Wireless Access Point Configuration by Genetic Programming

Jianjun Hu
Genetic Algorithm Research and Application Group
Dept. of Computer Science and Engineering
Michigan State University
East Lansing, MI 48823
Email: hujianju@cse.msu.edu

Erik Goodman
Genetic Algorithm Research and Application Group
Dept. of Electrical and Computer Engineering
Michigan State University
East Lansing, MI 48823
Email: goodman@egr.msu.edu

Abstract- **Wireless access point configuration problem in wireless LAN deployment can be formulated as a non-linear optimization problem with variable number of parameters. In this paper, a strongly-typed genetic programming is applied to solve an abstract version of this problem successfully. It is argued that this problem can be used as a potential benchmark problem for evaluating techniques and investigating issues in strongly typed genetic programming, topologically open-ended synthesis by genetic programming, and simultaneous topological and parametric search.**

I. INTRODUCTION

Wireless LAN systems are widely used today for their low cost and convenience. However, deployment of such systems to ensure quality of service at minimal cost is not a trivial problem. The basic task is to decide a configuration of access points in a service area whose parameter set includes the number of access points (AP) and the location and radio parameters for each AP to satisfy system requirements [1].

As a hard non-linear optimization problem, a variety of approaches have been applied to wireless LAN access point configuration including simplex method [2], genetic algorithm [3, 9, 10], simulated annealing [4, 5], and Tabu search [6]. Genetic algorithms have been criticized as inappropriate for this problem as their performance depends on their many running parameters and there is no convergence guarantee [7]. Instead, some deterministic approaches [7,1] are preferred for simplicity even though there is no guarantee of optimality. However, in many cases the population based evolutionary algorithms may be more desirable if the cost saving from the global optimality is important especially as computing power is cheap today. Another justification to use evolutionary algorithms is that there are many tradeoffs among multiple objectives of the system configuration and evolutionary algorithms have proved to be one of the most effective approaches for identifying a set of Pareto solutions simultaneously [8].

Application of genetic algorithms to the wireless access point configuration problems have been tried by several researchers. Martin et al. [9] applied a preferential ranking based genetic algorithm to transceiver location of indoor wireless communication networks. Their algorithm uses a fixed length representation to encode the positions of all transceivers, in which the number of transceivers has to be estimated in advance. A similar approach working on fixed number of base stations is also reported in [10]. Tang et al. [11] proposed an innovative variable length genetic algorithm named hierarchical GA (HGA) to solve handle the configuration problem with variable number of access points. In this method, firstly, the number (K) of access points that could provide sufficient coverage of the service to all clients is estimated. Then the genotype not only encodes K pairs of position parameters but also an additional binary string of length K, each bit controlling whether the corresponding access point is activated or not in the current solutions. This will allow the algorithm to identify the minimal number of access points. The disadvantage here is that one still has to estimate the value K that is not obvious for complex problems. If the K is too big, the search space is unnecessarily big. If K is too small, one has the risk of missing the optimal solution. However, this work fully demonstrates that the multi-objective evolutionary algorithms is very effective to explore the wireless access point configuration problem and can handle complex constraints.

As the wireless access point configuration problem can be formulated as a variable length search problem, it is natural to apply genetic programming to it. Since genetic programming searches the problem space starting from small solutions, the difficulty of estimating K in HGA [11] is largely avoided. This paper thus aims to demonstrate that genetic programming with its capability of simultaneous search of both topology and parameters can be used for the wireless access point configuration problem. Another more important motivation is to define a simplified version of this problem and introduce it to the genetic programming research community as an additional benchmark problem, suitable for testing strongly typed genetic programming and simultaneous topology and parameter search typical in evolutionary synthesis by genetic programming. Currently, there is no such a benchmark problem with the desirable properties such as scalable size and easy implementation.

The rest of the paper is organized as follows. A continuous version of wireless access point problem is defined in Section II and related genetic algorithms approaches to handle the variable length issue are discussed. The genetic programming approach for wireless AP configuration problem is then introduced in Section III, which is followed by the experimental results in Section IV. A discussion of the appropriateness of wireless AP configuration problem as a benchmark for strongly typed genetic programming and topologically open-ended synthesis is given in Section V. Finally, the conclusions are presented.

