An Evolutionary Approach to Planning IEEE 802.16 Networks

Ting Hu
Memorial University
St. John's, Canada
tingh@mun.ca

Yuanzhu Peter Chen
Memorial University
St. John's, Canada
yzchen@mun.ca

Wolfgang Banzhaf
Memorial University
St. John's, Canada
banzhaf@mun.ca

Robert Benkoczi
University of Lethbridge
Lethbridge, Canada
robert.benkoczi@uleth.ca

ABSTRACT

Efficient and effective deployment of IEEE 802.16 networks to service an area of users with certain traffic demands is an important network planning problem. We resort to an evolutionary approach in order to yield good approximation solutions. In our method, novel genetic variation operations are proposed to incorporate the feature of this real-world application of evolutionary algorithm.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search - *Heuristic Methods*

General Terms

Algorithms, Design, Experimentation

Keywords

Network Planning, Combinatorial Optimization, Genetic Algorithm

1. NETWORK PLANNING FORMULATION

We consider the *point-to-multipoint* PMP mode of WiMAX [2], where there can be two types of entities to form the wireless component of the network, the BSs and SSs. The BSs form the infrastructure for the SSs. An SS is allowed to communicate to a BS directly if the channel quality is sufficient for the given data rate. A network planning problem in this case is an optimization problem to cover the SSs in a geographical area using a small number of BSs.

The *network planning problem* can be formulated as a minimization problem on a weighted graph $G = (V, E)$. Specifically, there are two types of vertices in the graph, i.e., $V = B \cup S$, where B is the candidate basestation sites and S is the subscriber stations. For each $s \in S$ and $b \in B$, there is an edge between them if the channel gain $g(s, b)$ between s and b is greater than or equal to a given threshold δ for data reception. Therefore, graph G in this case is a bi-partite graph, where there are no edges within B or S themselves. Every $s \in S$ is associated with a capacity requirement of bandwidth c_s. The candidate basestation sites each have a capacity limit of C, which caps the total amount of bandwidth of its connected SSs.

A *feasible plan* is a mapping $M : S \mapsto B$ that satisfies the following constraints.

1. For each $s \in S$,
$$g(s, M(s)) \geq \delta. \tag{1}$$
2. If we define the *load* of a basestation $b \in B$ as
$$l(b) = \sum_{M(s)=b, s \in S} c_s$$
then we enforce a capacity limit on it, i.e.,
$$l(b) \leq C. \tag{2}$$

The total infrastructure cost of the network lies in the number of BSs in use. Therefore, our goal is to minimize $|M(S)|$ over all feasible plans.

2. EVOLUTIONARY APPROACH

The proposed framework is carefully designed to facilitate addressing this problem. We are aware that any simplicial application of GAs can be ineffective for a structured optimization problem like this.

2.1 Individual Representation

Given a set of subscriber stations S and a set of basestations B with their location information, we encode a mapping M from S to B as a two-tier chromosome. At the higher level, i.e., the *BS activation level*, we use an array of length $m = |B|$ to represent the BSs. In addition, each locus i of this chromosome stands for a BS b_i, referring to its service list containing all the SSs assigned to it. If there is no SS connected to a BS (i.e., this BS is not needed), its service list is $\emptyset$. This is the *SS assignment level*. Such a two-tier representation is depicted in Figure 1. Apparently, the total length of the service lists adds up to $|S|$, and for each b_i the total capacity demand in the list must not exceed the BS capacity for a solution to be feasible. Our goal is to minimize the total number of loci referring to non-empty service lists.

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.
ACM 978-1-60558-325-9/09/07.

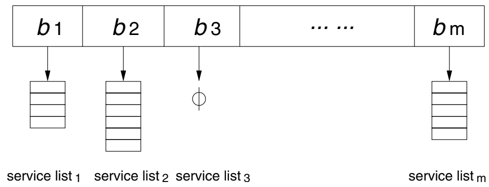

Figure 1: A chromosome representation

2.2 Evolution Framework

We evolve a constant-size population of individuals in the *steady-state mode* to approach the optimum. The process starts with randomly generating a population P of a given size. Next, each individual's fitness is evaluated. Then, the process enters an iteration outlined as follows.

