International Journal of Computing and Corporate Research
ISSN (Online) : 2249-054X
Volume 5 Issue 1 January 2015
International Manuscript ID : 2249054XV5I1012015-03

# EFFECTIVE SOLUTIONS AND ALGORITHMIC APPROACHES FOR SOLVING COMBINATORIAL OPTIMIZATION PROBLEMS USING METAHEURISTIC TECHNIQUES

*Vijayaraghavan A.*
*Professor*
*Department of Computer Science and Engineering*
*Shridevi Institute of Engineering and Technology*
*Karnataka, India*

## ABSTRACT

Metaheuristics are renowned to present very efficient elucidation to many of today's combinatorial optimization problems in engineering, industrial, economical and scientific domains such as transportation, bioinformatics, logistics, business etc. Scheduling, timetabling, vehicle routing, resource allocation are intelligently and successfully tackled with Metaheuristic approaches such as Simulated Annealing, Tabu Search, Ant Colony Optimization, Harmony Search, Scatter Search, Iterated Local Search. Metaheuristics present itself as highly promising choice for nearly-optimal solutions in reasonable time where exact approaches are not applicable due to extremely large running times or other limitations. Meta-heuristic is a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality. This paper highlights the various contemporary real life applications of Metaheuristics in the domain of industrial engineering and NP-hard problems.

**Keywords -** Metaheuristics, Methods, Industrial and Engineering Applications, Scope, Ant Colony Optimization

## INTRODUCTION

Metaheuristics can solve Combinatorial Optimization Problems, like cutting and packing, routing, network design, assignment, scheduling, or time-tabling problems, continuous parameter optimization problems, or the optimization of non-linear structures like neural networks or tree structures as they often appear in computational intelligence.

Evolutionary Algorithms (EAs), in particular, comprise a variety of related algorithms that are based on the processes of evolution in nature. In contrast to several other Metaheuristics, they work on a set of concurrent solutions and can easily be parallelized.

Especially the combination of evolutionary algorithms with problem-specific heuristics, local-search based techniques, approximation methods and exact techniques often make possible highly efficient optimization algorithms for many areas of application.

Metaheuristics are generally applied to problems for which there is no satisfactory problem-specific algorithm or heuristic; or when it is not practical to implement such a method. Most commonly used Metaheuristics are targeted to combinatorial optimization problems, but of course can handle any problem that can be recast in that form, such as solving boolean equations. In spite of overly-optimistic claims by some of their advocates, Metaheuristics are not a panacea, and their indiscriminate use often is much less efficient than even the crudest problem-specific heuristic, by several orders of magnitude.

### Main Features of a Good Metaheuristics

- Population intrinsic  parallelism
- Indirect Coding
- Cooperation adapted crossover
- Local search in solution space
- Diversity need to be controlled

- Easy to implement the restarts
- Randomness

**Commonly used metaheuristic methods**

- TS : Tabu search [Glover, 89 et 90]
- SA : Simulated annealing [Kirckpatrick, 83]
- TA : Threshold accepting [Deuck, Scheuer, 90]
- VNS : Variable neighborhood [Hansen, Mladenovi´c, 98]
- ILOCAL SEARCH : Iterated local search [Loren¸co et al, 2000]
- Genetic Algorithm : Genetic Algorithm, [Holland 1975 – Goldberg 1989]
- MA : Memetic Algorithm, [Moscatto 1989]
- Hybrid Genetic Algorithm
- Ant Colony Optimization, [Dorigo 1991]
- Scatter search, Laguna, Glover, [Marty 2000]

Innumerable variants and hybrids of these techniques have been proposed, and many more applications of Metaheuristics to specific problems have been reported. This is an active field of research, with a considerable literature, a large community of researchers and users, and a wide range of applications.

