American Journal of Applied Sciences 2 (12):1552-1557, 2005
ISSN 1546-9239

Application of Harmony Search to Vehicle Routing

[1]Zong Woo Geem, [2]Kang Seok Lee and [3]Yongjin Park
[1]Environmental Planning and Management Program, Johns Hopkins University
729 Fallsgrove Drive #6133, Rockville, Maryland 20850, USA
[2]Advanced Structure Research Station, Hanyang University, Seoul, South Korea
[3]Department of Transportation Engineering, Keimyung University, Daegu, South Korea

Abstract: A phenomenon-inspired meta-heuristic algorithm, harmony search, imitating music improvisation process, is introduced and applied to vehicle routing problem, then compared with one of the popular evolutionary algorithms, genetic algorithm. The harmony search algorithm conceptualized a group of musicians together trying to search for better state of harmony. This algorithm was applied to a test traffic network composed of one bus depot, one school and ten bus stops with demand by commuting students. This school bus routing example is a multi-objective problem to minimize both the number of operating buses and the total travel time of all buses while satisfying bus capacity and time window constraints. Harmony search could find good solution within the reasonable amount of time and computation.

Key words: Harmony search, genetic algorithm, vehicle routing

INTRODUCTION

Traffic researchers and engineers sometimes cope with situations where optimization is needed to find the optimal solution out of huge amount of combinatorial solutions. Traditionally efficient enumeration methods such as branch and bound (B&B) technique have been used for supporting these optimal decisions. However, their computational disadvantages such as requiring huge amount of computation and memory made people rely on another type of methodology, that is, evolutionary or meta-heuristic algorithms.

The basic ideas of existing evolutionary and meta-heuristic algorithms are motivated by natural phenomena. For example, the evolutionary algorithms[1-3] and the genetic algorithm (GA)[4,5] are inspired by biological evolutionary process; tabu search[6] and ant algorithm[7] from animal's behavior; and simulated annealing[8] from physical annealing process.

Harmony Search (HS) algorithm has been recently developed in an analogy with music improvisation process where musicians in an ensemble continue to polish their pitches in order to obtain better harmony[9]. The HS algorithm has been successfully applied to various real-world combinatorial optimization problems such as truss structure design, pipe network design, pump switching, and hydrologic parameter calibration[10-13]. Especially, for the pump switching problem which has 2^{40} ($\approx 1.1\times 10^{12}$) different combinations, HS found the less energy consuming operation than GA or even than B&B methods which were implemented as an IBM subroutine or MS-Excel Solver. Consequently, the HS algorithm provides a possibility of success in a combinatorial optimization problem in traffic engineering field.

In order to demonstrate the searching ability of HS, it is applied to a school bus routing problem (SBRP) which is a practical optimization problem. From a school's perspective, the SBRP aims to provide students with an efficient and equitable transportation service[14]. Many approaches have been developed to solve the SBRP[14-16]. This problem falls into a large class of problems called vehicle routing problem (VRP), in which a set of vehicles provides pickup, delivery or simply a service to customers dispersed in an area.

Generally, the wider the problem scope is, the harder the problem becomes in term of the solution technique such as B&B method. Because an exact method may take a very long time to obtain the optimal solution, people have turned to evolutionary or meta-heuristic algorithms such as GA that do not necessarily find the optimal solution, but tend to find good solutions within a reasonable amount of time[17]. Pattnaik *et al.*[18] proposed an optimization model of minimizing overall cost (both the operator's cost and user's) while determining a route configuration with a set of urban bus transit routes and associated frequencies using GA. In their approach, a set of candidate routes is first generated, and then GA is employed to find the optimal one. Chien *et al.*[19] introduced a GA model to the optimization of bus route and the corresponding headway while minimizing total cost, subject to geography, capacity, and budget constraints. Their results were validated by comparing with those obtained from exhaustive search algorithms. Jung and Haghani[20] used GA to solve a multi-vehicle

Corresponding Author: Zong Woo Geem, Environmental Planning and Management Program, Johns Hopkins University, 729 Fallsgrove Drive # 6133, Rockville, Maryland 20850, USA

pickup and delivery problem with time window. They formulated the problem as a mixed-integer linear program, with an objective function to minimize the total cost, consisting of the fixed cost of the vehicles, routing cost, and customer inconvenient cost.