1. Tournament selection of two parents out of P, denoted x and y.
2. Crossover between x and y.
3. Repair the offspring of previous step and denote them by x' and y'.
4. Mutate x' and y'.
5. Repair the output of previous step and denote them by x'' and y''.
6. Evaluate x'' and y'' and replace them back to P if they improve their parents x and y.
7. Go to Step 1 if not meeting termination condition.

The iterative process stops when the best fitness in P has remained the same for t (stagnation threshold) iterations.

2.3 Bi-polar Blend Crossover

A crossover is applied to two parents, denoted by $x = \langle x_1, x_2, \ldots, x_m \rangle$ and $y = \langle y_1, y_2, \ldots, y_m \rangle$, to obtain two children, $x' = \langle x'_1, x'_2, \ldots, x'_m \rangle$ and $y' = \langle y'_1, y'_2, \ldots, y'_m \rangle$. Our novel Bi-polar Blend method strives to move the SS assignment from less loaded BSs to more loaded ones so that some will eventually no longer be needed and can be deactivated. Such a crossover is a force to drive the activated BSs towards two extremes, either very heavily or very lightly loaded. Thus, more BSs are expected to be released. To do that, we define that x' inherits the greater load from its parents and y' inherits the less load. Specifically, for each locus i ($1 \leq i \leq m$), we define

$$x'_i = \begin{cases} x_i & \text{if } l(x_i) \geq l(y_i), \\ y_i & \text{otherwise,} \end{cases}$$

and

$$y'_i = \begin{cases} x_i & \text{if } l(x_i) < l(y_i), \\ y_i & \text{otherwise.} \end{cases}$$

2.4 Repair Heuristic

Note that an individual can become infeasible after the genetic variations. Therefore, we conduct the following greedy repair procedure upon a modified individual, denoted by x. For each $s \in S$, we consider all BSs in x that service it, denoted by $\bar{B}$. We first remove all overloaded elements in $\bar{B}$, i.e., load greater than C.

- If $\bar{B} \neq \emptyset$, we keep the most loaded element in $\bar{B}$ and release the rest of $\bar{B}$.
- Otherwise, i.e., s is not serviced by any BS, we search through all BSs within range to find the *best fit* if any. Here, by best fit we mean, when s is added, the BS that has the least residual capacity. If such a best fit exists, s is added to its load. Note that the identification of such a BS may imply activating a previously not-in-service candidate BS site. Else, however, we claim that x cannot be repaired and the current iteration is aborted and the evolutionary process continues with the next iteration.

This repair procedure is equally applicable to the output of both the crossover and mutation operations (next subsection). Note that it also works in such a general trend to drive the activated BSs towards two extremes that more lightly loaded BSs can be released.

2.5 Mutation

An individual is subject to a point mutation at the BS activation level. Specifically, we select an activated BS uniformly at random and simply clear its service list. We adopt such a mutation scheme for the following reasons.

- A mutation at the BS activation level, as apposed to the SS assignment level, yields sufficient genetic alteration for solution exploration. A mutation at the SS assignment level, in contrast, would yield a change which is usually too mild.
- Selecting a BS as a unit of mutation confines the changes to one locus of the network. It is, therefore, very well modularized.
- Random selection of an activated BS rather than deterministic, say the least loaded BS, is proved to be less directive and more effective in broadening the exploration space as per our preliminary tests.

As this mutation inevitably invalidates the solution, a subsequent repair procedure is also needed.

3. CONCLUSION

It is known that standard GA operations are fairly destructive to constrained combinatorial optimization problems and often lead to invalid solutions [1, 3]. To alleviate this problem, we devise crossover and mutation operations specifically for the network planning problem. The result of our BB crossover operation is two children with contrasty genetic features, one being thoroughly covering the SSs and the other being rather conservative. With these two extremes, a broader search space is explored, leading to the point of deleting under-utilized BSs gradually. Experiments show that our approach is effective and robust to different scenarios of network planning problem.

4. REFERENCES

[1] J. Hu and E. Goodman. Wireless Access Point Configuration by Genetic Programming. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation (CEC 2004)*, pages 1178–1184. IEEE Press, 2004.

[2] B. Li, Y. Qin, C. P. Low, and C. L. Gwee. A survey on Mobile WiMAX. *IEEE Communications Magazine*, 46(12):70–75, 2007.

[3] C. Reeves. Hybrid genetic algorithms for bin-packing and related problems. *Annals of Operations Research*, 63(3):371–396, 1996.