School Bus Routing using Harmony Search

Zong Woo Geem
Johns Hopkins University
729 Fallsgrove Drive #6133
Rockville, MD 20850
+1-301-294-3893
geem@jhu.edu

ABSTRACT

A recently-developed nature-inspired algorithm, Harmony Search, mimicking music improvisation, is introduced and applied to transportation problem (school bus routing), and compared with popular evolutionary algorithm (genetic algorithm). The Harmony Search is conceptualized using the musical process of searching for a perfect state of harmony. This algorithm was applied to a test network consisting 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 the number of operating buses and also the travel time of all buses, with bus capacity and time window constraints that are considered as penalty costs. Harmony Search could find good solution within the reasonable time with other advantages such as no derivative requirement and no initial value assumption. The presented routing model is expected to be applied to large-scale real networks in the future.

Categories and Subject Descriptors

J.2 [Computer Applications]: Physical Science s and Engineering – *Engineering*

General Terms

Algorithms, Design, Economics

Keywords

Harmony Search, School Bus Routing, Optimization

1. INTRODUCTION

Transportation researchers and professionals are sometimes coped with situations where optimal decisions need to be made. Traditionally various mathematical techniques have been used for supporting these optimal decisions. However, their computational disadvantages such as requiring derivative information, initial value assumption, or huge amount of computation and memory, made them rely on another type of methodology, that is, evolutionary or meta-heuristic algorithms.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GECCO 2005, June 25–29, 2005, Washington, DC, USA.
Copyright 2004 ACM 1-58113-000-0/00/0004…$5.00.

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 [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 was recently developed in an analogy with music improvisation process where musicians in an ensemble adjust the pitches of their instruments in order to obtain perfect harmony [9, 10]. The HS algorithm has been successfully applied to both benchmarking problems and real-world problems such as the traveling salesperson problem [9], Rosenbrock's function [10], optimal design of pipeline network [11], parameter calibration of hydrologic model [12], and optimal design of truss structures [13, 14]. Consequently, the HS algorithm provides a possibility of success in an optimization problem in transportation research field.

2. HARMONY SEARCH ALGORITHM

Adopting the idea that existing evolutionary or meta-heuristic algorithms are found in the paradigm of natural processes, a new algorithm can be conceptualized from a musical performance process (say, a jazz trio) involving searching for a better harmony. Musical performance seeks a best state (fantastic harmony) determined by aesthetic estimation, as the optimization process seeks a best state (global optimum: minimum cost; minimum error; maximum benefit; or maximum efficiency) determined by objective function evaluation. Aesthetic estimation is done by the set of the pitches sounded by joined instruments, as objective function evaluation is done by the set of the values produced by composed variables; the aesthetic sounds can be improved practice after practice, as the objective function values can be improved iteration by iteration.

Figure 1 shows the structure of the Harmony Memory (HM) that is the core part of the HS algorithm. Consider a jazz trio composed of saxophone, double bass, and guitar. There exist certain amount of preferable pitches in each musician's memory: saxophonist, {Do, Fa, Mi, Sol, Re}; double bassist, {Si, Do, Si, Re, Sol}; and guitarist, {La, Sol, Fa, Mi, Do}. If saxophonist randomly plays {Sol} out of its memory {Do, Fa, Mi, <u>Sol</u>, Re}, double bassist {Si} out of {<u>Si</u>, Do, Si, Re, Sol}, and guitarist {Do} out of {La, Sol, Fa, Mi, <u>Do</u>}, the new harmony (Sol, Si, Do) becomes another harmony (musically C-7 chord). And if this new harmony is better than existing worst harmony in the HM, the new harmony is included in the HM and the worst harmony is excluded from the HM. This procedure is repeated until fantastic harmony is found.

