Transit Network Frequencies-Setting Problem Solved Using a New Multi-Objective Global-Best Harmony Search Algorithm and Discrete Event Simulation

Edgar Ruano[1], Carlos Cobos[1(✉)], and Jose Torres-Jimenez[2]

[1] Universidad del Cauca, Popayán, Colombia
{eruano, ccobos}@unicauca.edu.co
[2] CINVESTAV, Ciudad Victoria, Mexico
jtj@cinvestav.mx

Abstract. The rise of Bus Rapid Transit Systems (BRTS) in urban centers involves complex problems of design and scheduling including the scheduling of route intervals across the bus network. The difficulty stems from the fact that transport systems keep to established routes and must set frequencies for each route to minimize costs (measured in terms of transport capacity wasted) and maximize the quality of service (minimizing the total time of users in the system). All this depends on the maximum number of buses available in the system. In an effort to find an alternative solution to the Transit Network Frequencies Setting Problem (TNFSP) on BRTS, this paper proposes using Multi-Objective Global Best Harmony Search (MOGBHS), a multi-objective heuristic algorithm based on three main components: (1) Global-Best Harmony Search, as a heuristic optimization strategy, (2) Non-Dominated Sorting, as a multi-objective optimization strategy, and (3) Discrete Event Simulation, for obtaining quality measures in the solutions found. To test the proposed approach, a simulation model was implemented for Megabus, a BRTS located in Pereira (Colombia), for which the frequency of the buses assigned to routes previously defined in the system was optimized so that operating costs were reduced to a minimum, while user satisfaction was maximized. The MOGBHS algorithm was compared with NSGA-II. It was concluded that MOGBHS outperformed NSGA-II in the number of optimal solutions found (Pareto front points), from 175% in 3,000 fitness function evaluations to 488% in 27,000 evaluations.

Keywords: Bus Rapid Transit Systems · Transit Network Frequencies Setting Problem · Global-Best Harmony Search · Multi-objective optimization · Non-Dominated Sorting · Discrete Event Simulation

1 Introduction

Around the world, Bus Rapid Transit Systems (BRTS) have proven to be a viable alternative solution to growing transportation needs of the population [1]. However, to implement a BRTS several problems must be addressed: the design of the network, the design of routes, the definition of frequencies for each route, the assignment of buses to

O. Pichardo-Lagunas and S. Miranda-Jiménez (Eds.): MICAI 2016, Part II, LNAI 10062, pp. 341–352, 2017.
DOI: 10.1007/978-3-319-62428-0_27

routes, and the assignment of personnel (drivers) to buses and routes. These problems are grouped into a global problem called "Transit Network Design and Scheduling Problem" (TNDSP) [2]. Solving them usually involves conflicting goals within them. For example, if the frequency of the routes is increased to improve quality service, the cost of operating the whole transportation system increases. In addition, each problem has it owns constraints that must be satisfied, such as the maximum availability of buses for the system.

Multi-objective optimization (MO) allows handling problems with multiple conflicting objectives, in order to find a set of non-dominated optimal solutions so that an end user is responsible for selecting the non-dominated solution that meets specific criteria [3]. This paper proposes an MO discrete optimization algorithm to be applied for solving a specific TNDSP problem, the Transport Network Frequencies Setting Problem (TNFSP). This approach of solving TNSFP using a MO discrete optimization algorithm was adopted based on the following observations: (1) major investigations have reported promising results using meta heuristics as a solution strategy, and (2) the hybridization of MO with Harmony Search Algorithm (HS) has shown good results in other areas of application, but this approach has not been used to solve the TNFSP [4, 5].