II. DESCRIPTION OF THE PROBLEM

The real-world wireless access point configuration problem can be formulated in many different ways. For example, [7] assumes that all APs are only located at a specified set of possible locations. In addition to the position information, parameters of an AP can include radio parameters related to transmission power, channel allocation, and antenna direction. The fitness evaluation of a configuration can be very complicated if time-consuming simulation by ray tracing is used to allow accurate estimation. In this paper, a different formulation is considered in which only the number of APs and the position information of each AP need to be evolved. This formulation keeps the essentials of the wireless configuration problem but enjoys much faster fitness evaluations and thus is suitable to be used as a benchmark problem.

The scenario of the wireless access point configuration problem is as follows: a community is planning to provide wireless Internet service to its citizens who are located around in an area of size W ×H. The citizens (clients) are labeled as circled C in Figure 1. A certain number of access points need to be placed at several places to cover all clients because each access point has a limited service radius. All access points are wired together and connected to a Internet gateway G. To reduce the cost, a design solution with minimal cost of devices of access points and minimum cost of the wire connecting those access points is considered as optimal. The design problem is to determine an optimal number of access points and the positions of each AP. We assume that APs can be put at any place (integer coordinates are used in the experiments in this paper). To make the problem harder, we can also choose to maximize the average signal strength at all clients, or add some constraints on the location of the access points.

More precisely, our problem is defined as follows:

Given a set of N clients located at

$(x_i^c, y_i^c), i = 1...N$ in an area of size $W \times H$,

where $x_i^c \in [0, W]$ *and* $y_i^c \in [0, H]$; the location of the gateway S at (x^g, y^g), and the service radius of an AP as r_s, find a configuration of a set of access points, each with a cost of C_{AP}, at locations (x_i^{AP}, y_i^{AP}), $i = 1...N_{AP}$ wired together and connected to the gateway port S such that each client is covered by at least one access point and the total cost of access points and wire is minimal, where the cost of a unit length wire is C_w:

$$\text{minimize} \quad f = C_{AP}.N_{AP} + C_w * \sum_{for\ all\ wire\ segment\ L_i} |L_i|$$

where L_i is the connection wire segments among access points

As the above problem requires simultaneous search of both the connection topology among APs and location parameters of all APs, genetic programming is a suitable approach for this problem.

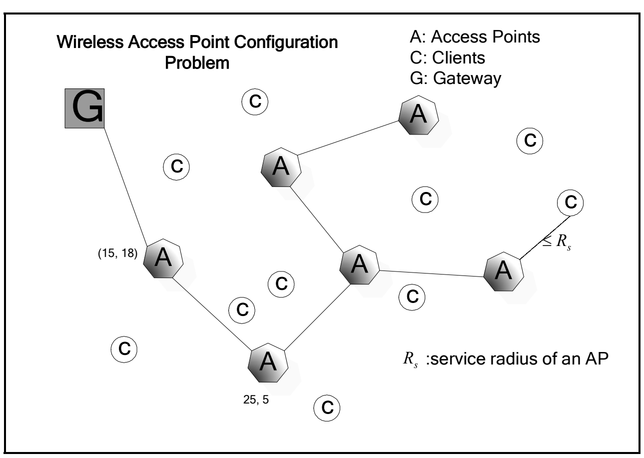

Figure1. The wireless access point configuration problem

III. THE GENETIC PROGRAMMING APPROACH

The wireless access point configuration problem described above can be solved by evolving a connected graph to minimize the device cost and wire cost. A strongly typed genetic programming approach for evolving graphs –cellular encoding [12] has been widely used in analog circuit synthesis [17], bond graphs [13], graph design[20], and etc. This approach has two components:

(1) The embryo: this is the starting graph upon which the operations in the evolved genetic programming trees will work on to grow the graph. In our case, the embryo is simply the starting Internet gateway position node G.

