Guided Maximum Entropy Method Algorithm for the Network Topology and Routing

Milan TUBA

***Abstract*—**This paper presents an algorithm that applies a guided maximum entropy method to the network design problem. Network design problem is a well known NP-hard problem which almost always involves underdetermined systems, especially when routing policy has to be determined. The maximum entropy method is a relatively new technique for solving underdetermined systems. We adjusted the network design problem, primarily the routing feasibility, to the maximum entropy method requirements. Computationally feasible algorithm is developed which includes additional constraints that direct uniformity of the solution in the desirable direction. Proposed algorithm computes a reasonable solution that is robust with respect to often required dynamic changes of the cost function. This modified method exploits the property of the MEM that it can smoothly move from cases where constraints can be satisfied to cases where constraints become desirable goals that are satisfied as much as possible. A software system was developed which includes all the mentioned features.

Keywords — Maximum entropy method, Network routing, Computer network topology, Optimization, Modeling.

I. Introduction

THE network design problem (NDP) is a very interesting NP-hard problem of great practical value and since it is untractable, heuristics and suboptimal solutions have been used for decades. It involves topology selection (subset of possible links), routing determination (paths for the offered traffic) and possibly capacity assignment. The goal is to minimize the cost, which can be a combination of the link costs and delay penalties, under possible additional constraints. Network design and analysis almost always involve underdetermined systems, especially when routing policy has to be determined. It is an open problem and since unique best solution can not be found, every new approach is promising in the sense that solution obtained can be better then previous ones, at least in some cases.

The maximum entropy method (MEM) is a relatively new technique for solving underdetermined systems which has been successfully applied in many different area. It is most frequently used in chemistry [1], but also in many other very diverse areas: character recognition [2], data analysis [3], image processing [4], [5], economy [6]. Theoretical developments also continue [7]. An analysis of both, network design problem and maximum entropy method, was done before [8] with the argument that maximum entropy method can be a reasonable way to approach the network design problem.

Manuscript received January 16, 2011.

The research was supported by the Ministry of Science, Republic of Serbia, Project No. III 44006

M. Tuba is with the Faculty of Computer Science, Megatrend University, Belgrade, Serbia, e-mail: tuba@ieee.org

It is intuitively clear that an optimal network should not have overloaded or underutilized links. The maximum entropy constraint favors uniform distribution and gives a starting topology and routing with smoothly distributed traffic that is expected to be close to the optimal solution. We adjusted the network design problem, primarily the routing feasibility, to the maximum entropy method requirements. Computationally feasible algorithm is developed which implements the standard maximum entropy method, includes adjustments for problems that do not involve probabilities initially, calculates a function that substitutes large sparse matrix, includes heuristic that speeds up calculations by avoiding to invert Jacobian matrix at each iteration, determines variables that define constraints for the routing feasibility, includes additional constraints that direct uniformity of the solution in the desirable direction, cancels opposing traffic and excludes underutilized links. Mentioned additional constraints are ”soft”, which is a unique feature of this algorithm, in the sense that they do not have to be satisfied; the solution will be pulled in the direction of satisfying them as much as possible. Some theoretical results are also established that direct initial approximation. Proposed algorithm computes a reasonable solution that is robust with respect to often required dynamic changes of the cost function. The maximum entropy solution can be a good starting point for further optimization considering that the cost function with delay penalties involves queuing theory that is usually computationally expensive.

II. The Maximum Entropy Method

The basic idea of the MEM is to get a unique solution from the underdetermined system by introducing the additional constraint that the entropy function should be maximized. The other methods that were used for solving underdetermined systems use the same technique: they introduce additional, artificial constraints that make the number of constraints equal to the number of unknowns. The difference is that the maximum entropy method introduces the most natural additional constraint: one that does not introduce any new, arbitrary and unwarranted information. It uses only the information that is given and makes no assumptions about missing information. Important property of the MEM is that it makes variables as equal as possible.