The proposed optimization algorithm, called MOGBHS has three main components: (a) Global Best Harmony Search (GBHS) [6] as a heuristic optimization strategy (global and local search in the solution space); (b) Non-Dominated Sorting for sorting solutions based on multiple objectives, and (c) Discrete Event Simulation for obtaining quality measures in the solutions found. The GBHS is a hybridization of Harmony Search Algorithm and Particle Swarm Optimization Algorithm, and is responsible for: (a) the generation of new individuals and, (b) the evolution of the individuals in each generation. The second component of MOGBHS carries out the sorting of solutions using the concept of non-dominated solutions to build a Pareto front. The third component of MOGBHS is a discrete event simulation implementation that is responsible for executing simulations in specific scenarios (controlled by the frequency of buses on each route, the specific defined routes and, the pattern of arrivals of the passengers, among others); and collecting the information for bus occupancy and time spent by customers in the system (these measures form the evaluation function to be used by the MOGBHS). In this paper, we have calculated the fitness of the generated solutions by MOGBHS based on the results from the simulation (BRTS) in a specific simulation tool called ARENA [7], in order to provide feedback to the algorithm to facilitate evolution.

MOGBHS was evaluated in a case study based on Megabus, an existing BRTS in Pereira (Risaralda, Colombia), and compared with NSGA-II [8]. In order to make this comparison, the activities involved were: (1) implementation of MOGBHS and NSGA-II; (2) review, selection and analysis of existing BRTS; (3) build a simulation model based on a simplification of selected BRTS; (4) test of MOGBHS and NSGA-II supported by the implemented simulation model; and (5) consolidation and analysis of results. On completing the test activities it was found that MOGBHS greatly improves the efficiency compared with the solutions generated by NSGA-II, from 175.12% in 3,000 fitness functions evaluations (FFE) to 488.68% in 27,000 FFEs.

The remainder of this paper comprises four more sections. In Sect. 2 a state of the art review about TNFSP solutions is presented. In Sect. 3 the multi-objective heuristic algorithm (MOGBHS) proposed is presented. Section 4 presents the results of the MOBGHS contrasted with the results of NSGA-II (a fast and elitist multi-objective genetic algorithm) for a specific case real-based used to make the comparison. Finally, Sect. 5 presents the conclusions of the work and some future activities the authors plan to undertake.

2 State of the Art

In the state of the art, many meta-heuristic approaches for solving the TNDSP are reported [4, 9–12]. Many of them have in common that they solve the problem one objective at a time and/or subdivide the problem in sub-problems and then solve each sub-problem separately. Examples of these meta heuristics are: (a) GRASP-TNDP that bases its operation on GRASP with the difference that instead of generating a single solution, it seeks a Pareto approximation [13]; and (b) the set of heuristics proposed in [14] that includes routines for generating routes and a genetic algorithm to find an optimal set of routes with corresponding frequencies for the implementation of a system of ZEV (Zero Emission Vehicles) public transport vehicles. Given that these algorithms do not use a multi-objective approach to solve the problem, the obtained solutions are in general not so useful (given that a predefined priority to solve each objective is implied). More recent works include: a genetic algorithm with elitism for transit network design [15], a bi-level modeling for the transit route and frequency design and a hybrid artificial bee colony algorithm in order to solve the entire problem [16], and a multi-objective approach with objectives alternation for transit network design and frequencies setting [17], among others.

In order to solve a multi objective problem, with two or more objectives generally in conflict with each other, an evolutionary approach was originally proposed. The algorithms that follow this approach are called Multi-Objective Evolutionary Algorithms (MOEA). Reported MOEA implementations have used three basic approaches to deal with the objectives: (a) Naive; (b) Non-Dominated Sorting; and (c) Pareto Strength [3]. The strategy of a non-dominated system basically consists in taking the solutions that are part of the Pareto front. The solutions of this front will be considered the best and therefore belong to the result of the optimization process. One of the best algorithms based on this strategy is the Non-dominated Sorting Genetic Algorithm II (NSGA-II), which to ensure good dispersion of the solutions on the Pareto front uses the Crowding Distance measure [8].

The non-dominated and strength Pareto based approaches deals with many objectives and have proven to be more successful in solving the TNDP [14, 18, 19]. A good review of the use of simulation models and multi-objective optimization to solve the TNDP can be found in [20]. The main differences between the proposal by Wang and the proposed approach of this paper (MOBGHS) are: (a) the problem to be solved is the TNFSP; (b) the objectives in conflict to be optimized are: minimize both the average time spent by passengers in the system and the average wasted bus capacity; (c) the algorithm used to solve the problem is based on Global-Best Harmony

Search (moreover as far as we know this is the first time that this algorithm has been applied to solve the TNFSP problem as a multi-objective problem); and (d) the model for carrying out the simulation is totally different: discrete event simulation.