In this study, a recently-developed music phenomenon-inspired HS algorithm is applied to the test network of SBRP for the first time and compared with one of the most popular algorithms, GA.

HARMONY SEARCH ALGORITHM

From the idea that existing evolutionary or meta-heuristic algorithms imitate natural, scientific, or behavioral phenomena around us, a new algorithm can be conceptualized from a music performance process (say, a jazz improvisation) involving searching for a better harmony. Just like music improvisation seeks a best state (fantastic harmony) determined by aesthetic estimation, optimization process seeks a best state (global optimum) determined by objective function evaluation; Just like aesthetic estimation is determined by the set of the pitches played by ensembled instruments, function evaluation is determined by the set of the values assigned for decision variables; Just like aesthetic sound quality can be improved practice after practice, objective function value can be improved iteration by iteration.

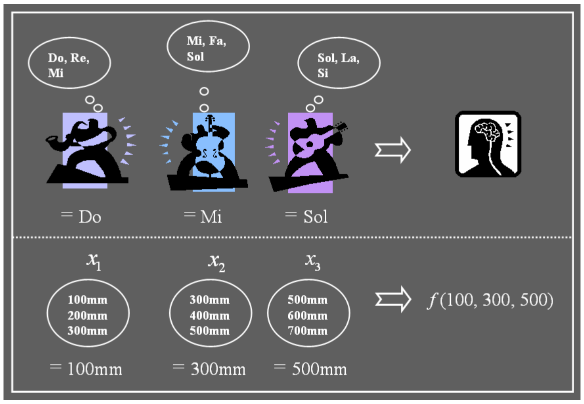

Fig. 1: Analogy between improvisation and Optimization

The analogy between improvisation and optimization is shown in Fig. 1. Each music player (saxophonist, double bassist, and guitarist) can correspond to each decision variable (x_1, x_2, and x_3), and the range of each music instrument (saxophone = {Do, Re, Mi}; double bass = {Mi, Fa, Sol}; and guitar = {Sol, La, Si}) corresponds to the range of each variable value (x_1 = {100, 200, 300}; x_2 = {300, 400, 500}; and x_3 = {500, 600, 700}). If the saxophonist toots the note Do, the double bassist plucks Mi, and the guitarist plucks Sol, their notes together make a new harmony (Do, Mi, Sol). If this new harmony is better than existing harmony, the new harmony is kept. Likewise, the new solution vector (100mm, 300mm, 500mm) generated in optimization process is kept if it is better than existing harmony in terms of objective function value. Just as the harmony quality is enhanced practice after practice, the solution quality is enhanced iteration by iteration.

According to the above algorithm concept, the HS algorithm consists of the following five steps: parameter initialization; harmony memory initialization; new harmony improvisation; harmony memory update; and termination criterion check.

Parameter initialization: In the first step, the optimization problem is specified as follows:

Minimize (or Maximize) $f(\boldsymbol{x})$ (1)

Subject to $x_i \in \mathbf{X}_i,\ i = 1,2,\ldots,N$ (2)

where $f(\cdot)$ is an objective function; $\boldsymbol{x}$ is a solution vector composed of each decision variable x_i; $\mathbf{X}_i$ is the set of possible range of values for each decision variable, that is, $\mathbf{X}_i = \{x_i(1), x_i(2), \ldots, x_i(K)\}$ for discrete decision variables (Generally, candidate values are sorted like $x_i(1) < x_i(2) < \ldots < x_i(K)$. However, there is no sorting required in this vehicle routing problem because each variable value represents just node number rather than physical amount such as node demand); N is the number of decision variables (= number of music instruments); and K is the number of candidate values for the discrete decision variables (= number of pitches for each instrument).

The HS algorithm parameters are also specified in this step: HMS (harmony memory size; = number of simultaneous solution vectors in harmony memory), HMS (harmony memory considering rate), PAR (pitch adjusting rate), and number of improvisations (= number of objective function evaluations). These algorithm parameters are explained in the following steps.

