1 has been widely published in the optimization literature. The great majority of studies have considered only member-size optimization, but some have also considered structural shape and topology in addition to member

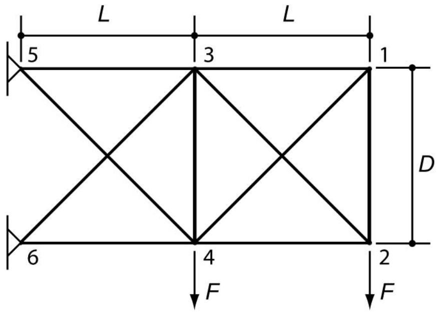


Fig. 1. The 10-bar truss example

sizes. The following discussion presents the problem as studied by Balling et al. (2006) and Rajan (1995).

In this example, $L = 360$ in (9.14 m) and $F = 100$ kip (445 kN). Stress was constrained to be less than 25 ksi (172 MPa). Vertical displacements at joints 2 and 4 were constrained to less than 2 in. (5.08 cm). The modulus of elasticity was 10,000 ksi (68.9 GPa). The material density was $0.1728\ \text{kip/ft}^3$ ($27.1\ \text{kN/m}^3$). Each of the 10 members had a section selected from 32 discrete cross-sectional areas given by Rajan (1995). Shape is optimized by allowing the vertical coordinates of joints 1, 3, and 5 to range from $D = 180$ in. (4.57 m) to $D = 1{,}000$ in. (25.4 m). Topology is optimized by allowing all members to be removed except the member between joints 6 and 4 and the member between joints 4 and 2. The member removal rate was $\alpha = 0.2$. For the eight members that were removable, the corresponding section variables were not considered in the calculation of distance between solutions, as the value of such a section variable has no bearing on a solution when its corresponding member is removed. Note that the topologic variables for removable members are included in the computation of distance.

Although Rajan (1995) and Balling et al. (2006) both studied this problem, there are important differences in their analyses. Rajan used population size $P = 40$ and number of generations $G = 96$, for a total of 3,840 model evaluations. In contrast, Balling et al. used $P = 1{,}000$, and $G = 500$, for a total of 500,000 model evaluations. Part of the difference in results is from different objectives. The Rajan study was unimodal, whereas the Balling et al. study was multimodal. The study presented here was configured for computational demand on a scale similar to the Rajan study, while attempting to achieve most of the multimodal benefits achieved by the Balling et al. study.

Results

The goal of multimodal optimization is to produce a diverse range of high-quality alternatives. For this example, one measure of success toward that goal is the number of different topologies generated and the fitness of the best instance of each. Fig. 2 shows three of the topologies typically generated during an optimization run; the line thickness in the figures is drawn proportional to the cross-section area of the truss member. Fitness results are reported in normalized form as fitness divided by the best result found by conventional harmony search (FH-GR), which was 3.06 kips (13.6 kN). Table 1 summarizes the normalized fitnesses for the best topologies found by each strategy; all values in the table were averaged over 10 runs.

Consider first the effect of close-harmony improvisation. Comparing the methods that use global replacement (FH-GR and CH-GR), close harmony found more feasible topologies than full harmony (an average of 8.2 compared with 5.4) and improved fitness for each of the top six topologies. Comparing the methods that

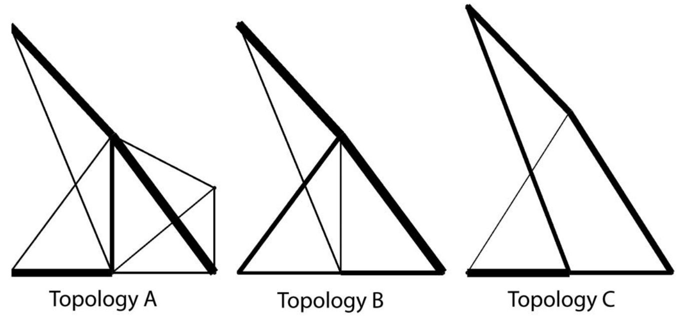


Fig. 2. Examples of selected topologies commonly found in the 10-bar truss problem; the member thickness is drawn proportional to the cross-sectional area

Table 1. Average Number of Feasible Topologies and Average Best Normalized Fitness for Top Six Topologies

Method	Avg. number of feasible topologies	Normalized fitness: 1st topo.	2nd topo.	3rd topo.	4th topo.	5th topo.	6th topo.
FH-GR	5.4	1.00	1.12	1.25	1.50	1.89	2.38
FH-LR	6.7	1.05	1.10	1.18	1.26	1.57	2.00
CH-GR	8.2	0.993	1.09	1.18	1.37	1.64	1.89
CH-LR	9.9	1.02	1.08	1.16	1.22	1.29	1.45

used local replacement (FH-LR and CH-LR), the same trend held; close-harmony again produced more feasible topologies (an average of 9.9 compared with 6.7), and the fitness of each of the top six topologies was better than those found by full harmony. For both replacement strategies, close harmony produced better results than full harmony in both the fitness and the diversity.