The Harmony Search (HS) optimization algorithm was originally proposed in [21]. It is based on jazz musicians' improvisation process, performed in search of a perfect harmony, in short: HS randomly generates, evaluates and sorts a population in a place called Harmony Memory (HM). Then, for a determined number of improvisations it generates one element at a time where each variable of that harmony may be completely random or taken from the HM (probability defined by Harmony Memory Consideration Rate parameter, HMCR), and in case of it being taken from HM, may or may not (depending on the parameter Pitch Adjustment Rate, PAR) be altered by addition or subtraction of an arbitrary value called Bandwidth (BW) also defined in the configuration. Later some variants/improvements for the HS algorithm were proposed in: (a) improved harmony search (IHS) where the accuracy and convergence rate of HS was improved, arranging for PAR and BW parameters to depend on the current iteration number [22]; (b) global-best harmony search (GBHS) that introduces swarm intelligence, changing the pitch-adjustment by extraction of values from the best harmony in HM [6]; (c) others such as global-best harmony search using learnable evolution models (GHS + LEM) [23], improved global-best harmony search [24], parameter adaptive harmony search [25] and, global dynamic harmony search (GDHS) [26].

3 Details of the MOGBHS Applied to Solve the TNFSP

In an attempt to improve user satisfaction and reduce operating costs in BRTS, a multi-objective algorithm that searches configurations of bus output frequencies is proposed. The objectives are: (1) Minimize the average time spent by passengers on the system, and (2) Minimize the average wasted bus capacity. It additionally included the constraint of the maximum available fleet (maximum amount of buses) for the BRTS.

Given that; (1) HS, IHS and GBHS have in common that they have a small number of parameters to tune the performance of the algorithm (7 parameters for IHS and 5 for the other two); (2) HS, IHS and GHS have a fast convergence, demand a modest amount of computer resources, and have a low probability of getting trapped in a local optima; (3) GBHS improves IHS results (which in turn improves the accuracy and convergence rate of HS); (4) GBHS facilitates the parameter tuning; and (5) GBHS works efficiently in continuous and discrete problems; in this research it was decided to use GBHS as the core of the proposed MO algorithm.

GBHS was originally designed to work with a unique objective. Therefore, in this work it was adapted to work with more than one objective using non-dominated sorting, based on the concepts of ordering by Pareto front and Crowding distance [3, 8]. The proposed algorithm, called Multi-Objective Global Best Harmony Search (MOGBHS) generates a set of harmonies and stores them in harmony memory, evaluates all the objectives for each element in HM, and then sorts by Pareto front and Crowding Distance. Then improvisations (evolutionary iterations) are carried out, in each element of which a new harmony is generated by applying the logic of the GBHS

algorithm. The New Harmony is evaluated, added (or inserted) to the existing HM, the HM then is sorted by Pareto fronts and Crowding distance, and the element that make the population exceed the maximum harmony memory size (HMS) is eliminated (the worst harmony in HM). Figure 1 shows the MOGBHS pseudo-code.