(2) Function and terminal set: there are two types of functions and terminals. One type is the set of topological operators (Figure 2) used to add, remove, connect or disconnect the APs (or nodes in general). The other is the set of numeric operators for establishing the parameters of the access point. All the GP functions and terminals are described in the Table I.

Table I. GP FUNCTIONS AND TERMINALS

Topological operators	
f_InsertV	insert a node into a given edge.
f_AttachV	attach a new node to current node
f_EndV	terminate growth at a node modifiable site
f_EndE	terminate growth at an edge modifiable site
f_CutV	remove a node from the graph and all related edges
f_CutE	remove an edge from the graph
Numerical operators	
f_add	add two numbers
f_sub	subtract two numbers
f_mul	multiply two numbers
f_div	divide two numbers

(3) The fitness function: we define our fitness function as follows. The total fitness is composed of the weighted sum of three fitness criteria that evaluate different aspects of an access point configuration.

$$\text{coverFitness} = \frac{4.0 * nCoveredClients}{nCoveredClients + nClients}$$

where *nColveredClients* is the number of clients covered by the solution, *nClients* is the number of all clients to be covered. This fitness criteria evaluates what well the clients

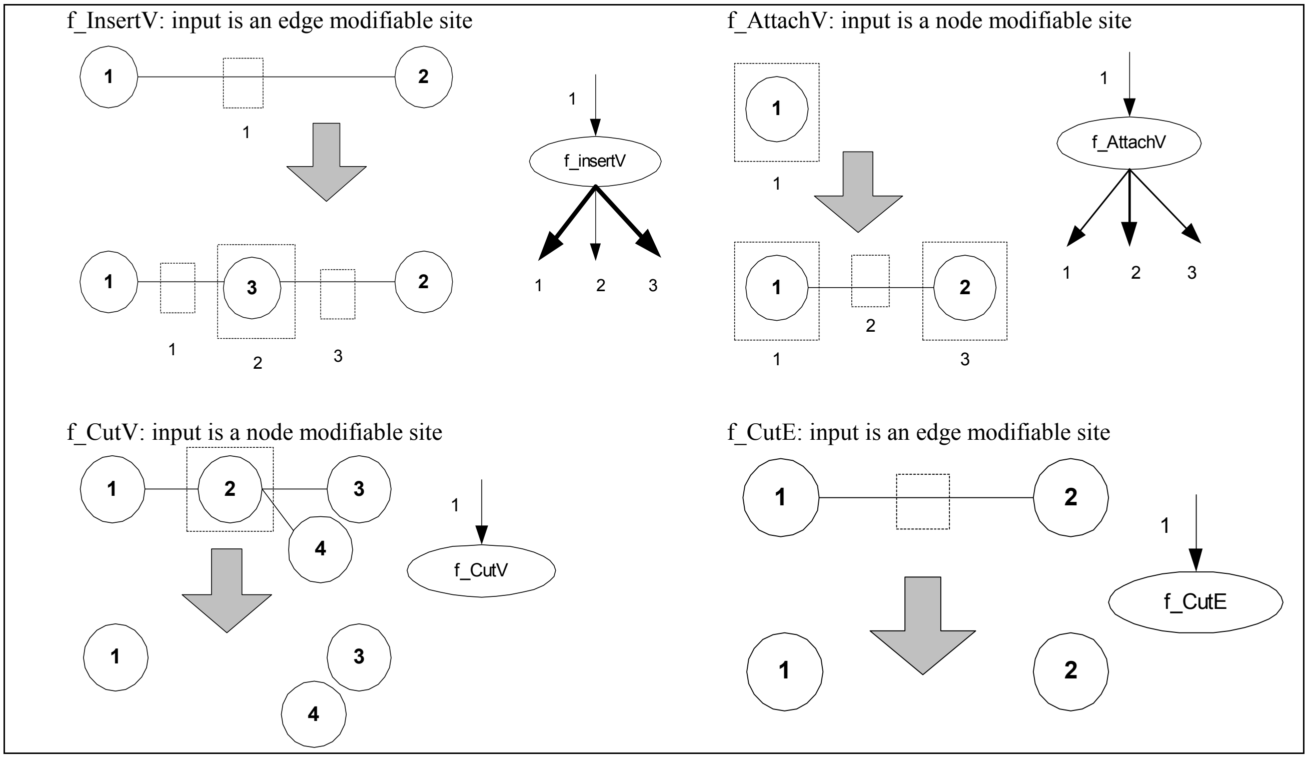

Figure 2. Topological operators in GP function and terminal set.