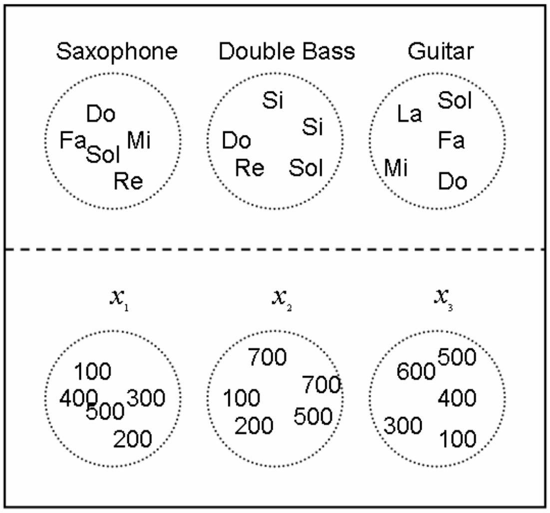

Figure 1. Structure of Harmony Memory

In real optimization, each musician can be replaced with each decision variable, and its preferred sound pitches can be replaced with each variable's preferred values. Let us set that each decision variable represents pipe diameter between two nodes and the music pitches {Do, Re, Mi, Fa, Sol, La, Si} correspond to pipe diameters {100mm, 200mm, 300mm, 400mm, 500mm, 600mm, 700mm}. And if first variable chooses {500mm} out of {100mm, 400mm, 300mm, <u>500mm</u>, 200mm}, second one {700mm} out of {<u>700mm</u>, 100mm, 700mm, 200mm, 500mm}, and third one {100mm} out of {600mm, 500mm, 400mm, 300mm, <u>100mm</u>}, those values (500mm, 700mm, 100mm) make another solution vector. And if this new vector is better than existing worst vector in the HM, the new vector is included in the HM and the worst vector is excluded from the HM. This procedure is repeated until certain stopping criterion is satisfied.

According to the above algorithm concept, the steps in the procedure of HS for the school bus routing problem are as follows:

Step 1. Initialize the Problem and Algorithm Parameters.

Step 2. Initialize the Harmony Memory (HM).

Step 3. Improvise a new harmony from the HM.

Step 4. Update the HM.

Step 5. Repeat Step 3 and 4 under the stopping criterion.

2.1 Initialize the Parameters

In Step 1, the optimization problem is specified as follows:

$$\text{Minimize } f(\mathbf{x}) \tag{1}$$

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

Where $f(\mathbf{x})$ is an objective function; $\mathbf{x}$ is the set 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 $(x_i(1) < x_i(2) < \ldots < x_i(K))$; N is the number of decision variables (number of music instruments); and K is the number of possible values for the discrete variables (pitch range of each instrument).

The HS algorithm parameters are also specified in this step: Harmony Memory Size (HMS) (= number of solution vectors), Harmony Memory Considering Rate (HMCR), Pitch Adjusting Rate (PAR), and Stopping Criteria (= number of improvisation). Here, HMCR and PAR are the parameters of HS algorithm explained in Step 3.

2.2 Initialize the Harmony Memory

In Step 2, the Harmony Memory (HM) matrix, as shown in Equation 3, is filled with as many randomly generated solution vectors as the size of the HM (i.e., HMS) and sorted by the values of the objective function, $f(\mathbf{x})$.

$$\mathbf{HM} = \begin{bmatrix} \boldsymbol{x}^1 \\ \boldsymbol{x}^2 \\ \ldots \\ \boldsymbol{x}^{HMS} \end{bmatrix} \tag{3}$$

2.3 Improvise a New Harmony

A new harmony vector, $\boldsymbol{x}' = (x'_1, x'_2, \ldots, x'_N)$ is generated by anyone of following three actions: HM consideration; Pitch adjustment; or totally random generation. For instance, the value of the first decision variable (x'_1) for the new vector can be chosen from anyone of values stored in HM ($x_1^1 \sim x_1^{HMS}$). Values of other variables (x'_i) can be chosen in the same manner.

Here, there is a possibility that totally random value can be chosen using the HMCR parameter, which varies between 0 and 1 as follows:

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

The HMCR sets the rate of choosing one value from the historical values stored in the HM, and (1-HMCR) sets the rate of randomly choosing one value from the possible range of values.

Next, each component of the new harmony vector, $\boldsymbol{x}' = (x'_1, x'_2, ..., x'_N)$ is examined whether it should be pitch-adjusted. This procedure uses the PAR parameter that sets the rate of adjustment for the pitch from the HM as follows:

$$\text{Pitch adjusting for } x'_i \leftarrow \begin{cases} \text{Yes} & \text{w.p.} \quad PAR \\ \text{No} & \text{w.p.} \quad (1-PAR) \end{cases} \tag{5}$$

The pitch adjusting process is performed only after a value is chosen from the HM. The value (1-PAR) sets the rate of doing nothing. If the pitch adjustment decision for x'_i is Yes and x'_i is assumed to be $x_i(k)$ (the *k*th element in $\mathbf{X}_i$), the pitch-adjusted value of $x_i(k)$ is

$$x'_i \leftarrow x_i(k+m) \text{ for discrete decision variables} \tag{6}$$

Where m is the neighboring index, $m \in \{..., -2, -1, 1, 2, ...\}$.