The algorithms require the following parameters

- NI: number of improvisations performed by the algorithm
- PARmin, PARmax: Pitch Adjusting Rate minimum and maximum respectively, used to calculate the PAR(γ) value for each iteration (see line 9 in Fig. 1)
- HMCR: Harmony Memory Consideration Rate, used to decide whether a variable from a new improvise is taken from harmony memory or randomly generated
- HMS: Harmony Memory Size defines the maximum length of solution vector that represents the harmony memory
- NR: number of routes/variables in the target system
- NO: number of objectives to evaluate and optimize
- Limits: integer array of size [NR, 2], where for each variable (that represents a route in the system), the maximum and minimum are defined for the value of the output interval for this route
- MNB: Maximum Number of Buses into the system, used to calculate the feasibility of a solution.

```
01  HM.PopulationRandomInitialize(Limits, MNB)
02  HM.NonDominatedOrderCalculate()
03  HM.CrowdingDistanceCalculate()
04  HM.Sort() /*Based on Pareto Front and Crowding Distance if solutions are in the same Pareto front */
05  for i = 1 to NI do
06          ActualHarmony = new Harmony() /* Create a new empty solution (improvisation or harmony) */
07          for j = 1 to NR do
08                  if Random(0,1) < HMCR then
09                          ActualHarmony.Intervals[j] = HM.Harmonies[Random(0,HMS)].Intervals[j]
10                          PAR = PARMin + (((PARMax - PARMin) *i) / NI)
11                          if Random(0,1) < PAR then
12                                  ActualHarmony.Intervals[j] = HM.FromBestHarmony(Random(0,NR))
13                          end if
14                  else
15                          ActualHarmony.Intervals[j] = HM.RandomSelection(j, Limits)
16                  end if
17          end for
18          if HM.InPopulation(ActualHarmony) == false and ActualHarmony.Viable(MNB) == true then
19                  HM.Evaluate() /* Set fitness values using results of ARENA simulation */
20                  HM.Add(ActualHarmony)
21                  HM.NonDominatedOrderCalculate()
22                  HM.CrowdingDistanceCalculate()
23                  HM.Sort()
24                  HM.RemoveTheWorst()
25          end if
26  end for
```

Fig. 1. Multi-Objective Global Best Harmony Search (MOGBHS).

Each harmony (solution) generated by MOGBHS includes:

- A vector of integers with NR positions called Intervals, where Intervals[α] $\forall\ \alpha \in$ [1, NR] is the interval of time between the outputs of the buses that serve the route α
- A vector of doubles with NO positions called Evaluations, where Evaluations[β] $\forall\ \beta \in$ [1, NO] is the fitness value of the current solution (harmony) for the objective β
- An integer variable called Front that stores the number of Pareto front points for this harmony compared to the members of the population at any given time
- A float variable called CrowdingDist that stores the crowding distance of the harmonies to other elements at the same Pareto front.

The *PopulationRandomInitialize* procedure is responsible for generating an initial population of size HMS, considering the restrictions defined in Maximum Number of Buses into the system (MNB) and Limits parameters. The *NonDominatedOrderCalculate* procedure set the Pareto front number for each element in the harmony memory. This procedure is adapted from [8]. The *CrowdingDistanceCalculate* procedure takes the existing population and provides for each element the crowding distance to harmonies at the same Pareto front. Finally, the *Sort* function, as the name implies, sorts the population according to the following criteria: (1) in ascending order based on the Pareto front number if the harmonies (solutions) have a different front number and, (2) in descending order based on crowding distance when the harmonies have the same Pareto front number.

In addition to the above functions the algorithm uses the following auxiliary functions:

- *InPopulation*: check if a harmony exists in the HM
- *IsViable*: check if a harmony is a feasible solution based on the size of the available fleet (MNB parameter). Returns true when the New Harmony can be implemented with this fleet size
- *Add*: This procedure adds a new item to the end of a specific list
- *Length*: Returns the length of a list
- *RemoveTheWorst*: removes the worst harmony from harmony memory (HM) based on Pareto front number and crowding distance.

Given the need to evaluate the fitness of each harmony generated by the algorithm against the selected objectives and considering some successful cases of algorithms supported by simulations, it was proposed to use discrete event simulation for this purpose. Therefore before the execution of the algorithm, a model of discrete event simulation was created to measure the time spent by passengers and wasted bus capacity in the system. Later, during the execution of MOGBHS, the *Evaluate* function uses the intervals values of the current harmony into the simulation model, executes the simulation, reads the assessment for each of the objectives and loads them in the Evaluations vector of the harmony.

4 Case Study

To test the solution approach to TNFSP on BRTS the following activities were performed:

- Implementation of the MOGBHS algorithm
- Review, Selection and Analysis of BRTS to model
- Implementation of a simplified version of the selected BRTS using the proposed meta-model as a concept test
- Testing of the algorithm with the simulation model.

1. **Implementation and Test Environment:** MOGBHS was implemented in C# programming language, chosen for its ability to manage files and character strings for information extraction from the output of the simulations, speed of language (being compiled) and experience of the research group (over 7 years).