## CASE STUDY - SIMULATED ANNEALING AS AN EXCELLENT METAHEURISTIC TECHNIQUE

Simulated Annealing is commonly said to be the oldest among the metaheuristics and surely one of the first algorithms that had an explicit strategy to avoid local minima. The fundamental idea is to allow moves resulting in solutions of worse quality than the current solution (uphill moves) in order to escape from local minima. The probability of doing such a move is decreased during the search.

The name Simulated Annealing (SA) is taken from annealing in metallurgy, a well known technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat makes the atoms become unstuck from their initial positions (a local minimum of the internal energy) and stroll randomly through states of elevated energy; the slow cooling gives more chances of finding configurations with lower internal energy than the initial one.

Each step in the SA algorithm replaces the current solution by an arbitrary "nearby" solution, chosen with a probability which depends on the difference between the corresponding function values and on a global parameter T (called the temperature), that is gradually decreased during the process. The dependency is such that the current solution changes almost randomly when T is large, but increasingly "downhill" as T goes to zero.

The method was independently described by Scott Kirkpatrick, C. Daniel Gelatt and Mario P. Vecchi in 1983, and by Vlado Černý in 1985. The method is an adaptation of the Metropolis-Hastings algorithm, a Monte Carlo method to generate sample states of a thermodynamic system, invented by N. Metropolis et al. in 1953.

The table below shows the mapping of physical annealing to Simulated Annealing.

| **Thermodynamic Simulation** | **Combinatorial Optimization** |
|---|---|
| System States | Feasible Solutions |
| Energy<br>Change of State | Cost<br>Neighboring Solutions |
| Temperature | Control Parameter |
| Frozen State | Heuristic Solution |

**Table 1**: Relationship between physical annealing and Simulated Annealing

Using these mappings, any combinatorial optimization problem can be converted into an annealing algorithm.

The major advantage of SA over other methods is an ability to evade becoming trapped at local minima. This algorithm employs a random search, which not only accepts changes that decrease objective function, f, but also some changes that increase it. The latter are accepted with a probability

$$p = \exp(-\delta f/T)$$

where $\delta f$ is the increase in f and T is a control parameter.

The algorithm starts by generating an initial solution and by initializing the temperature parameter T. Then the following is repeated until the termination condition is satisfied: A solution s' from the neighborhood N(s) of the solution s is randomly sampled and it is accepted as new current solution depending on f(s), f(s') and T. s' replaces s if f(s') < f(s) or, in case f(s') >= f(s), with a probability which is a function of T and f(s') - f(s). The probability is generally computed following the Boltzmann distribution exp(-(f(s') - f(s))/T).
The temperature T is decreased during the search process, thus at the beginning of the search the probability of accepting uphill moves is high and it gradually decreases, converging to a simple iterative improvement algorithm. This process is analogous to the annealing process of metals and glass, which assume a low energy configuration when cooled with an appropriate cooling schedule. Regarding the search process, this means that the algorithm is the result of two combined strategies: random walk and iterative improvement. In the first phase of the search, the bias toward improvements is low and it permits the exploration of the search space; this erratic component is slowly decreased thus leading the search to converge to a (local) minimum. The probability of accepting uphill moves is controlled by two factors: the difference of the objective functions and the temperature. On the one hand, at fixed temperature, the higher the difference f(s')- f(s), the lower the probability to accept a move from s to s'. On the other hand, the higher T, the higher the probability of uphill moves.

## COMPONENTS IN SIMULATED ANNEALING

- Solution space
- Cost function
    - Determines how “good” a particular solution is

- Perturbation rules
  (Transforming a solution to a new one)
- Simulated Annealing engine
  - A variable T, analogous to temperature
  - An initial temperature $T_0$ ($T_0 = 40,000$)
  - A freezing temperature Tempfreezing (Tempfreezing = 0.1)
  - A cooling schedule ($T = 0.95 * T$)

Another variant of Simulated Annealing also exists with the name Adaptive simulated annealing (ASA), in which the algorithm parameters that control temperature schedule and random step selection are automatically adjusted with the advancement of algorithm. It makes the algorithm more efficient and less sensitive to user defined parameters than canonical Simulated Annealing.