The effects of local replacement compared with global replacement are less clearly positive. For both the full-harmony and close-harmony cases, local replacement found lower quality results for the first-ranked topology than did global replacement: 4.57% worse for the case of full harmony and 2.30% worse for the case of close harmony. For the second-ranked topology, however, local replacement found slightly better quality (2.08%) for the full-harmony case and essentially equal quality for the close-harmony case. For the third- through sixth-ranked topologies, local replacement found successively better quality, by more than 20% for the 5th and 6th ranked topologies.

Note that the study of Balling et al. (2006) found 26 feasible topologies and a best weight of 2.736 kips (12.17 kN) for a solution with topology C from Fig. 2. Those results are significantly better than the results produced by any of the strategies here, in which the best average over 10 runs is 3.04 kips (13.5 kN). The Balling et al. study, however, used 500,000 model evaluations per run, rather than the 4,075 used in this study (75 evaluations on initialization of harmony memory plus 4,000 design cycles), so it is reasonable to expect a better result. The study of Rajan (1995) found a best weight of 3.254 (14.47 kN) kips by using 3,840 model evaluations. The CH-GR method, mentioned previously, which is most effective for unimodal optimization, produced a best result in its 10 runs of 2.90 kips (12.9 kN) and a worst result of 3.16 kips (14.1 kN); these are better results than those of the Rajan study with a comparable number of model evaluations. The CH-LR method, which is most effective for multimodal optimization, produced a best result in 10 runs of 2.91 (12.9 kN) kips and a worst result of

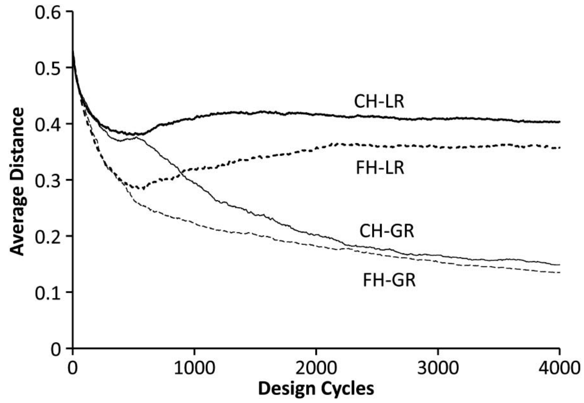


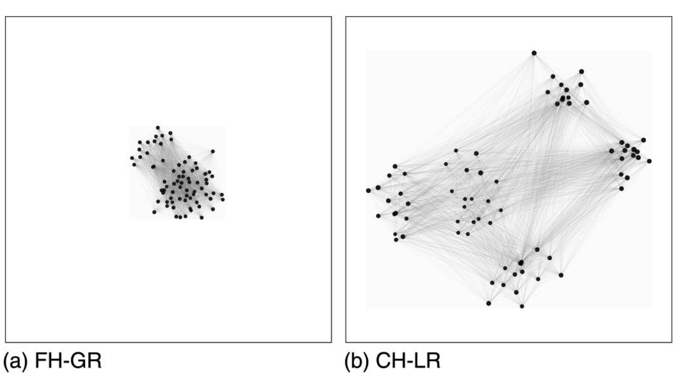
Fig. 3. Average distance among solutions for the four search strategies

3.25 kips (14.5 kN); this is better on average than the result of the Rajan study and essentially equal in the worst case.

Measuring Diversity of Solutions

One measure of diversity is the average distance between all pairs of solutions. Fig. 3 plots the history of this average distance for each of the four methods. The FH-GR curve clearly shows the tendency of conventional harmony search to converge. The CH-GR curve shows that close-harmony improvisation provides higher average distance during the early cycles but eventually declines to be nearly equal to the FH-GR curve. Local replacement has a more significant impact on average distance. The FH-LR curve follows the FH-GR curve closely until about cycle 500, at which point it diverges to a trajectory in which the average distance is increasing rather than decreasing. The divergence occurs at this point because the local replacement mechanism is not invoked until the algorithm encounters a crowded or overcrowded neighborhood, which for this problem does not occur until about the 500th design cycle. Up until that point, FH-LR and FH-GR have the same behavior, as the curves reflect. The CH-LR and CH-GR curves show a similar pattern.