Harmony memory initialization: In Step 2, the Harmony Memory (HM) as shown in Equation 3, is crammed with as many randomly generated solution vectors as the size of the HM (i.e., HMS).

$$\left[\begin{array}{ccccc|c} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 & f(\boldsymbol{x}^1) \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 & f(\boldsymbol{x}^2) \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{N-1}^{HMS-1} & x_N^{HMS-1} & f(\boldsymbol{x}^{HMS-1}) \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_{N-1}^{HMS} & x_N^{HMS} & f(\boldsymbol{x}^{HMS}) \end{array}\right] \quad (3)$$

New harmony improvisation: A new harmony vector, $\boldsymbol{x}' = (x_1', x_2', \ldots, x_N')$ is improvised by following three rules: (1) random selection, (2) HM consideration, and (3) pitch adjustment.

Random selection: As a musician plays any pitch within the instrument range (for example, {Do, Re, Mi, Fa, Sol, La, Si} in Fig. 2), the value of decision variable x_i' is randomly chosen within the value range $\mathbf{X}_i$.

HM Consideration. As a musician plays any pitch out of the preferred pitches in his/her memory (for example, {Do, Mi, Do, Sol, Do} in Fig. 3), the value of decision variable x_i' is chosen from any pitches stored in HM ({ $x_i^1, x_i^2, \ldots, x_i^{HMS}$ }) with a probability of HMCR ($0 \le \text{HMCR} \le 1$) while it is randomly chosen with a probability of (1-HMCR) in random selection process as previously described.

$$x_i' \leftarrow \begin{cases} x_i' \in \{x_i^1, x_i^2, ..., x_i^{HMS}\} & w.p. \quad HMCR \\ x_i' \in \boldsymbol{X}_i & w.p. \quad (1\text{-}HMCR) \end{cases} \quad (4)$$

Pitch Adjustment. Once one pitch is obtained in HM consideration, a musician can further adjust the pitch to neighboring pitches (for example, the note Sol can be adjusted to Fa or La as shown in Fig. 4) with a probability of HMCR × PAR ($0 \le \text{PAR} \le 1$) while the original pitch obtained in HM consideration is just kept with a probability of HMCR × (1-PAR).

$$x_i' \leftarrow \begin{cases} x_i(k+m) & \text{w.p.} \quad HMCR \times PAR \\ x_i' & \text{w.p.} \quad HMCR \times (1-PAR) \end{cases} \quad (5)$$

where x_i' obtained in HM consideration and $x_i(k)$ (the k$^{\text{th}}$ element in $\mathbf{X}_i$) are identical; m ($m \in \{..., -2, -1, 1, 2, ...\}$) is a neighboring index used for discrete decision variables (m has normally +1 or -1).

Violated harmony consideration: Once the new harmony $\boldsymbol{x}' = (x_1', x_2', ..., x_N')$ is obtained using the above-mentioned three rules, it is then verified whether it violates problem constraints. Although the new harmony violates the constraints, it can be still kept in HM by taxing penalty, just as rule-violated harmony (for example, parallel fifth violation in Fig. 5) was still used in composition by the famous composers such as Ludwig van Beethoven. This technique is used for considering constraints in this study.

Harmony memory update: If the new harmony vector, $\boldsymbol{x}' = (x_1', x_2', ..., x_N')$ is better than the worst harmony in the HM, judged by objective function value, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

Termination criterion check: If the termination criterion (number of improvisations) is reached, computation is stopped. Otherwise, Steps 3 and 4 are repeated.

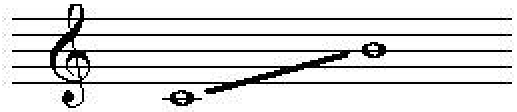
Fig. 2: Range of music instrument

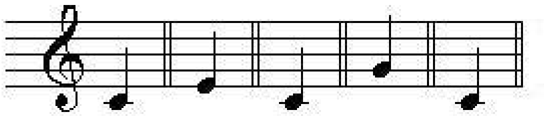
Fig. 3: Preferred pitches stored in harmony memory

Fig. 4: Pitch adjustment

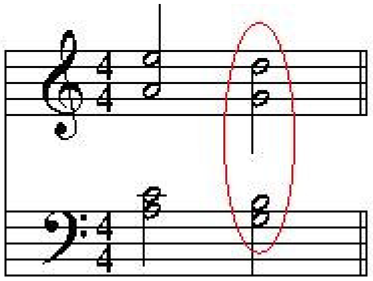
Fig. 5: Harmony violating parallel fifth

SCHOOL BUS ROUTING PROBLEM

The music-inspired HS algorithm is applied to school bus routing problem which is a multi-objective problem to minimize both the number of operating buses and the travel time of all buses, with two major constraints (bus capacity and time window). The study network to be optimized consists of one bus depot, one school, and ten bus stops as shown in Fig. 6. Each bus stop is demanded by certain number of commuting students, and travel time (in minutes) between two stops is specified in the figure.