The experiments were run on an ASUS N56VZ computer with an Intel Core i7-3630QM Processor at 2.4 GHz and 8 GB of RAM. The following software was used: Windows 8 (64 bits), Microsoft Visual Studio 2010 (.Net Framework 3.5), and Arena 14.0.

2. **Review, Selection and Analysis of BRTS to Use:** In selecting a BRTS to model for MOGBHS testing, the whole range of such systems in Colombia was reviewed, including Transmilenio (Bogota D.C), Mio (Cali, Valle), Metrolinea (Bucaramanga, Santander) and Megabus (Pereira, Risaralda). Due to the amount of documentation available, the size and configuration, and simplicity of routes it was decided to take Megabus as a basis for BRTS.

The Megabus system has 37 stations and 3 routes; however it was subsequently decided to create a simplified version that had the same number of routes with the same design and layout, but with fewer stations. This is because the aim is to look for a proof of concept of the solution strategy proposed in this investigation. The real test of the system will be made in a second phase, mainly involving additional time to implement the BRTS in ARENA and the achievement of real system parameters (for example, the real average arrival time of the passengers, their distribution and destinations).

To simplify the model, stations with similar characteristics were grouped and only one station was selected for each group, causing the total number of stations to be modeled to be reduced to eleven (11). There was a similar reduction from two to a single Bus Central. Figure 2 presents the simplified model, in which the green color indicates the path traveled by vehicles with route number one (1), the red color route number two (2) and the blue color for the buses that make up route number three (3).

3. **Implementation of Simulation Model:** ARENA [7] was the modeling and implementation of discrete event simulation software selected to create the simulation model that MOGBHS requires for evaluation of the objectives to be minimized. The operation of ARENA is based on SIMAN, a general purpose simulation language that facilitates integration with programming languages with the ability to

manage files. Using the simplified BRTS, the research group proceeded with the generation of the simulation model. A screenshot of the BRTS simulation model is presented in Fig. 3. Once the model is completed, and using ARENA, SIMAN language source files were generated, these files serve as a resource for the MOGBHS algorithm implementation that generates, evaluates and improves specific harmonies using results of simulations associated with each harmony.

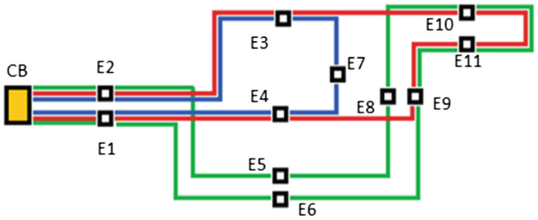

Fig. 2. Megabus simplified route design (Color figure online)