Another useful perspective on diversity is to consider the solutions in harmony memory as a graph of nodes and links, in which each solution is a node and the link lengths correspond to the distance between solutions. Such a graph will be called a design-distance graph. In general, it is not mathematically possible to generate a two-dimensional drawing of such a graph in which all the links are drawn to an accurate scaling of distance, but it is possible to generate an approximate drawing that minimizes the error in link lengths. Fig. 4 shows two such design–distance graphs: one for a selected run of the FH-GR strategy in part A; and one for a selected run of the CH-LR strategy in part B. The graphs were generated by using the neato software package in the Graphviz suite from AT&T Research Labs (AT&T Research 2010). The width of the square graph area corresponds to a distance of 1.0, and the diameter of each circle is proportional to the fitness of the corresponding solution. Part A of the figure graphically illustrates the tendency of the FH-GR method to converge. Part B illustrates the effect of local replacement in maintaining diversity among solutions by limiting the number of solutions in any neighborhood.



(a) FH-GR (b) CH-LR

Fig. 4. Design-distance graphs for two search methods; each dot represents a solution, and the length of the link between two dots is approximately proportional to the normalized Euclidean distance between the solutions; the CH-LR method produces many longer links, reflecting greater diversity among solutions

Basket-Handle Arch Bridge

Overview

To apply the application to a more realistic example, the following discussion considers the arch structure shown in Fig. 5, a simplified model of a pedestrian bridge with the arches in a basket-handle configuration. The span $L = 1{,}200$ in (30.5 m); the width $W = 96$ in (2.44 m). Cross sections are selected either from a list of 32 AISC HSS round sections or a list of 75 AISC HSS rectangular tube sections. The lists are compiled by selecting the lightest section for each shape variety. The model includes one geometric decision variable, the span-to-depth ratio of the arch (γ), which ranges from 4 to 12, plus the five section variables indicated in Fig. 5 as follows:

Rib: The main arch rib, a round HSS section.
Brace: Members that connect the two arches near the crown, a round HSS section.
Hanger: The suspenders which transfer load from the deck to the arch, a round HSS section.
Tie beam: The longitudinal beams at the deck level, a rectangular HSS section.
Transverse beam: The beams which span between the tie, a rectangular HSS section.

Concerning structural modeling, the structure uses a pin support for each arch at one end of the bridge and roller supports at the other. The hanger members are pin-ended, and the connections between the arches and the tie beams are also pinned. Concerning materials, the rectangular tube sections use a yield stress of 46 ksi (317 MPa), and the round sections use a yield stress of 42 ksi (290 MPa). Dead load includes the self weight of the model plus a superimposed dead load on the deck area of $0.08\ \text{kip}/\text{ft}^2$ (3.83 kPa). The live load is $0.85\ \text{k}/\text{ft}^2$ (4.07 kPa) distributed on the deck area. Superimposed dead and live loads are applied to the nodes of each transverse beam, at the beam ends, and a midspan node according to tributary area. The vertical deflection of the nodes of the tie beam are limited to $L/1{,}000$ for live load only. The analysis includes four load combinations: two to check stiffness and two to check strength and stability. The combinations for stiffness include one with full live load and one with live load on half the span; these combinations use linear analysis. The combinations for strength and stability include one with dead plus live and another with dead plus live load on half the span; these combinations are factored according to the AISC LRFD code (AISC 2001) and use large displacement analysis. Note that these load combinations are unrealistically simple, in particular because they do not account for lateral loads. Concerning stability criteria, the algorithm checks each member to account for member-level compression stability according to the AISC LRFD requirements discussed previously by using an effective length factor $K = 1.0$ and an unsupported

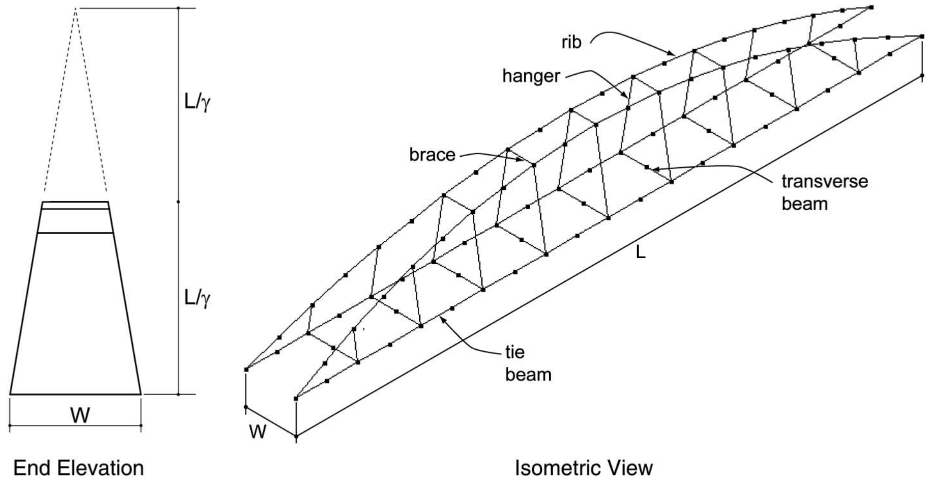


Fig. 5. Configuration of basket-handle bridge example

length equal to the member length. System-level stability was considered as described in the discussion of stability constraints.