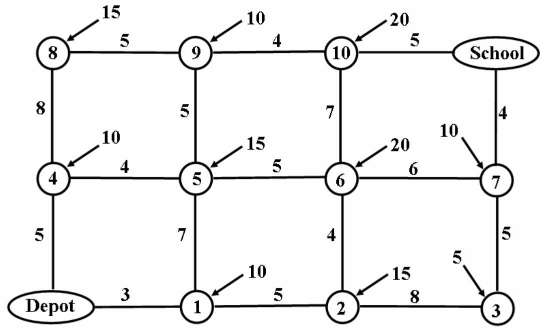

Fig. 6: Diagram of school bus routing network

Decision variables: Decision variables and other variables used in this study are as follows:

x_i = decision variable representing demand node (bus stop) i, which has bus k as a variable value, $i \in DN$, $k \in VS$. In this study, there are ten nodes and four candidate buses.

$\boldsymbol{x}$ = set of decision variables x_i, $i \in DN$.

$nbus(\boldsymbol{x})$ = number of operating buses. The number is equal to or less than the number of candidate buses.

$$lk_{ij}^k = \begin{cases} 1 & \text{if bus } k \text{ travels node } i \text{ and node } j \\ 0 & \text{otherwise} \end{cases},$$

$k \in VS, i \in STDN, j \in DNED$.

$$vcp^k = \begin{cases} 1 & \text{bus capacity violation in bus } k \\ 0 & \text{otherwise} \end{cases}, \ k \in VS.$$

$$vtm^k = \begin{cases} 1 & \text{time window violation in bus } k \\ 0 & \text{otherwise} \end{cases}, \ k \in VS.$$

Problem parameters: Problem parameters used in this study are as follows:

fc = fixed cost per school bus.

rc = routing cost (e.g., energy, labor, etc) per moving time.

sp_{ij} = shortest path between node i and node j. In this study, the shortest path sp_{ij} (in minutes) between node i and node j is calculated by Floyd and Warshall's algorithm.

$pc1$ = penalty cost for bus capacity violation.

$pc2$ = penalty cost for time window violation.

$nset(VS)$ = number of elements in set VS (number of candidate buses).

DM_i^k = number of boarding students on bus k in node i, $k \in VS$, $i \in DN$.

BC^k = bus capacity of bus k, $k \in VS$.

bt = boarding time per student.

TW^k = time window of bus k, $k \in VS$.

Sets: Sets used in this study are as follows:

DN = set of demand nodes (= bus stops).

ST = starting node (= bus depot).

ED = ending node (= school).

$STDN$ = union set of starting node and demand nodes ($ST \cup DN$).

$DNED$ = union set of demand nodes and ending node ($DN \cup ED$).

VS = set of vehicles.

Problem formulation:

$$\text{Minimize} \quad f(\boldsymbol{x}) = fc \times nbus(\boldsymbol{x}) + rc \times \sum_k \sum_{i \in STDN} \sum_{j \in DNED} sp_{ij} lk_{ij}^k + pc1 \times \sum_k vcp^k + pc2 \times \sum_k vtm^k \qquad (6)$$

$$\text{Subject to } nbus(\boldsymbol{x}) \le nset(VS) \qquad (7)$$

$$\sum_i DM_i^k \le BC^k, \quad k \in VS \qquad (8)$$

$$\sum_{i \in STDN} \sum_{j \in DNED} sp_{ij} lk_{ij}^k + \sum_i DM_i^k bt \le TW^k, \quad k \in VS \ (9)$$