## VLSI FLOORPLANNING USING SIMULATED ANNEALING

VLSI design is a method used to build electronic components - microprocessors and memory chips comprising millions of transistors. VLSI design is basically divided into number of phases. The first stage generates a set of indivisible rectangular blocks called cells. In the second stage, interconnection information is used to determine the relative placements of these cells. In the third stage, the goal of optimizing the total area is achieved using various techniques. This is the stage called Floorplan Optimization or simply floorplanning which is considered in this paper using a metaheuristic technique Simulated Annealing. Floorplanning is an important part of the design process, since its area usually dominates the cost of a chip. This paper highlights the the potential of a metaheuristic technique, Simulated Annealing to solve this optimization problem called VLSI floorplanning.

Floorplanning is important in VLSI (Very Large Scale Integrated circuit) design automation. VLSI is the process of creating integrated circuits by combining thousands of transistor-based circuits into a single chip. The VLSI design automation is one of the most computational expensive and complicated processes with significant impact into computer chips manufacturing. The floorplanning problem aims to arrange a set of rectangular modules on a rectangular chip area so as to optimize an appropriate measure of performance. This problem is known to be NP-hard, and is particularly challenging if the chip dimensions are fixed.

## FLOORPLAN PROBLEM

For a set of blocks B = {$b_1, b_2, \ldots, b_n$}, block $b_i$ is rectangular and having fixed width and height. The goals of floorplan optimization problem are to minimize the area of B and reduce wire lengths of interconnects subject to the constraints that no pair of blocks overlaps. Floorplanning minimizes a specified cost metric such as a combination of the area $A_{total}$ and wire length $W_{total}$ induced by the assignment of $b_i$'s, where $A_{total}$ is measured by the final enclosing rectangle of P and $W_{total}$ the summation of half the bounding box of pins for each net.

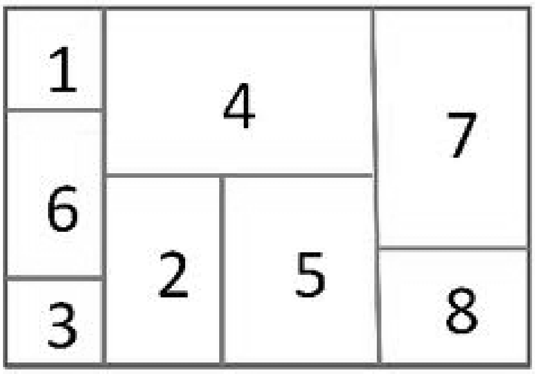
$$\text{Cost} = \alpha * A_{total} + \beta * W_{total}$$

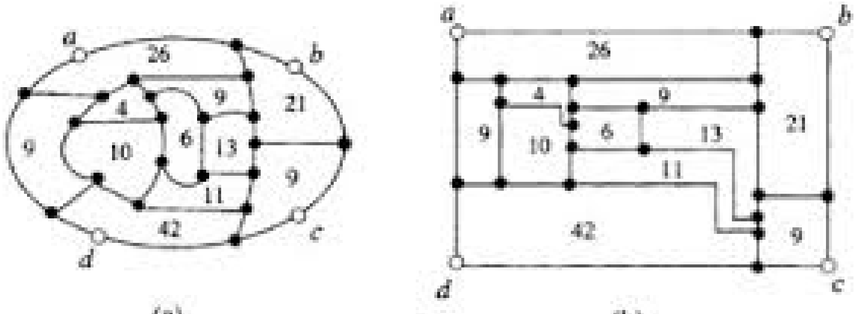
Where,

$A_{total}$ = Total area of the packing.

$W_{total}$ = Total wire length of packing.

$\alpha$ and $\beta$ = User specified constant.