are covered by current configuration. The more clients covered, the better.

$$\text{wirecostFit} = \frac{10000}{10000 + wireLength}$$

where *wireLength* is the total length of all wire segments that connect access points. This fitness criteria evaluates the goodness as regard to the cost of wires used to connect the access points. The shorter the wires used, the better.

$$\text{access\_point\_costFit} = \frac{nClients}{nClients + 1.5 * N_{AP}}$$

where nClients is the number of clients to be serviced and N_{AP} is the number of access points. This fitness criteria calculates the goodness as regard to the cost of wireless access points that are needed. The fewer access points used, the better.

totalfitness= 0.7*coverFit+0.1* wirecostFit +

0.2* access_point_costFit

We also penalize the fitness of a solution by half if it contains isolated access points.

The set of topological operators in Figure 2 are able to construct any tree like graph, which covers our optimal solutions. It is still the case even we remove the last two operators. Note that these topological operators are used only for topology search and thus may not have any physical meaning. In the following experiments, only the following function set is used:

$$\text{f} = \{\text{f\_insertV}, \text{f\_AddV}, \text{f\_add}, \text{f\_sub}, \text{f\_mul}, \text{f\_div}\}$$

IV. EXPERIMENTAL RESULTS

A. Experimental Setting

To evaluate the proposed approach, a standard strongly typed genetic programming with the function set described in Section III is implemented. The problem instances are generated randomly with a problem generator. In the experiments reported here, the width and length of the community area are both 1000 unit length. The cost of access points and wires is estimated seperately as relative fitness compard to a standard. The number of clients is set as 25 and 40 for two experiments discussed below. To make it simpler, all coordinates of the access points are integer values. The running parameters for both experiments are specified as below:

Population size: 1000
Maximum tree depth: 12
Initialization tree depth: 3-5
Crossover rate: 0.9
Mutation rate: 0.05
Maximum generations: 300

Since all wireless access points are connected together with wires, it is clear that the connection topology with minimal wire cost for a given number of access points is a minimal spanning tree. So there are two options for experimentation purpose. We