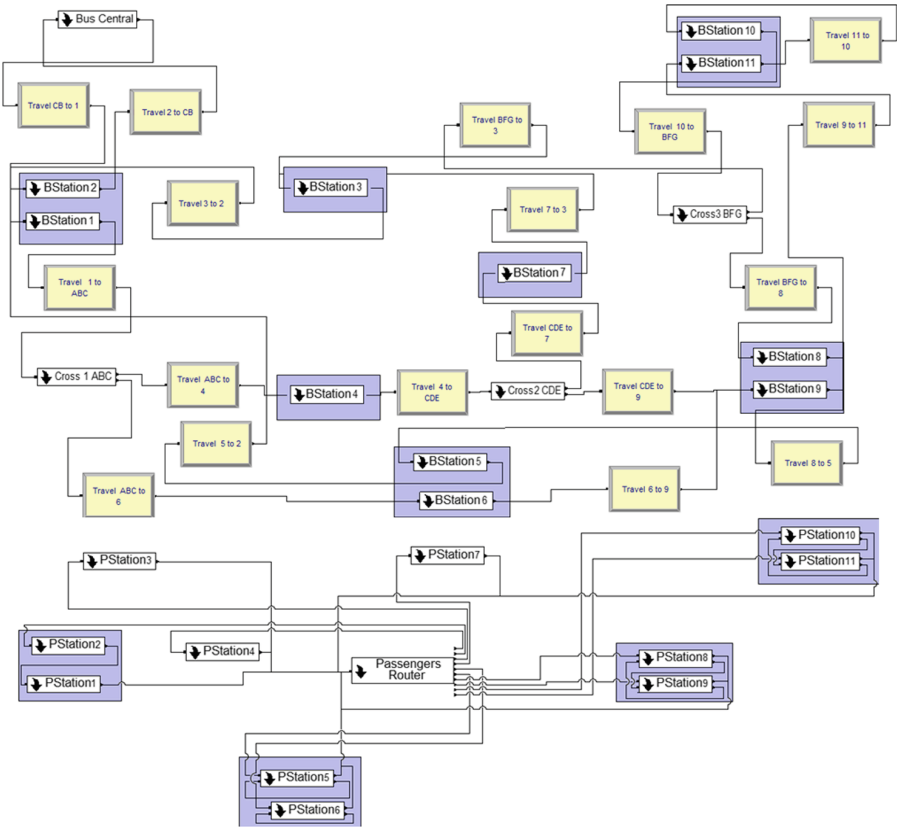

Fig. 3. BRTS simulation model

4. **Experimentation:** An NSGA-II algorithm implementation was also used in order to compare the results of MOGBHS algorithm. The results were obtained from the average of 30 runs of each algorithm (MOGBHS and NSGA-II), using its own setting and with different numbers of fitness function evaluations (FFE).

The configuration parameters for MOGBHS algorithm were: Number of Variables: 3, corresponds to the number of intervals to configure the three routes for operating the BRT system, equivalent to the number of routes in "real" system; Harmony Memory Size (HMS): 100; Harmony Consideration Rate (HMCR): 0.7; Pitch Adjustment Rate (PAR) minimum: 0.1; Pitch Adjustment Rate (PAR) maximum: 0.9; It was decided that a reasonable domain for intervals for output routes of buses would be between 1 min and 30 min, including limits; and Improvisation Number: 3,000 to 27,000.

Values for PAR and HMCR were set based on the values recommend in [6, 22]. The defined value for HMS is higher than recommended in state of the art because is unviable to obtain a good Pareto front with small values of HMS. The maximum improvisation number was calculated with the number of possible configurations for a single route (30) and with the number of considered routes (3), resulting in 27,000 possible configurations.

Given that in the state of art NSGA-II [3, 8] is reported as one of the best algorithms in the field of multi-objective optimization, it was decided to include it as a baseline and a point of comparison against the results obtained by MOGBHS. NSGA-II involves selecting parents by binary tournament, simulated binary crossover (SBX) and polynomial mutation. Since NSGA-II was created to solve continuous problems, it was necessary to adjust the simulation and exploration operations to limit results to valid solutions in the space of discrete values. The parameters for the test were based on those recommended in [8], namely: Probability of mutation: 1/3; Probability of crossing: 0.8; Population size: 100; Number of variables: 3; Allowed output intervals: 1 to 30 min; and Number of evaluations (fitness function evaluations): 3,000 to 27,000.

To obtain all the best solutions in order to be able to calculate the effectiveness of both algorithms, and considering relatively few variables and few possibilities, an exhaustive search was run. Given the number of options per route (30), and the number of routes in BRTS considered (3), the number of options amounted to 27,000. After implementing the exhaustive search, 99 solutions were found for the first Pareto front, considered as the best existing solutions.

In Fig. 4, the effectiveness of MOGBHS and NSGA-II are compared (optimal solutions number found divided by total number of optimal solutions in the Pareto front), every 3,000 FFE over the 30 tests in each algorithm. Similarly, Table 1 shows a comparison of the average effectiveness of the two algorithms (MOGBHS and NSGA-II), taking measurements every 3,000 evaluations of the objective functions.

MOGBHS found more optimal solutions than NSGA-II after 3,000 evaluations (FFEs), that is, it is more effective. The effectiveness is improved from 175.12% in 3,000 FFEs up to 488.68% in 27,000 FFEs. A Wilcoxon signed rank test shows with 95% confidence that MOGBHS results outperformed the NSGA-II results.