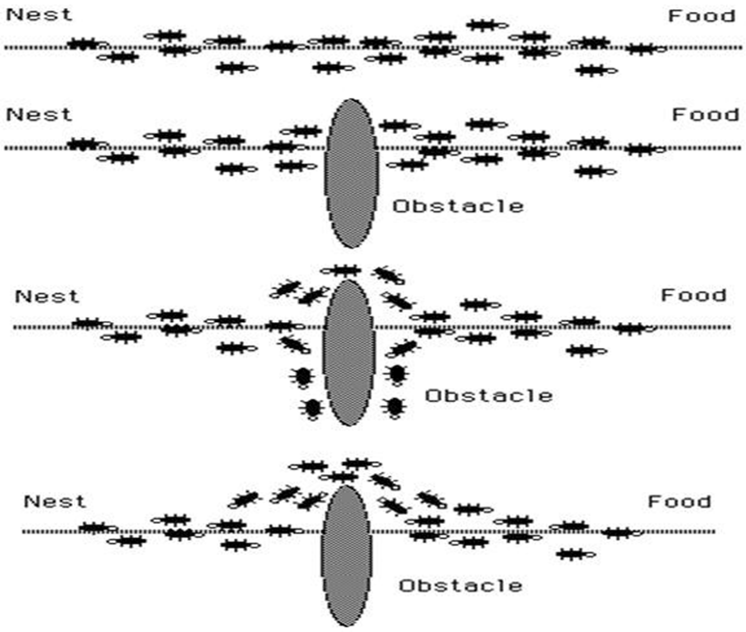


(a)
(b)

Figure 1 : VLSI Floorplan Layout

## ANT COLONY OPTIMIZATION

Ant Colony Optimization is one of the recognized techniques in Metaheuristics for solving computational problems which can be reduced to finding good paths through graphs. They are inspired by the behavior of ants in finding paths from the colony to food.



***[Figure 2 - Ant's Strategy for optimizing path to Food which further laid foundation of Ant Colony Optimization]***

Ant colony optimization algorithms have been used to produce near-optimal solutions to the traveling salesman problem. They have an advantage over simulated annealing and Genetic algorithm approaches when the graph may change dynamically; the ant colony algorithm can be run continuously and adapt to changes in real time. This is of interest in network routing and urban transportation systems.

Current applications of ACO algorithms fall into the two important problem classes of static and dynamic combinatorial optimization problems. Static problems are those whose topology and cost do not change while the problems are being solved. This is the case, for example, for the classic TSP, in which city locations and intercity distances do not change during the algorithm's run-time. Differently, in dynamic problems the topology and costs can change while solutions are built.

An example of such a problem is routing in telecommunications networks, in which traffic patterns change all the time.

## REAL LIFE INDUSTRIAL APPLICATIONS OF ANT COLONY OPTMIZATION

Ant Colony Optimization efficiently Solves NP hard Problems.  One of the best examples of an NP-hard problem is the decision problem **SUBSET-SUM** which is described as : given a set of integers, does any non empty subset of them add up to zero? It means a yes/no question, and happens to be NP-complete.

We can also elaborate it as : Given a set of integers, whether the sum of any non-empty subset equal exactly zero ? For example, given the set { 8, −4, −3, 1, -5}, the answer is YES because the subset { 8, −3, -5} sums to zero.

## FAMOUS REAL LIFE ENGINEERING AND INDUSTRIAL APPLICATIONS :

- Routing
  - TSP (Traveling Salesman Problem)
  - Vehicle Routing
  - Sequential Ordering
- Assignment
  - QAP (Quadratic Assignment Problem)
  - Graph Coloring
  - Generalized Assignment
  - Frequency Assignment
  - University Course Time Scheduling
- Scheduling

  - Job Shop
  - Open Shop
  - Flow Shop
  - Total tardiness (weighted/non-weighted)
  - Project Scheduling
  - Group Shop
- Subset
  - Multi-Knapsack
  - Max Independent Set
  - Redundancy Allocation
  - Set Covering
  - Weight Constrained Graph Tree partition
  - Arc-weighted L cardinality tree
  - Maximum Clique
- Other
  - Shortest Common Sequence
  - Constraint Satisfaction
  - 2D-HP protein folding
  - Bin Packing
- Machine Learning
  - Classification Rules
  - Bayesian networks
  - Fuzzy systems
- Network Routing
  - Connection oriented network routing
  - Connection network routing