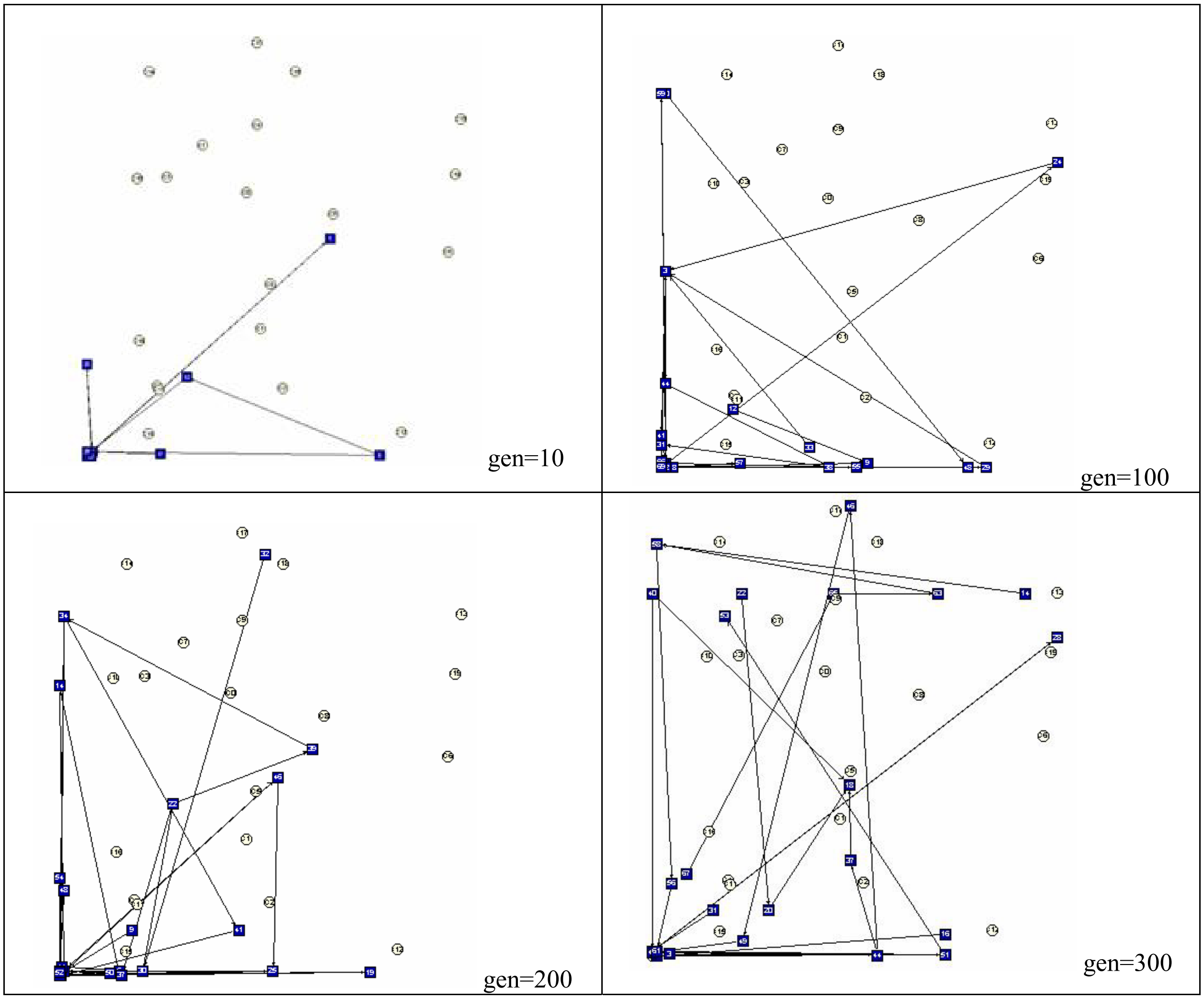

Figure 3. Best solution at generation 10, 100, 200, 300 by genetic programming without MST algorithm.

can let genetic programming to find this minimal spanning tree while optimizing the number of access points, their positions, and the connection topology. Another more problem-specific approach is to apply the minimal spanning tree (MST) algorithm to the evolved configuration of access points to determine the connection topology. As will be shown below, the latter approach can be much more efficient than the first approach.

B. Experimental Result without MST algorithm

Figure 3 shows the best solutions with a 25 clients problem over the evolution at generation 10, 100, 200, 300. There are some interesting observations of this evolutionary process. Since more reward is allocated to solutions that cover more clients, the evolution tends to focus on adding more access points into the solution to cover all clients and after that the cost minimization process is started by removing redundant access points and better wiring topologies. The second observation is that current formulation fitness tends to evolve complicated solutions with diversified connection topolgies, which is far from being optimal topolgy. This may result from the insufficient selection pressure regardig the wire cost. It is also found that there exists topology convergence problem. That's once a good configuration of access points and their topology is found, it is extremely difficult to find another better or competitive configurations with quite different configuration topology. One reason may be due to the unbalanced topological and parametric search discussed in [14]. Because it is much easier to find a relative good individuals by mutating only the parameters of current good individuals than by inserting or removing a random new access points, the population quickly converge to a few configurations. However, from Figure 3 (4), we can also find that the location of the access points seems to be reasonable except that the connection topology is a mess. To verify that, we did another set of experiments using minimal spanning tree algorithm to find the wiring topology.