The HMCR and PAR parameters introduced in the Harmony Search help the algorithm find globally and locally improved solution, respectively.

2.4 Update the Harmony Memory

If the new harmony vector, $\boldsymbol{x}' = (x'_1, x'_2, ..., x'_N)$ is better than the worst harmony in the HM, judged in terms of the objective function value, the new harmony is included in the HM and the existing worst harmony is excluded from the HM. The HM is then sorted by the objective function value.

2.5 Repeat Step 3 and 4

Step 3 and 4 are repeated until the termination criterion (for example, the number of maximum improvisation) is satisfied.

3. SCHOOL BUS ROUTING PROBLEM

In order to demonstrate the searching ability of the Harmony Search, the HS is applied to a school bus routing problem (SBRP). The SBRP is a practical optimization problem that is closely related to many people's daily lives. From a school's perspective, the SBRP aims to supply students with an efficient and equitable transportation service [15]. Many approaches [15-17] have been developed to solve the SBRP. This problem falls into a large class of problems called vehicle routing problems (VRPs), in which a set of vehicles provides pickup, delivery or simply a service to customers dispersed in an area.

The intractability of the SBRP depends on how the problem is formulated and how many constraints are incorporated. Generally, the wider the problem scope is, the harder the problem becomes, especially to the exact methods such as the branch and bound method (B&B). 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 genetic algorithms (GA) that do not necessarily find the optimal solution, but tend to find good solutions within a reasonable amount of time [18]. The VRPs that the evolutionary or meta-heuristic algorithms had success in applying include transit routing problem and pickup and delivery problem.

Pattnaik et al. [19] proposed an optimization model of minimizing overall cost (both the operator's 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. [20] 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 [21] used GA to solve a multi-vehicle 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. The proposed GA scheme can solve a pickup and delivery problem within an extremely short time compared with the B&B method.

4. PROBLEM FORMULATION

School bus routing in this study is a simple multi-objective problem to minimize 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 Figure 2. Each stop is demanded by certain amount of commuting students, and travel time is specified between two stops.

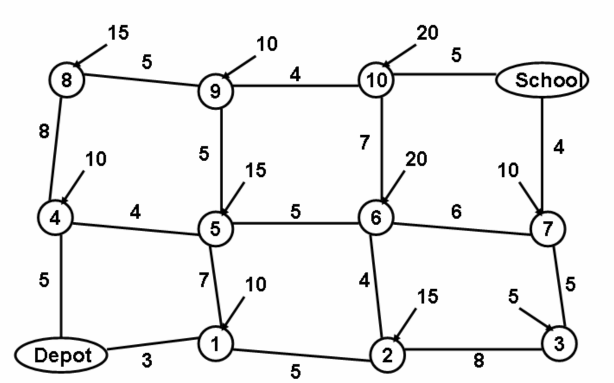

Figure 2. Study Network of School Bus Routing

4.1 Decision Variables

x_i = decision variable having served bus k for demand node i, $i \in DN$, $k \in VS$

$\mathbf{x}$ = vector of decision variables x_i, $i \in DN$

$nbus(\mathbf{x})$ = number of operating buses

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

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

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

4.2 Parameters

fc = fixed cost per school bus

rc = routing cost per moving time

sp_{ij} = shortest path between node i and node j

$pc1$ = penalty cost for bus capacity violation

$pc2$ = penalty cost for time window violation

$nset(VS)$ = number of elements in set VS

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

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

bt = boarding time per student

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

4.3 Sets

DN = 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 = vehicle set

4.4 Formulation

Minimize

$$f(\mathbf{x}) = fc \times nbus(\mathbf{x}) + rc \times \sum_k \sum_{i \in STDN} \sum_{j \in DNED} sp_{ij} lk_{ij}^k + pc1 \times \sum_k vcp^k + pc2 \times \sum_k vtm^k \quad (7)$$

subject to

$$nbus(\mathbf{x}) \le nset(VS) \quad (8)$$

$$\sum_i DM_i^k \le BC^k, \quad k \in VS \quad (9)$$

$$\sum_{i \in STDN} \sum_{j \in DNED} sp_{ij} lk_{ij}^k + \sum_i DM_i^k bt \le TW^k, \quad k \in VS \quad (10)$$

By assigning a discrete decision variable x_i to each bus stop (demand node) i to denote the specific bus k $(k \in VS)$ that serves the stop, the objective function of the problem is to minimize both the number of operating buses and the moving time of the buses as first and second terms in Equation 7. The third and fourth terms represent penalty costs for the violation of bus capacity and time window, respectively. Where the fixed cost per school bus fc is assumed as \$100,000/bus; routing cost per moving time rc is \$105/min; shortest path between node i and node j, sp_{ij} (in minutes), 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 x_i ($i \in STDN$) and x_j ($j \in DNED$) have bus k; penalty cost for bus capacity violation $pc1$ is \$100,000; and penalty cost for time window violation $pc2$ is \$100,000 for the computation.