  - Optical network routing
- For TSPs (Traveling Salesman Problem), relatively efficient
  - for a small number of nodes, TSPs can be solved by exhaustive search
  - for a large number of nodes, TSPs are very computationally difficult to solve (NP-hard) – exponential time to convergence

Large number of different ACO algorithms to exploit different problem characteristics

```
ANT COLONY OPTIMIZATION
Generate an initial colony
While stopping conditions are not met
    For each ant of the colony
        initialize ant memory
        While Curent State  = Target state
            Apply ant decision under pheromone information
            Move to next state
            Deposit pheromone information on the transition
        End While
    End For
    Evaporate pheromone information
End While
```

Solving SUDOKU Puzzles : Metaheuristics in Genetic Algorithmme Theory :
[Rhyd Lewis, Centre for Emergent Computing, Napier University, Scotland]

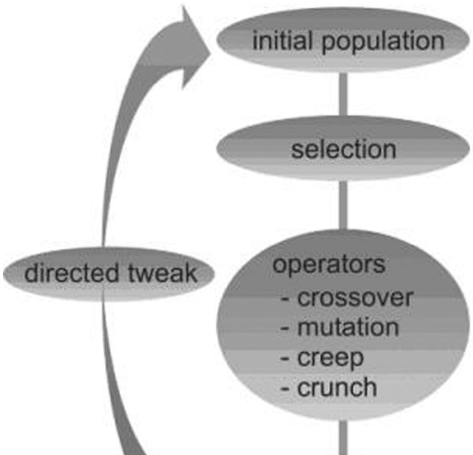
It was the first application of a metaheuristic technique to the popular sudoku puzzle. When applying this algorithm to a large number of problem instances taken from the UK press (of various degrees of ‘difficulty’). The algorithm does not just get close to optimality (as is often the case with optimization techniques), but consistently finds the solution in reasonably short amounts of time. It was proved that this method, particularly for lower order puzzles, is more robust than many existing algorithms, because in order to be successful, it does not necessarily depend on problem instances being logic-solvable.

Through a large number of experiments with the instance generator, the study witnessed the existence of easy-hard-easy style phase-transition in larger puzzles, which are similar to those found in other NP-complete problems such as constraint satisfaction problems (Smith, 1994), timetabling problems (Ross, Corne, and Terashima-Marin, 1996), and graph colouring problems (Turner, 1988).

Other techniques used for solving NP-Hard and other Computational Problems and optimization are :

## GENETIC ALGORITHMS

Genetic Algorithms maintains a pool of solutions rather than just one. The process of finding superior solutions mimics that of evolution, with solutions being combined or mutated to alter the pool of solutions, with solutions of inferior quality being discarded.



***[ Figure 3 - Genetic Algorithm ]***

## SIMULATED ANNEALING (SA)

**Simulated Annealing (SA)** is a related global optimization technique which traverses the search space by generating neighboring solutions of the current solution. A superior neighbor is always accepted. An inferior neighbor is accepted probabilistically based on the difference in quality and a temperature parameter. The temperature parameter is modified as the algorithm progresses to alter the nature of the search.

In the simulated annealing (SA) method, each point s of the search space is compared to a state of some physical system, and the function E(s) to be minimized is interpreted as the internal energy of the system in that state. Therefore the goal is to bring the system, from an arbitrary initial state, to a state with the minimum possible energy.

## TABU SEARCH (TS)

**Tabu search (TS)** is similar to Simulated Annealing, in that both traverse the solution space by testing mutations of an individual solution. While simulated annealing generates only one mutated solution, Tabu search generates many mutated solutions and moves to the solution with the lowest fitness of those generated. In order to prevent cycling and encourage greater movement through the solution space, a Tabu list is maintained of partial or complete solutions. It is forbidden to move to a solution that contains elements of the Tabu list, which is updated as the solution traverses the solution space.