C. Experimental Result with MST algorithm

Figure 4 shows the best solutions at generation 10, 100, 200, 300 for a 40 clients wireless access point configuration problem. One can find that the evolution dynamic of this experiment is quite different with the previous one although some similarity is shared. For example, the algorithm still works by growing the solution from partial coverage to full coverage by allocating more

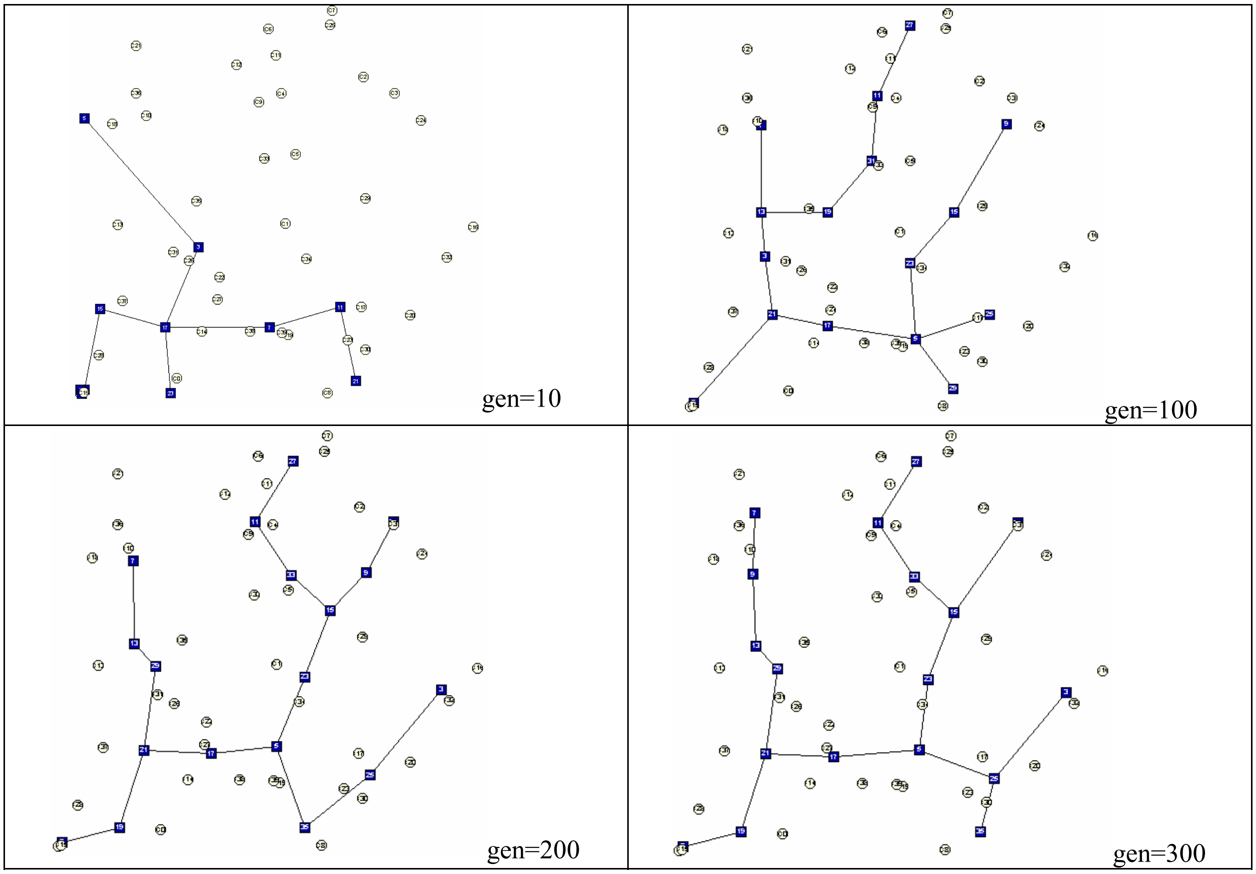

Figure 4. Best solution at generation 10, 100, 200, 300 by genetic programming with MST algorithm.

access points gradually. Another observation is that this approach with minimum spanning tree to connect access point has a much less problem with the topology convergence. Even at late evolutionary stage after 200 generations, large modification of the configuration, redundancy removing, and local fine-tuning continuingly happened. A careful analysis of the final result shows that this approach has found a very interesting and good solution, which demonstrates the applicability of genetic programming for wireless access point configuration problem.