Results

The motivation for this example is that it is multimodal. Optimization runs reveal two distinct solution types that result in good fitness. Examples of these types are shown in Figs. 6 and 7, and will be called the slender-arch type and the stiff-arch type. Although the stiff-arch solution type is clearly suboptimal in fitness, it could be preferred when aesthetics are considered. The challenge for a multimodal optimization algorithm is to find good solutions of both types so that the designer can be aware of the range of viable options.

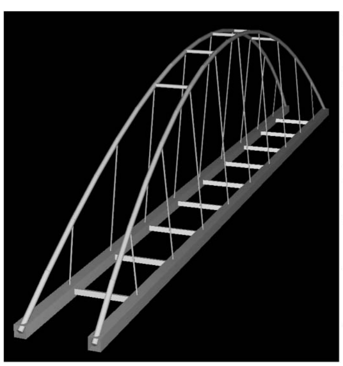


Fig. 6. A representative example of the slender-rib solution type

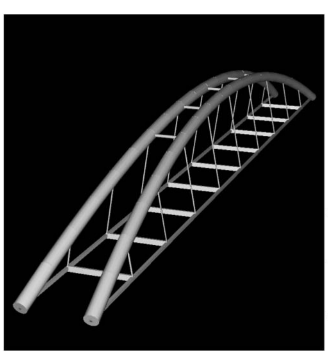


Fig. 7. A representative example of the stiff-rib solution type

Table 2 summarizes the results of the four methods. The reported fitnesses are normalized with respect to the average of the best result of the FH-GR method, 17.2 kips (76.5 kN). For each of the four methods, the table lists the average, best, and worst normalized fitness for the best solution found over the 10 runs. In addition, the table includes notes concerning whether the solutions at the conclusion of each run included only the stiff-rib type, only the slender-rib type, or included both types. The results show that global replacement has a strong tendency to converge. Both the FH-GR and CH-GR methods converged to the locally optimal stiff-rib solution type in five of the 10 runs, and FH-GR converged to the slender-rib solution type in four of the 10 runs. In contrast, the methods employing local replacement, FH-LR and CH-LR, concluded with both solution types in all of their respective runs. For this problem, it is clear that local replacement is superior to global replacement, not only for multimodal optimization but also for unimodal optimization.

Concerning the question of full-harmony versus close-harmony improvisation, the effect of close-harmony improvisation is mildly positive. The CH-GR method produces an average 1.40% percent better than FH-GR, and the CH-LR method produces an average 1.53% better than FH-LR. Because the global replacement strategy is inferior for this problem, the comparison of the full-harmony and close-harmony strategies focuses on the methods that employ local replacement. In the preceding discussion, Table 2 compared the fitness of the best solution found by each method; in considering multimodal optimization, it is also important to examine the fitness of locally optimal alternative solutions. Table 3 compares the best slender-rib solutions and the best stiff-rib solutions for the FH-LR

Table 2. Normalized Fitness Results for Ten Runs of Each Method

	Normalized fitness			
Method	Average	Best	Worst	Notes
FH-GR	1.00	0.927	1.03	5 runs concluded all stiff-rib type.
				1 run concluded all slender-rib type.
				4 runs concluded with both types.
FH-LR	0.978	0.953	1.03	All runs found both solution types.
CH-GR	0.986	0.939	1.03	5 runs concluded all stiff-rib type.
				4 runs concluded all slender-rib type.
				1 run concluded with both types.
CH-LR	0.963	0.907	0.996	All runs found both solution types.

Table 3. Comparison of Normalized Fitness for Each Solution Type

	Normalized fitness					
	Slender-rib solution			Stiff-rib solution		
Method	Avg.	Best	Worst	Avg.	Best	Worst
FH-LR	1.00	0.974	1.05	1.06	1.05	1.08
CH-LR	0.985	0.927	1.02	1.05	1.04	1.07

and CH-LR methods. The fitness values are normalized with respect to the average of the best result of the FH-LR method, 16.8 kips (74.8 kN). The results show that for the slender-rib solution, CH-LR produces better average, best, and worst results among the 10 runs. For the stiff-rib solution type, CH-LR produces average, best, and worst results about 1% better than FH-LR, a modest but consistent difference.