Tabu Search is a powerful algorithmic approach that has been applied with great success to many difficult combinatorial problems. A particularly nice feature of TS is that, like all approaches based on Local Search, it can quite easily handle the dirty complicating constraints that are typically found in real-life applications. It is thus a really practical approach.

Tabu Search (TS) is basically a heuristic method which was originally proposed by Glover in 1986 for solving various combinatorial problems in the literature of operations research. In several cases, the methods described provide solutions very close to optimality and are among the most effective, if not the best, to tackle the difficult problems at hand. TS have been considered extremely popular in finding good solutions to the large combinatorial problems encountered in many practical settings. Several research papers and books have surveyed the rich TS literature.

Fred Glover pointed its research on allowing Local Search methods to overcome local optima. The basic principle of TS is to pursue LOCAL SEARCH whenever it encounters a local optimum by allowing non-improving moves; cycling back to previously visited solutions is prevented by the use of memories, called Tabu lists that record the recent history of the search, a key idea that can be linked to Artificial Intelligence concepts.

## HARMONY SEARCH (HS)

**Harmony search (HS)** is an algorithm based on the analogy between music improvisation and optimization. Each musician (variable) together seeks better harmonies (vectors).

**The Harmony Search has so far tackled the applications in various industrial fields as follows:**

**Civil Engineering**

- Water Network Design
- Dam Scheduling

**Structural Engineering**

- Dome Truss Design
- Grillage Structure Design
- Transmission Tower Design

**Traffic Engineering**

- School Bus Routing

**Mechanical Engineering**

- Pipeline Leakage Detection

**Energy Engineering**

- Pump Switching

**Space Engineering**

- Satellite Heat Pipe Design

**Geological Engineering**

- Soil Slope Stability

**Environmental Engineering**

- Flood Model Parameter Calibration

**Agricultural Engineering**

- Large-Scale IrriGenetic Algorithmtion Network Design

**Petroleum Engineering**

- Petroleum Structure Mooring

**Industrial Engineering**

- Fluid-Transport Minimal Spanning Tree
- Metaheuristics for meltshop scheduling in the steel industry
- Ant Colony Optimization to Solve Train Timetabling Problem of Mass Rapid Transit
- Designing Survivable Fiber-Optic Networks
- The material allocation problem in the steel industry

**Information Technology**

- Web-Based Hydrologic Optimization
- Use of Space Technology in Disaster Management

## CONCLUSION

However, lot of advancements has been pursued in finding exact solutions to the combinatorial optimization problems using techniques such as integer programming, dynamic programming, cutting planes, and branch and cut methods, still there are many hard combinatorial problems which are unsolved and require good heuristic methods. The goal of Metaheuristics is to produce good-quality efficient solutions without necessarily providing any guarantee of solution quality. Modern Metaheuristics include Simulated Annealing, Genetic Algorithms, Tabu Search, GRASP, ant colony optimization, and their hybrids. No doubt Metaheuristics have been one of the most stimulating topics to explore in the field of optimization.

## REFERENCES

[1] Fleurent, C. and J.A. Ferland (1996), "Genetic and Hybrid Algorithms for Graph Colouring", AnnaLocal Search of Operations Research **63**, 437-461.

[2] Gendreau, M. (2002), "Recent Advances in Tabu Search", in Essays and Surveys in Metaheuristics, C.C. Ribeiro and P. Hansen (eds.), Kluwer Academic Publishers, pp. 369-377.

[3] Hertz, A. and D. de Werra (1991), "The Tabu Search Metaheuristic: How We Used It", AnnaLocal Search of Mathematics and Artificial Intelligence **1**, 111-121.

[4] Kirkpatrick, S., C.D. Gelatt Jr. and M.P. Vecchi (1983), "Optimization by Simulated Annealing", Science **220**, 671-680.