V. DISCUSSIONS

Theoretically, both genetic algorithms and genetic programming can be used to evolve solutions with variable sizes and are suitable for the wireless access point configuration problem. However, the variable-length inherent variable-length representation and the capability to evolve open-ended solution gives genetic programming a clear advantage since no estimation of the minimal number of needed access points is needed. Comparison of the results of the two genetic programming approaches with or without minimum spanning tree algorithm shows that problem heuristic can greatly speed up the evolution process. We also find that it is difficult to define a good fitness function for a given problem. Usually, an unexpected bias may be introduced. For example, the insufficient pressure for topology in the experiment of Section IV.A leads to the evolution of somewhat arbitrary connection topology of access points.

Genetic programming is a versatile technique that can used to provide open-ended solutions. As an important application area, topologically open-ended synthesis by genetic programming has provided the most significant results for this field [15,16, 17]. Unfortunately, there does not exist a good set of benchmark problems that can be used to investigate the issues in this application area such as evolvability and encoding, parameter search in evolutionary synthesis, simultaneous search of topologies and parameters. Existing well-known benchmark problems including the Royal tree problem introduced by Punch et al. [18], symbolic regression problem [19], even parity problem [19], ant Santa Fe trail problem [19] are all not suitable for benchmarking techniques for automated synthesis by genetic programming. It is of course possible that one can use the analog circuit synthesis or the controller synthesis problem used by Koza [17] as the benchmark problems. The trouble is that these problems are very complex to set up and the simulation is a extremely time-consuming process.

Another possible option is to use the bond graph synthesis problem investigated in [13]. The proposed eigenvalue problem has all the features of the general topologically open-ended synthesis problem and the simulation can be pretty fast with their C++ simulators. However, the difficulty of understanding bond graphs, which is widely used in mechanical engineering though, may intimidates many genetic programming researchers from playing with that.

VI. CONCLUSIONS

In this paper, a strongly typed genetic programming is successfully applied to solve one version of the wireless access point configuration problem of wireless LAN. The variable length encoding feature and the capability of simultaneous search of both connection topology and the location parameters of the access points makes genetic programming an viable option for this problem.

The defined wireless access point problem promises to provide a new benchmark test problem[1] for genetic programming, which can be used to evaluate techniques and issues of strongly typed genetic programming, topologically open-ended evolutionary synthesis, representation and evolvability, and etc. This problem is characterized by its scalable problem size by including more clients of service, extensible difficulty by adding multiple design objectives and topological or parametric constraints, easy incorporation of domain specific heuristic.

Future works of this paper include defining a library of standard benchmark instances with different sizes that allows genetic programming researchers to compare against the performance of their techniques. As to the proposed application of this approach to real-world problems, it is desirable to incorporate average signal quality and other realistic performance requirements as additional design objectives later on. In addition, as genetic programming is usually regarded as more appropriate for variable-length optimization problem, it would be interesting to compare its performance with that of other approaches such as the hierarchical GA [11].

ACKNOWLEDGMENTS

The first author acknowledges the support of this work by the Dissertation Completion fellowship from the College of Engineering, Michigan State University, East Lansing, MI, 48823.

REFERENCES

[1] M. Kamenetsky, M. Unbehaun, "Coverage Planning for Outdoor Wireless LAN Systems", *Proc. IEEE International Zurich Seminar on Broadband Communications Access Transmission Networking*, IZS'2002, Zurich, Switzerland, February 2002

[2] M.H.Wright, "Optimization Method for Base Station Placement in Wireless Applications, " *Proc. IEEE Vehicular Technology Conference* 98, pp.287-391, May 1998.

[3] K. Lieska, E. Laitinen, and J. L¨ahteenm¨aki, "Radio coverage optimizationwith genetic algorithms," in *Proceedings of PIMRC*, Boston, MA,September 1998, vol. 1, pp. 318–322.