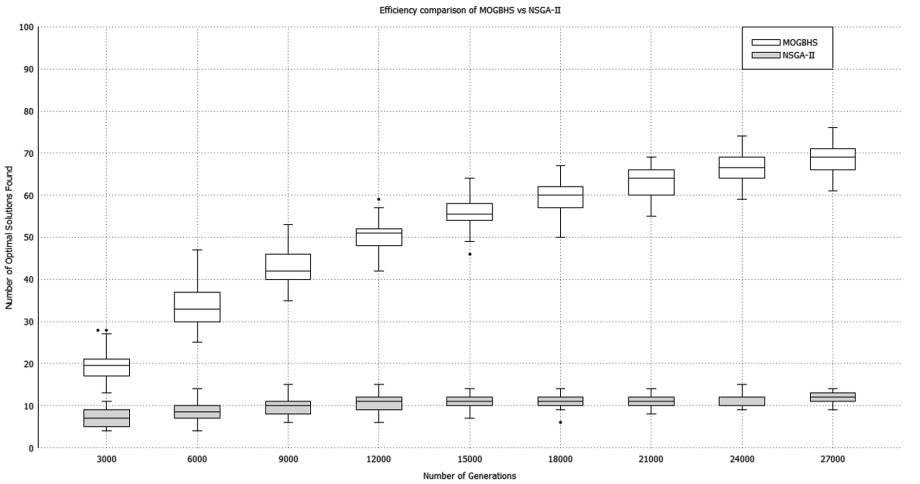

Fig. 4. Graphic effectiveness comparison of MOGBHS vs NSGA-II

Table 1. Effectiveness comparison of MOGBHS vs NSGA-II

	MOGBHS		NSGA-II		
FFE	Optimal solutions found (AVG)	Effectiveness (%)	Optimal solutions found (AVG)	Effectiveness (%)	Effectiveness improve (%)
3000	19.53	19.73	7.10	7.17	175.12
6000	33.93	34.28	8.73	8.82	288.55
9000	42.73	43.16	9.87	9.97	333.11
12000	50.40	50.91	10.80	10.91	366.67
15000	55.40	55.96	10.97	11.08	405.17
18000	59.60	60.20	10.80	10.91	451.85
21000	63.47	64.11	11.13	11.25	470.06
24000	66.30	66.97	11.60	11.72	471.55
27000	68.48	69.17	11.63	11.75	488.68

5 Conclusions and Future Work

The proposed algorithm (MOGBHS) is based on (a) Global-Best Harmony Search as a heuristic optimization strategy, (b) Non-Dominated Sorting as a multi-objective optimization strategy, and (c) Discrete Event Simulation (based on ARENA and SIMAN) for obtaining quality measures in the solutions (harmonies) found.

The MOGBHS was compared against an NSGA-II implementation using a real test case from the city of Pereira, and the comparison indicated that MOGBHS effectiveness is superior to NSGA-II, from 175.12% (in 3,000 FFEs) to 488.68% (in 27,000 FFEs).

MOGBHS can be used to solve various problems of multi-objective optimization. In this respect, the research team hopes to make a comprehensive evaluation of the discrete multi-objective benchmark problems and compare the performance of the algorithm against others in the state of the art, including NSGA-II and SPEA-2 [27].

Given that the evaluation of an individual requires to do a simulation (this is the most time-consuming part of the solution process) it appears promising to speed up the convergence, remembering some previous solutions to avoid resampling them (approach similar to the one followed in Tabu Search [28, 29]). Also, given the large number of possible combinations of the parameters of the algorithm, we plan to use Covering Arrays [30] and meta-algorithms in order to identify the best parameter settings for the algorithm in more complex scenarios.

