Random Tree Optimization for Energy-Efficient Broadcast in All-Wireless Networks

Fulu Li
Media Lab, MIT
Email: fulu@mit.edu

Shie Mannor
ECE, McGill University
shie@ece.mcgill.ca

Andrew Lippman
Media Lab, MIT
lip@media.mit.edu

Abstract

We study optimization methods for source-initiated energy-efficient broadcast in all-wireless networks. Past studies on energy-efficient broadcast [1,3,5] focus on *deterministic* optimization to construct an energy-efficient broadcast tree. We present a Random Tree Optimization (**RTO**) approach that transforms the *deterministic* optimization problem into a related *stochastic* one. We apply the cross-entropy (**CE**) method [2,4] to this problem. Preliminary results show that it achieves considerable power savings compared with state-of-the-art approaches.

I. The Problem Formulation

Given a source node and a group of intended destination nodes, say N destination nodes, in a wireless ad hoc network, the problem can be stated as *how to construct a broadcast tree such that the total required energy is minimal.* This problem has been proved to be NP-complete [1]. We assume that the power level of a transmission can be chosen within a given range of values and the use of *omni-directional* antennas at each node. Thus, all nodes within communication range of a transmitting node can receive its transmission, which is also known as wireless broadcast advantage (**WBA**) [5].

II. The RTO Process

We call the presented approach Random Tree Optimization (**RTO**) algorithm. As we will see that the algorithm operates *iteratively* by *randomly* generating improved sample trees till the *stochastic* process converges based on our predefined termination criteria and the performance function. The basic idea is to translate the *deterministic* optimization problem into a related *stochastic* optimization one and then use Rare Event Simulation (**RES**) techniques to find the solution.

First, we define the performance function $F(tree)$ as the total required power of a tree. There are two key components in RTO algorithm: (1) the random generation of the sample trees; (2) the update of the transition probability matrix at each iteration based on the performance in the previous round. The basic idea is that if it performs well for a given transmission in the previous round, it will have higher probability to transmit for the next round.

We use a Markov chain to construct a sample tree. We define $Q=(q_{i,j})_{((N+1)\times(N+1))}$ as the one-step transition matrix, where N is the number of destination nodes and $q_{i,j}$ denotes the probability that there is a transmission from node i to node j.

2.1. Initialization of Transition Probability Matrix

The initial matrix Q_0 can be set as follows: (a) the column corresponding to transmissions to the root node and the diagonal elements are zeros as no node transmits to itself and no node transmits to the root node;(b) for other elements $q_{i,j}$, we associate it with the reciprocal of the required power for this transmission and normalize it for each column.

2.2. Random Tree Generation

The random tree generation algorithm proceeds by randomly choosing a parent node based on the transition probability matrix for a given non-parented node (except the root node) among its non-descendent nodes till each destination node has a parent node.

2.3. Update of Transition Probability Matrix

At each iteration of the RTO algorithm based on the CE method, we need to calculate the benchmark value of γ_t as follows:

$$\gamma_t = \min\{f : Q_{t-1}(F(T)\le f)\ge \rho\}, \qquad (1)$$

where ρ normally takes a value of 0.01 so that the event of obtaining high performance is not too rare.

There are several choices to set the termination conditions. Normally, If for some $t \ge l$, say $l=5$,

$$\gamma_t = \gamma_{t-1} = \dots = \gamma_{t-l}, \qquad (2)$$

then stop the optimization process.

The updated value of $q_{i,j}$ can be estimated as:

$$q_{i,j}^{e} = \frac{\sum_{k=1}^{M} H_{\{F(T_k)\le\gamma\}} H_{\{T_k \in T_{i,j}\}}}{\sum_{k=1}^{M} H_{\{F(T_k)\le\gamma\}}}. \qquad (3)$$

where M stands for the number of sample trees, $H_{\{\}}$ is an indicator function, $T_{i,j}$ denotes the set of trees in which there is a transmission from node i to node j. Roughly, the transition probability estimation formula above states that if it has better performance for a given transition in the previous round, it will have bigger chance to transmit in the next round.