While each decision variable x_i stands for each bus stop (demand node) i, and has a value of the specific bus number k ($k \in VS$) that serves the bus stop i, the objective function of the problem is to minimize both the total number of operating buses and the moving time of the buses as described in the first and the second terms of Equation 6. The third and fourth terms represent penalty costs for the violation of bus capacity and time window, respectively. The fixed cost per each operating bus, fc is assumed as \$100,000/bus; routing cost per moving time, rc is \$105/min; shortest path sp_{ij} (in minutes) between node i and node j is calculated by Floyd and Warshall's algorithm; connection status between node i and node j for bus k, lk_{ij}^k has 1 when bus k passes along from node i ($i \in STDN$) to node j ($j \in DNED$); penalty cost for every bus capacity violation, $pc1$ is \$100,000 when any bus carries more than 45 students; and penalty cost for every time window violation, $pc2$ is \$100,000 when any bus operates more than 32 minutes to convey students to the school.

Equation 7 represents the maximum number constraint for the operating buses, where the number of operating buses can be equal to or less than the number of candidate buses (= 4 buses in this study). Equation 8 shows that the number of boarding students in a bus is less than or equal to the bus capacity (= 45 students for each bus). Equation 9 shows that the travel time of a bus is less than or equal to the time window (= 32 minutes for each bus). Travel (in-vehicle) time in this study consists of both bus moving time and student boarding time, where each student's boarding time bt is 6 seconds.

COMPUTATION AND RESULTS

In order to apply the HS algorithm to the school bus routing problem, problem and algorithm parameters are specified: number of musical instruments (number of demand nodes) = 10; pitch range of each instrument (value range of each decision variable) = {bus 1, bus 2, bus 3, bus 4}; HMS = 10 ~ 100; HMCR = 0.3 ~ 0.95; PAR = 0.01 ~ 0.05; and stopping criterion = 1,000 improvisations. Next, harmonies (solution vectors) are randomly generated from the possible range as many as HMS. After that, a new harmony is improvised based on three rules (random selection, HM consideration, and pitch adjustment). The new harmony vector $\boldsymbol{x}'$ is then put to the objective function to obtain total cost

Table 1: Results of school bus routing problem

Routing Cost ($)	Bus #	Routes	# of Students	Travel Time (min)
307,980	1	Depot → 8 → 9 → 10 → School	45	31.5
	2	Depot → 4 → 5 → 6 → School	45	28.5
	3	Depot → 1 → 2→ 3 → 7 → School	40	29.0
	4	Do Nothing	-	-
410,185	1	Depot → 2→ 6 → School	35	25.5
	2	Depot → 1 → 3 → 7→ School	25	27.5
	3	Depot→ 5 → 9 → 10 → School	45	27.5
	4	Depot → 4→ 8→ School	25	29.5

which consists of fixed bus cost, bus travel cost, and two penalty costs. If bus k violates equation 8, the capacity violation variable, vcp^k , becomes 1, and the penalty cost for the capacity violation is added the objective function. If bus k violates equation 9, the time window violation variable, vtm^k , becomes 1, and the penalty cost for the time window violation is added to the objective function. If the total cost for the new harmony is better than the cost of any harmony stored in the HM, the new harmony is included in the HM, and the existing harmony with the worst cost is excluded from the HM. The new harmony is continuously improvised until the stopping criterion is satisfied.

When applied to the school bus routing problem, HS algorithm could find the global optimum ($307,980) only after 1,000 improvisations (or function evaluations), which was examined by the total enumeration for this 4^{10} ($\approx 1.05 \times 10^6$) combinatorial problem. It took 6.6 seconds on Intel 233MHz CPU to search 0.1% of total solution space (= $10^3 / 1.05 \cdot 10^6$). Table 1 shows the optimal route, number of commuting students, and travel time for the global optimal solution and near-optimal solution. "Do Nothing" in 5th row of the table means that the bus 4 does not have to be operated because only three buses can serve all the students in the problem.

HS results were also examined by comparing with those of popular evolutionary technique, GA. In order to fairly compare HS with GA, the number of objective function evaluations and the number of computational runs are same in both algorithms: in HS, the number of function evaluations is 1,000 and the number of runs is 20 with different HMS, HMCR, and PAR; and in GA, the number of function evaluations is 1,000 (= population size × number of generations) and the number of runs is 20 with different population size, crossover rate, and mutation rate, recommended by Koumousis and Georgiou[21].