Conclusions and Future Work

The examples demonstrate the clear tendency of the global replacement strategy to converge toward one solution. In the 10-bar truss example, this tendency led to the positive result that the FH-GR and CH-GR strategies produced the best global optima for that problem. In the arch-bridge example, this tendency led to the negative result that both the CH-GR and FH-GR strategies converged to a locally optimal solution type in five of their 10 runs. In addition, even in the cases where global replacement led to a superior global optimum, local replacement always resulted in better locally optimal alternatives, making it a better strategy for multimodal optimization. The arch-bridge problem also demonstrated that local replacement can be an effective strategy for unimodal optimization, by avoiding convergence to a locally optimal solution. Concerning close-harmony improvisation, in both examples, close-harmony improvisation produced better fitness than full-harmony improvisation, although the improvements were modest in some cases. In addition, close harmony significantly improved diversity of solutions for the truss problem. In summary, local replacement has a profound effect on improving the diversity of the search result, whereas close-harmony improvisation has a consistent but sometimes mild effect on improving fitness.

Concerning future research, further study is needed on the question of adaptive methods for determining the close-harmony radius and the neighborhood radius. The examples presented here set the close-harmony radius equal to the average of the distance among solutions, but there is clearly opportunity to develop more sophisticated methods. For the neighborhood radius, the examples presented used 0.25 times the known feasible diameter $D_f^{(i)}$. This method is partially adaptive in that it is calculated on the basis of $D_f^{(i)}$ but also depends on the 0.25 factor, which is rather arbitrary. In addition to investigating such adaptive methods, future work will test these strategies on a wider range of structural optimization problems.

References

American Institute of Steel Construction. (2001). "Manual of steel construction: Load and resistance factor design." AISC, Chicago, IL.

AT&T Research. (2010). "Graphviz—Graph visualization software." ⟨http://www.graphviz.org/⟩ (Jun. 25, 2010).

Balling, R. J., Briggs, R. R., and Gillman, K. (2006). "Multiple optimum size/shape/topology designs for skeletal structures using a genetic algorithm." *J. Struct. Eng.*, 132(7), 1158–1165.

Deb, K. (2000). "An efficient constraint handling method for genetic algorithms." *Comput. Methods Appl. Mech. Eng.*, 186(2–4), 311–338.

Degertekin, S. O. (2008). "Harmony search algorithm for optimum design of steel frame structures: A comparative study with other optimization methods." *Struct. Eng. Mech.*, 29(4), 391–410.

Gao, X.-Z., Wang, X., and Ovaska, S. J. (2009). "Harmony search mothods for multimodal and constrained optimization." *Music-inspired harmony search algorithm*, Z. W. Geem, ed., Springer Verlag, Berlin, 39–51.

Geem, Z. W., Kim, J. H., and Loganathan, G. V. (2001). "A new heuristic optimization algorithm: Harmony search." *Simulation*, 76(2), 60–68.

Hasançebi, O., Çarbaş, S., Doğan, E., Erdal, F., and Saka, M. P. (2009). "Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures." *Comput. Struct.*, 87(5–6), 284–302.

Hasancebi, O., Ferhat, E., and Mehmet, P. S. (2010). "Adaptive harmony search method for structural optimization." *J. Struct. Eng.*, 136(4), 419–431.

Lee, K. S., Geem, Z. W., Lee, S. H., and Bae, K. W. (2005). "The harmony search heuristic algorithm for discrete structural optimization." *Eng. Optim.*, 37(7), 663–684.

Mahdavi, M., Fesanghary, M., and Damangir, E. (2007). "An improved harmony search algorithm for solving optimization problems." *Appl. Math. Comput.*, 188(2), 1567–1579.

Rajan, S. D. (1995). "Sizing, shape, and topology design optimization of trusses using genetic algorithm." *J. Struct. Eng.*, 121(10), 1480–1487.

von Buelow, P. (2007). "Advantages of evolutionary computation used for exploration in the creative design process." *J. Integr. Des. Process Sci.*, 11(3), 5–18.

Wang, C.-M., and Huang, Y.-F. (2010). "Self-adaptive harmony search algorithm for optimization." *Expert Syst. Appl.*, 37(4), 2826–2837.

Winslow, P., Pellegrino, S., Sharma, S. B., and Happold, B. (2008). "Free form grid structures." *Struct. Eng.*, 86(3), 19–20.