Equation 8 is the maximum operating bus number constraint. The number of operating buses is less than or equal to the number of candidate buses (= 4 buses). Equation 9 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 10 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 work consists of moving time and boarding time, where each student's barding time bt is 6 seconds.

5. COMPUTATIONAL RESULTS

In order to apply the HS algorithm to the school bus routing problem, parameters of HS algorithm are specified such as the number of musical instruments (= 10 of demand nodes), pitch range of each instrument (= {bus 1, bus 2, bus 3, bus 4}), Harmony Memory Size (HMS) (= 10 ~ 100), Harmony Memory Considering Rate (HMCR) (= 0.3 ~ 0.95), and stopping criteria (= 1000 improvisation). Next, harmonies (solution vectors) are randomly generated from the possible range as many as HMS and sorted by objective function value.

In Step 3, a new harmony is generated from the HM. For instance, the bus of the first demand node in the new vector can be chosen any bus out of the stored buses (for example, {bus3, bus1, bus 2, bus 2, bus 3, bus 1, bus 4, bus 2, bus 4, bus 2}) of the first demand node in HM. The buses of other demand nodes can be chosen in the same manner. On the other hand, in smaller possibility (1-HMCR), a bus can be chosen from all the possible range {bus 1, bus 2, bus 3, bus 4}.

The new harmony $\mathbf{x}'$ is put in the objective function to obtain total cost which consists of fixed bus cost, bus moving cost, and two penalty costs. If bus k violates equation 9, the variable for bus capacity violation, vcp^k , becomes 1, and penalty cost for the capacity violation is added. If bus k violates equation 10, the variable for time window violation, vtm^k , becomes 1, and penalty cost for the time window violation is added.

In Step 4, if the cost of the new harmony is better than the worst cost of any harmony in the HM, the new harmony is included in the HM, and the existing worst cost harmony is excluded from the HM. After that, the HM is sorted.

Finally, in Step 5, the computation is terminated when the stopping criteria is satisfied. If not, Step 3 and 4 are repeated.

The HS is computed for the school bus routing problem and its results are examined by comparing with those of popular evolutionary technique, genetic algorithm (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 (Table 1), the number of function evaluations is 1,000 (= number of improvisation) and the number of runs is 20 with various HMS (10 ~ 100) and HMCR (0.3 ~ 0.95) which parameter values are frequently used in previous HS applications; and in GA (Table 2), the number of evaluation is 1,000 (= population size × number of generations) and the number of runs is 20 with various population size (PS, 10 ~ 100), and mutation rate (MR, 0.01 ~ 0.1), recommended by Koumousis and Georgiou [22].

From the results presented in Table 1 and 2, both algorithms could find the best solution ($307,980) that is the optimal solution of the study network, demonstrated by an exact algorithm. The average routing costs and computing times are $399,870 and 6.6 seconds in HS, and $409,597 and 6.7 seconds in GA on Pentium II 233 MHz.

Table 1. Computational Results of Harmony Search

HMCR / HMS	0.3	0.5	0.7	0.9	0.95
10	410185	410290	407665	410185	410500
20	410185	410185	410185	**307980**	**307980**
40	410185	410395	410185	410185	410185
100	410185	410185	410185	410185	410185

Table 2. Computational Results of Genetic Algorithm

MR / PS	0.01	0.03	0.05	0.07	0.1
10	410395	509240	410185	410185	407350
20	**307980**	410185	406930	410185	410185
40	410290	410185	410185	410185	406930
100	410395	410290	410185	410185	410290

Table 3 shows the route, number of commuting students, and travel time of each bus in optimal solution ($307,980) and near-optimal solution ($410,185). "Do Nothing" in 5th row of the table means that the bus is not operated while satisfying all the constraints.

Table 3. 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

6. CONCLUSIONS

A newly developed algorithm, Harmony Search is modeled for solving the school bus routing problem, and the results of HS computation are compared with those of genetic algorithm. HS mimics musician's behaviors in music improvisation process. Musician's behaviors such as memory considering, pitch adjusting, and random choosing are effectively translated as local and global solution search schemes.