References

1. Cervero, R., Bus Rapid Transit (BRT): An efficient and competitive mode of public transport. IURD Working Paper 2013–01 (2013)
2. Farahani, R.Z., et al.: A review of Urban transportation network design problems. Eur. J. Oper. Res. **229**(2), 281–302 (2013)
3. Luke, S.: Essentials of Metahuristics. Lulu, Raleigh (2010)
4. Mazloumi, E., et al.: Efficient transit schedule design of timing points: a comparison of ant colony and genetic algorithms. Transp. Res. Part B: Methodol. **46**(1), 217–234 (2012)
5. Sivasubramani, S., Swarup, K.: Environmental/economic dispatch using multi-objective harmony search algorithm. Electr. Power Syst. Res. **81**(9), 1778–1785 (2011)
6. Omran, M.G., Mahdavi, M.: Global-best harmony search. Appl. Math. Comput. **198**(2), 643–656 (2008)
7. Automation, R.: Arena simulation software, vol. 24, Accessed Nov 2013
8. Deb, K., et al.: A fast and elitist multiobjective genetic algorithm: NSGA-II. Evolutionary Computation, IEEE Trans. **6**(2), 17 (2002)
9. Ibarra-Rojas, O.J., et al.: Planning, operation, and control of bus transport systems: a literature review. Transp. Rese. Part B: Methodol. **77**, 38–75 (2015)
10. Kechagiopoulos, P.N., Beligiannis, G.N.: Solving the Urban transit routing problem using a particle swarm optimization based algorithm. Appl. Soft Comput. **21**, 654–676 (2014)
11. Nikolić, M., Teodorović, D.: Transit network design by bee colony optimization. Expert Syst. Appl. **40**(15), 5945–5955 (2013)
12. Yu, B., et al.: Transit route network design-maximizing direct and transfer demand density. Transp. Res. Part C: Emerg. Technol. **22**, 58–75 (2012)
13. Mauttone, A., Urquhart, M.E.: A route set construction algorithm for the transit network design problem. Comput. Oper. Res. **36**(8), 2440–2449 (2009)
14. Beltran, B., et al.: Transit network design with allocation of green vehicles: a genetic algorithm approach. Transp. Res. Part C: Emerg. Technol. **17**(5), 475–483 (2009)
15. Nayeem, M.A., Rahman, M.K., Rahman, M.S.: Transit network design by genetic algorithm with elitism. Transp. Res. Part C: Emerg. Technol. **46**, 30–45 (2014)
16. Szeto, W.Y., Jiang, Y.: Transit route and frequency design: bi-level modeling and hybrid artificial bee colony algorithm approach. Transp. Res. Part B: Methodol. **67**, 235–263 (2014)
17. Arbex, R.O., da Cunha, C.B.: Efficient transit network design and frequencies setting multi-objective optimization by alternating objective genetic algorithm. Transp. Res. Part B: Methodol. **81**, 355–376 (2015)

18. Cipriani, E., Gori, S., Petrelli, M.: Transit network design: a procedure and an application to a large Urban area. Transp. Res. Part C: Emerg. Technol. **20**(1), 3–14 (2012)
19. Mauttone, A., Urquhart, M.: A multi-objective metaheuristic approach for the transit network design problem. Publ. Transport **1**(4), 253–273 (2009)
20. Wang, J., Sun, G., Hu, X.: Optimization of transit operation strategies: a case study of Guangzhou, China Annual Meeting of the Transportation Research Board (2013)
21. Lee, K.S., Geem, Z.W.: A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput. Methods Appl. Mech. Eng. **194** (36–38), 3902–3933 (2005)
22. Mahdavi, M., Fesanghary, M., Damangir, E.: An improved harmony search algorithm for solving optimization problems. Appl. Math. Comput. **188**(2), 1567–1579 (2007)
23. Cobos, C., Estupiñán, D., Pérez, J.: GHS + LEM: Global-best Harmony Search using learnable evolution models. Appl. Math. Comput. **218**(6), 2558–2578 (2011)
24. El-Abd, M.: An improved global-best harmony search algorithm. Appl. Math. Comput. **222**, 94–106 (2013)
25. Kumar, V., Chhabra, J.K., Kumar, D.: Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems. J. Comput. Sci. **5**(2), 144–155 (2013)
26. Khalili, M., et al.: Global dynamic harmony search algorithm: GDHS. Appl. Math. Comput. **228**, 195–219 (2014)
27. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength pareto evolutionary algorithm. TIK-report, vol. 103 (2001)
28. Glover, F.: Tabu search—Part I. ORSA J. Comput. **1**(3), 190–206 (1989)
29. Glover, F.: Tabu search—Part II. ORSA J. Comput. **2**(1), 4–32 (1990)
30. Torres-Jimenez, J., Izquierdo-Marquez, I.: Survey of covering arrays. In: 2013 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). IEEE (2013)