[5] Laporte, G. and I.H. Osman (eds.) (1996), "Metaheuristics in Combinatorial Optimization", AnnaLocal Search of Operations Research **63**, J.C. Baltzer Science Publishers, Basel, Switzerland.

[6] Lokketangen, A. and F. Glover (1996), "Probabilistic Move Selection in Tabu Search for 0/1 Mixed Integer Programming Problems", in Meta-Heuristics: Theory and Applications, I.H. Osman and J.P. Kelly (eds.), Kluwer Academic Publishers, pp. 467-488.

[7] Osman, I.H. (1993), "Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem", AnnaLocal Search of Operations Research **41**, 421-451.

[8] Osman, I.H. and J.P. Kelly (eds.) (1996), Meta-Heuristics: Theory and Applications, Kluwer Academic Publishers, Norwell, MA.

[9] Pesant, G. and M. Gendreau (1999), "A Constraint Programming Framework for Local Search Methods", Journal of Heuristics **5**, 255-280.

[10] Ribeiro, C.C. and P. Hansen (eds.) (2002), Essays and Surveys in Metaheuristics, Kluwer Academic Publishers, Norwell, MA.

[11] Camp, Charles V., Bichon, Barron, J. and Stovall, Scott P. (2005) "Design of Steel Frames Using Ant Colony Optimization," Journal of Structural Engineeering, 131 (3):369-379.

[12] Fjalldal, Johann Bragi, "An Introduction to Ant Colony Algorithms." http://www.informatics.sussex.ac.uk/research/nlp/Genetic Algorithmzdar/teach/atc/1999/web/johannf/ants.html, accessed April 24, 2005.

[13] Geem, Z. W., "Optimal Cost Design of Water Distribution Networks using Harmony Search," Engineering Optimization, 38(3), 259-280, Apr. 1, 2006.

[14] Geem, Z. W., "Comparison Harmony Search with Other Meta-Heuristics in Water Distribution Network Design," Proceedings of 8th Annual International Symposium on Water Distribution Systems Analysis (WDSA 2006), ASCE, CD-ROM, Cincinnati, OH, USA, Aug. 27-30 2006.

[15] Lee, K. S. and Geem, Z. W., "A New Structural Optimization Method Based on the Harmony Search Algorithm," Computers & Structures, 82(9-10), 781-798, 2004.

[16] Erdal, F., Saka, M. P., "Optimum Design of Grillage Systems Using Harmony Search

[17] Algorithm," Proceedings of 8th International Conference on Computational Structures

[18] Technology (CST 2006), Las Palmas de Gran Canaria, Spain, Sept. 12-15, 2006.

[19] Geem, Z. W., Lee, K. S., and Tseng, C. -L., "Harmony Search for Structural Design,"

[20] Geem, Z. W., Lee, K. S., and Park, Y., "Application of Harmony Search to Vehicle Routing," American Journal of Applied Sciences, 2(12), 1552-1557, 2005.

[21] Kim, S. -H, Yoo, W. -S., Oh, K. -J., Hwang, I. -S., Oh, J. -E., "Transient Analysis and Leakage Detection Algorithm using GENETIC ALGORITHM and HS Algorithm for a Pipeline System," Journal of Mechanical Science and Technology, KSME, 20(3), 426-434, 2006.

[22] Geem, Z. W., "Harmony Search in Water Pump Switching Problem," Lecture Notes in Computer Science, 3612, 751-760, 2005.

[23] Geem, Z. W. and Hwangbo, H., "Application of Harmony Search to Multi-Objective Optimization for Satellite Heat Pipe Design," Proceedings of US-Korea Conference on Science, Technology, & Entrepreneurship (UKC 2006), CD-ROM, Teaneck, NJ, USA, Aug. 10-13 2006.

[24] Li, L., Chi, S. -C., Lin, G., "Genetic Algorithm Incorporated with Harmony Procedure and its Application to Searching of Non-Circular Critical Slip Surface in Soil Slopes," Shuili Xuebao, 36(8), Aug. 2005.