The proposed HS model for the school bus routing problem is to minimize the multi-objective function, consisting of the number of operating buses, the travel time of all buses, and penalty costs related with bus capacity and time window violations. HS could find the optimal solution or near-optimal solutions within reasonable time with advantages including no derivative information requirement, no initial value assumption, no huge memory requirement, and alternative solutions.

From the above-mentioned advantages, the HS algorithm appears to be successfully applied to optimization problems in transportation engineering field. Especially, the presented school bus routing model is expected to be applied to large-scale real networks interfaced with other data-supporting packages such as geographical information system in the future.

7. REFERENCES

[1] Fogel, L. J., Owens, A. J. and Walsh. M. J. *Artificial Intelligence Though Simulated Evolution*. John Wiley, Chichester, UK, 1966.

[2] De Jong, K. *Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Ph.D. Thesis, University of Michigan, Ann Arbor, 1975.

[3] Koza, J. R. *Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems*. Report No. STA-CS-90-1314, Stanford University, 1990.

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

[5] Goldberg, D. E. *Genetic Algorithms in Search Optimization and Machine Learning*. Addison Wesley, MA, 1989.

[6] Glover, F. Heuristic for Integer Programming using Surrogate Constraints. *Decision Sciences*, *8*, 1 (1977), 156-166.

[7] Dorigo, M., Maniezzo, V., and Colorni, A. The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, *26*, 1 (1996), 29-41.

[8] Kirkpatrick, S., Gelatt, C., and Vecchi, M. Optimization by Simulated Annealing. *Science*, *220*, 4598 (1983), 671-680.

[9] Geem, Z. W., Kim, J. H., and Loganathan, G. V. A New Heuristic Optimization Algorithm: Harmony Search. *Simulation*, *76*, 2 (2001), 60-68.

[10] Geem, Z. W., and Tseng, C. –L. New Methodology, Harmony Search and its Robustness. In *Late-Breaking Papers of Genetic and Evolutionary Computation Conference (GECCO '02)* (New York City, USA, July 9-13, 2002), 174-178.

[11] Geem, Z. W., Kim, J. H., and Loganathan. G. V. Harmony Search Optimization: Application to Pipe Network Design. *International Journal of Modelling and Simulation*, *22*, 2 (2002), 125-133.

[12] Kim, J. H., Geem, Z. W., and Kim, E. S. Parameter Estimation of the Nonlinear Muskingum Model using Harmony Search. *Journal of the American Water Resources Association*, *37*, 5 (2001), 1131-1138.

[13] Kang, S. L, and Geem, Z. W. A New Structural Optimization Method Based on the Harmony Search Algorithm. *Computers and Structures*, *82*, 9-10 (2004), 781-798.

[14] Kang, S. L, and Geem, Z. W. A New Structural Optimization Method for Structures with Discrete Sizing Variables. In *Proceedings of 2004 Conference of the Structural Engineering Institute of ASCE (STRUCTURES 2004)* (Nashville, USA, May 22-26, 2004), CD-ROM.

[15] Bowerman, R., Hall, B., and Calamai, P. A Multi-Objective Optimization Approach to Urban School Bus Routing: Formulation and Solution Method. *Transportation Research A*, *29A*, 2 (1995), 107-123.

[16] Bennett, B. T., and Gazis, D. C. School Bus Routing By Computer. *Transportation Research*, *6*, (1972), 317-325.

[17] Tsay, H. –S., and Fricker, J. D. Practical Approach for Solving School Bus Routing. *Transportation Research Record*, TRB, *1202*, (1988), 44-56.

[18] Bodin, L., Golden, B., Assad, A., and Ball, M. The State of Art in the Routing and Scheduling of Vehicle and Crews. *Computers and Operations Research*, *10*, (1983), 63-211.

[19] Pattnaik, S. B., Mohan, S., and Tom, V. M. Urban Bus Transit Route Network Design using Genetic Algorithm. *Journal of Transportation Engineering*, ASCE, *124*, 4 (1998), 368-375.

[20] Chien, S., Yang, Z., and Hou, E. Genetic Algorithm Approach for Transit Route Planning and Design. *Journal of Transportation Engineering*, ASCE, *127*, 3 (2001), 200-207.

[21] Jung, S., and Haghani, A. Genetic Algorithm for a Pickup and Delivery Problem with Time Windows. *Transportation Research Record*, TRB, *1733*, (2000), 1- 7.

[22] Koumousis, V. K., and Georgiou, P. G. Genetic Algorithms in Discrete Optimization of Steel Truss Roofs. *Journal of Computing in Civil Engineering*, ASCE, *8*, 3 (1994), 309-325.