While there are solid theoretical justifications for Equation (3), we refer the readers to [2,4], and focus on the algorithms that were implemented in practice. In order to avoid overly quick convergence to 1s and 0s for the update of $q_{i,j}$, which could limit the randomness of the sample trees, normally we use a *smoothed* update procedure in which

$$q_{i,j}^{t} = \alpha \times q_{i,j}^{e} + (1-\alpha) \times q_{i,j}^{t-1}, \qquad (4)$$

where $q_{i,j}^{t-1}$ is the value of $q_{i,j}$ in the previous round and $q_{i,j}^{e}$ is the estimated value of $q_{i,j}$ based on the performance in the previous round according to Equation (3), and $q_{i,j}^{t}$ stands for the value of $q_{i,j}$ for the current round. Empirically, a value of α between $0.4 \le \alpha \le 0.9$ gives the best results [4].

2.4. The RTO Algorithm based on CE

1. *Set* $t = 1$ *and set* Q_0 *according to the initialization of* $q_{i,j}$ *in Sec. 2.1.*
2. *Randomly generate sample trees according to Sec. 2.2 (normally we generate* $20N^2$ *sample trees).*
3. *Calculate* γ_t *according to Equation (1).*
4. *Update* $q_{i,j}$ *according to Equation (3) and (4).*
5. *If for some* $t \ge l$, *say* $l = 5$, *such that* $\gamma_t = \gamma_{t-1} = \ldots = \gamma_{t-l}$, *then stop; otherwise, reiterate from step 2.*

III. Preliminary Results

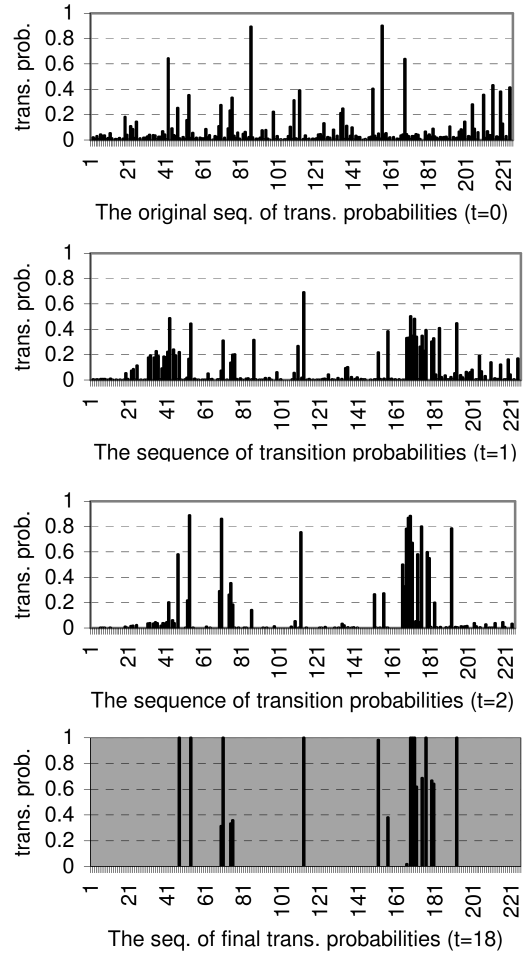

Figure 1: An example run of the evolving of the transition probability matrix during the RTO process.

Figure 1 shows an example run of the evolving of the transition probability matrix during the RTO process. In this example, the number of nodes in the network is 15 and the number of elements in the transition probability matrix is 225 (the index of the X-axis in the figure). The power attenuation factor is 2 and the initial transition probability matrix is obtained according to Section 2.1.

As typical to the CE method, the transition probabilities quickly converge, with some transition probabilities converging to one and others to zero. Essentially, transitions that lead to good solutions are reinforced and transitions that lead to poor solutions become diminished.