[4] H. Anderson and J. McGeehan, "Optimizing microcell base station locations using simulated annealing techniques," in *Proceedings of VTC*, Stockholm, Sweden, June 1994, vol. 2, pp. 858–862.

[5] S. Hurley, "Planning effective cellular mobile radio networks, " *IEEE Trans. On Vehicular Technolog*y, 51, pp243-253, Mar. 2002.

[6] C.Y.Lee and H.G. Kang, "Cell planning with capacity expansion in mobile communications: a tabu search approach, " *Proc. IEE VTC2000*, 49, pp.1678-1691, Sept. 2000

[7] E. Koichi and W. Yoshinori, "Automatic Cell Design for Wide Area Wireless LAN Systems," *NEC research & deveopment*, 44(4) 2003. Special Issue on Devices and Systems for Mobile Communications.

[8] Carlos A. Coello Coello, David A. Van Veldhuizen and Gary B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, New York, March 2002, ISBN 0-3064-6762-3.

[9] D. Martin, Adickes, M. D., Billo, R. E., Norman, B. A., Banerjee, S., Nnaji, B. O., and Rajgopal, J. "Optimization Of Indoor Wireless Communication Network Layouts," *Technical Report No. TR 99-5*, Department of Industrial Engineering, University of Pittsburgh.

[10] Lóránt Farkas, István Laki, Lajos Nagy, "Base Station Position Optimization in Microcells using Genetic Algorithms", *Proc. ICT'2001*, Bucharest, Romania, 2001.

[11] Kit-Sang Tang, Kim-Fung Man and S. Kwong, "Wireless Communication Network Design in IC Factory," *IEEE Transactions on Industrial Electronics*, Vol. 48, No. 2, pp. 452--459, April 2001.

[12] Gruau, F. 1992. Genetic Synthesis of Boolean Neural Networks with a Cell Rewriting Developmental Process. In *Proceedings of the International Workshop on Combinations of Genetic Algorithms and Neural Networks (COGANN-92)*, L. Darrell Whitley and J. David Schaffer, editors. 55–74. Los Alamitos: IEEE Press.

[13] K. Seo, J. Hu, Z. Fan, E. D. Goodman, and R. C. Rosenberg, "Automated Design Approaches for Multi-Domain Dynamic Systems Using Bond Graphs and Genetic Programming," *The International Journal of Computers, Systems and Signals*, vol.3, no.1, pp.55-70, 2002.

[14] J. Hu, K. Seo, S. Li, Z. Fan, R. C. Rosenberg, E. D. Goodman, "Structure Fitness Sharing for Evolutionary Design by Genetic Programming," *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO-2002, New York, July, 2002, pp. 780-787.

[15] J. R. Koza,, Keane, Martin A., and Streeter, Matthew J. 2003a. Evolving inventions. *Scientific American*. February 2003. 288(2) 52 – 59.

[16] J. R. Koza, Keane, Martin A., and Streeter, Matthew J. 2003c. What's AI done for me lately? Genetic programming's human-competitive results. *IEEE Intelligent Systems*. Volume 18. Number 3. May/June 2003. Pages 25 – 31.

[17] J. R. Koza, Keane, Martin A., Streeter, Matthew J., Mydlowec, William, Yu, Jessen, and Lanza, Guido. 2003. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers. ISBN 1-4020-7446-8.

[18] W.F. Punch, Dou- glas Zongker, and Erik D. Goodman, "The royal tree problem, a benchmark for single and multiple population genetic programming," Peter J. Angeline and K. E. Kinnear, Jr., editors, *Advances in Genetic Programming 2*, chapter 15, pages 299-316.

---

[1] Source code is available for download at http://www.egr.msu.edu/~hujianju/evograph/evograph.htm

[19] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.

[20] S. Luke, and L. Spector. Evolving Graphs and Networks with Edge Encoding: Preliminary Report. In Koza, John R. (editor), *Late-Breaking Papers at the Genetic Programming 1996* Conference. Palo Alto, CA., 1996.