After being applied to the test routing problem, both algorithms could find the global optimum solution. However, HS reached it twice while GA did it once out of 20 different runs. The average costs are $399,870 in HS, and $409,597 in GA, respectively. In addition, HS performed each run slightly faster than GA on the same machine.

CONCLUSION

A recently-developed music phenomenon-inspired algorithm, HS was introduced and modeled for solving the school bus routing problem. The objective of proposed HS model for the school bus routing is to minimize the total cost of multi-objective function which consists of bus operating cost, bus travel time, and penalties related with bus capacity and time window violations. HS model could find global optimum within far less function evaluations comparing with total enumeration. HS model also found better solution than GA in terms of number of reaching global optimum, average cost out of multiple runs, and computing time.

From these results, the HS algorithm appears to have a potential to be successfully applied to combinatorial problems in traffic engineering field. Especially, the presented bus routing model is expected to be applied to large-scale routing networks for the future study.

REFERENCES

1. Fogel, L.J., A.J. Owens and M.J. Walsh, 1966. Artificial Intelligence Though Simulated Evolution. John Wiley, Chichester, UK.
2. De Jong, K., 1975. Analysis of the Behavior of a Class of Genetic Adaptive Systems. Ph.D. Thesis, University of Michigan, Ann Arbor, MI.
3. Koza, J.R., 1990. Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems. Report No. STA-CS-90-1314, Stanford University, Stanford, CA.
4. Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI.
5. Goldberg, D.E., 1989. Genetic Algorithms in Search Optimization and Machine Learning. Addison Wesley, MA.
6. Glover, F., 1977. Heuristic for Integer Programming using Surrogate Constraints. Decision Sciences, 8: 156-166.
7. Dorigo, M., V. Maniezzo, and A. Colorni, 1996. The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Trans. on Systems, Man, and Cybernetics, Part B, 26: 29-41.

8. Kirkpatrick, S., C. Gelatt, and M. Vecchi, 1983. Optimization by Simulated Annealing. Science, 220: 671-680.
9. Geem, Z.W., J.H. Kim and G.V. Loganathan, 2001. A new heuristic optimization algorithm: Harmony search. Simulation, 76: 60-68.
10. Lee, K.S. and Z.W. Geem, 2004. A new structural optimization method based on the harmony search algorithm. Computers and Structures, 82: 781-798.
11. Geem, Z.W., In Press. Optimal cost design of water distribution networks using harmony search. Engineering Optimization.
12. Geem, Z.W., 2005. Harmony search in water pump switching problem. Lecture Notes in Computer Science, 3612: 751-760.
13. Kim, J.H., Z.W. Geem and E.S. Kim, 2001. Parameter estimation of the nonlinear muskingum model using harmony search. J. Am. Water Resources Assoc., 37: 1131-1138.
14. Bowerman, R., B. Hall and P. Calamai, 1995. A multi-objective optimization approach to urban school bus routing: Formulation and solution method. Transport. Res. A, 29A: 107-123.
15. Bennett, B.T. and D.C. Gazis, 1972. School bus routing by computer. Transport. Res., 6: 317-325.
16. Tsay, H.-S. and J.D. Fricker, 1988. Practical approach for solving school bus routing. Transport. Res. Rec., TRB., 1202: 44-56.
17. Bodin, L., B. Golden, A. Assad and M. Ball, 1983. The state of art in the routing and scheduling of vehicle and crews. Comp. Oper. Res., 10: 63-211.
18. Pattnaik, S.B., S. Mohan and V.M. Tom, 1998. Urban bus transit route network design using genetic algorithm. J. Transport. Engg. ASCE., 124: 368-375.
19. Chien, S., Z. Yang and E. Hou, 2001. Genetic algorithm approach for transit route planning and design. J. Transport. Engg. ASCE., 127: 200-207.
20. Jung, S. and A. Haghani, 2000. Genetic algorithm for a pickup and delivery problem with time windows. Transport. Res. Rec. TRB., 1733: 1-7.
21. Koumousis, V.K. and P.G. Georgiou, 1994. Genetic algorithms in discrete optimization of steel truss roofs. J. Comp. in Civil Engg. ASCE., 8: 309-325.