We consider many randomly-generated network examples (100 instances for each round of simulations), in which a specified number of nodes are randomly generated within a square region, say 10×10, the location of each node is *randomly* generated and the source node is *randomly* selected among the randomly-generated nodes. For the clarity and simplicity of the comparison, we use the same assumption as that in [5] that each node has enough power to cover all the other nodes and the power level of each node can be adjusted within a given range. We also consider the propagation loss exponents of λ =2 and λ = 3 in our experiments. For the performance comparison between the algorithms, we consider *normalized tree power*. For example, suppose we have three approaches to generate broadcast trees, say approaches A, B and C. Let p_A, p_B and p_C stand for the required tree power for the trees generated by approaches A, B and C, respectively for the *same* network topology. The normalized tree power for each of these approaches is given by the following:

$$p_A^{'} = \frac{p_A}{\min(p_A, p_B, p_C)},$$

$$p_B^{'} = \frac{p_B}{\min(p_A, p_B, p_C)},$$

$$p_C^{'} = \frac{p_C}{\min(p_A, p_B, p_C)}. \quad (5)$$

The mean normalized tree power and its standard deviation by RTO, BIP (Broadcast Incremental Power)[5] and MST (Minimum Spanning Tree) (averaged over 100 randomly generated network instances for each round of experiments) in a variety of circumstances are shown in Table 1. As shown in Table 1, RTO *consistently* outperforms BIP and MST in all circumstances. In particular with λ equal to 2, RTO can save up to **20%** and **30%** power compared with BIP and MST respectively.

EXPERIMENTS	RTO	BIP	MST
N =15, λ =2	1.0 ± 0.0	1.20 ± 0.12	1.325 ± 0.15
N =40, λ =2	1.02 ± 0.02	1.19 ± 0.13	1.26 ± 0.18
N =100, λ =2	1.0 ± 0.0	1.13 ± 0.03	1.22 ± 0.05
N =15, λ =3	1.01 ± 0.01	1.11 ± 0.06	1.16 ± 0.10
N =40, λ =3	1.02 ± 0.01	1.10 ± 0.06	1.16 ± 0.08
N =100, λ =3	1.0 ± 0.0	1.05 ± 0.03	1.07 ± 0.04

Table 1: Mean normalized tree power and its standard deviation by RTO, BIP and MST (averaged over 100 randomly generated network instances for each round of experiments) in a variety of circumstances (with different number of nodes in the network (N) and different power attenuation factor values (λ)).

IV. Conclusion and Future Directions

We considered energy-efficient broadcast in all-wireless networks since energy-efficiency is an important consideration for the design of wireless communication protocols due to the fact that the lifetime of a wireless network depends on the power consumption of each node, each of which is normally equipped with a limited power supply such as batteries. This problem has been introduced in [5] and it has received much attention in some recent studies [1,3,5].

The major contribution of this work is the development of the RTO algorithm, which is based on the cross-entropy method ([2,4]). We proposed a random tree generation algorithm based on the transition probability matrix and explored efficient ways to initialize the probability matrix in different circumstances. We conducted extensive experiments to examine the performance of RTO compared with other existing state-of-the-art approaches. Our empirical results indicate that it demonstrates the best performance of its kind.

One downside of the RTO algorithm is that the computation complexity of the procedure is in the order of $O(N^5)$ if we choose to generate 20 N^2 (empirically, this is a reasonable sample size) random sample trees at each round. We plan to consider other techniques for speeding-up the RTO algorithm. An interesting direction we plan to pursue is to implement the RTO algorithm in a distributed fashion, so that the nodes are divided into groups, on which sub-trees are generated separately and then merged to a single tree.

Acknowledgement

The authors would like to thank the Digital Life Consortium at MIT Media Lab for the support.

Reference

[1] Cagalj M., J. Hubaux, C. Enz, “Minimum-Energy Broadcast in All-Wireless Networks: NP-Completeness and Distribution Issues”, *in the proc. of ACM MOBICOM ‘2002*.

[2] De Boer P., D.P. Kroese, S. Mannor, R.Y. Rubinstein, “A Tutorial on the Cross-Entropy Method”, in *Annals of Operations Research*, 2004.

[3] Li F., I. Nikolaidia, “On Minimum-Energy Broadcast in All-Wireless Networks”, *in the Proc. of IEEE LCN ‘2001*.

[4] Rubinstein R., “The Cross-Entropy Method for Combinatorial and Continuous Optimization”, Methodology And Computing in Applied Probability, 1999.

[5] Wieselthier J., G. Nguyen, A. Ephremides, “On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks”, *in the proc. of IEEE INFOCOM ‘2000*, pp. 585-594.