General MEM model calls for random variables and probabilities, but for most problems more suitable is a system of k equations with n variables $v_i,\ \ k < n$, and constraints:

$$\begin{aligned} x_{1,1}v_1 + x_{1,2}v_2 + ... + x_{1,n}v_n &= l_1 \\ x_{2,1}v_1 + x_{2,2}v_2 + ... + x_{2,n}v_n &= l_2 \end{aligned} \quad (1)$$

$$...$$

$$x_{k,1}v_1 + x_{k,2}v_2 + ... + x_{k,n}v_n \ = \ l_k$$

Variables v_i are converted to probabilities by normalization: $p_i \ = \ v_i / \sum_{j=1}^{n} v_j$ and $m_i \ = \ l_i / \sum_{j=1}^{n} v_j$. The system Equation (1) then becomes

$$\sum_{i=1}^{n} x_{r,i} p_i = m_r \ , \qquad r = 1, 2, ..., k \tag{2}$$

This is equivalent to the classical definition of the MEM where it is assumed that for a discrete random variable X the values $x_1,\ x_2, ...,\ x_n$ that it can take are known, but the corresponding probabilities $p_1,\ p_2, ...,\ p_n$ are not known. The expected values for $k \ < \ n-1$ functions of X (for example, the first k moments) are also known and represent constraints:

$$E[\ f_r(X)] \ = \ m_r \qquad r = 1, 2, ..., k. \tag{3}$$

Equation (2) (or (3)) gives (together with $\sum p_i = 1$) $\ k + 1 < n$ constraints for n unknown variables $p_1,\ p_2,\ ...,\ p_n$. This system is under-determined and has an infinite number of solutions. The unique solution is looked for that respects constraints and maximizes the entropy of the system:

$$H(p_1,\ p_2,\ ...,\ p_n) \ = \ -K \ \sum_{i=1}^{n} p_i \ ln(p_i)$$

The method of Lagrange multipliers is used. When Lagrange multipliers $\lambda,\ \mu_1,\ \mu_2, ...,\ \mu_k$ are introduced and partial derivatives equated with zero we get $n + k + 1$ equations for $n+k+1$ unknown variables $p_1,\ p_2, ...,\ p_n,\ \mu_1,\ \mu_2, ...,\ \mu_k,\ \lambda$. The system now has a unique solution, but it is not linear and some numerical method has to be used.

A. MEM Solution

The method of Lagrange multipliers is used. This will not guarantee that probabilities are non-negative. The substitution $p_i \ = \ e^{-q_i}$ is introduced, but this gives a stronger constraint than the one required: all probabilities are now positive definite (none of them can be zero). The problem now is to maximize

$$H(q_1,\ q_2, ...,\ q_n) \ = \ \sum_{i=1}^{n} q_i \ e^{-q_i} \tag{4}$$

under the conditions

$$\sum_{i=1}^{n} e^{-q_i} \ = \ 1 \tag{5}$$

$$\sum_{i=1}^{n} e^{-q_i} f_r(x_i) \ = \ m_r \ , \qquad r = 1, 2, ..., k \tag{6}$$

Lagrange multipliers $\lambda,\ \mu_1,\ \mu_2, ...,\ \mu_k$ are introduced with the function:

$$F(q_1,\ q_2, ...,\ q_n) \ = \ \sum_{i=1}^{n} q_i \ e^{-q_i} \ + \ \lambda \sum_{i=1}^{n} e^{-q_i} \tag{7}$$

$$+ \ \sum_{r=1}^{k} \mu_r \sum_{i=1}^{n} e^{-q_i} \ f_r(x_i)$$

All partial derivatives should be zero:

$$\frac{\delta F}{\delta q_i} \ = \ e^{-q_i} \ [1 - q_i - \lambda - \sum_{r=1}^{k} \mu_r \ f_r(x_i)] \ = \ 0 \ , \ \ i = 1, 2, ..., n \tag{8}$$

Since e^{-q_i} is never zero

$$q_i \ = \ 1 - \lambda - \sum_{r=1}^{k} \mu_r \ f_r(x_i) \ , \qquad i = 1, 2, ..., n \tag{9}$$

The problem is now solved: Equations (5), (6), and (9) give $n + k + 1$ equations for $n + k + 1$ unknown variables $p_1,\ p_2, ...,\ p_n,\ \mu_1,\ \mu_2, ...,\ \mu_k,\ \lambda$. The system should have unique solution, but it is not linear and some numerical method has to be used.

To make the calculations easier, the partition function is introduced:

$$Z(\mu_1,\ \mu_2, ...,\ \mu_k) \ = \ \sum_{i=1}^{n} p_i \ e^{-\lambda} \ = \ \sum_{i=1}^{n} e^{-\lambda - q_i}$$

$$Z(\mu_1,\ \mu_2, ...,\ \mu_k) \ = \ \frac{1}{e} \sum_{i=1}^{n} e^{\sum_{r=1}^{k} \ \mu_r \ f_r(x_i)} \tag{10}$$

It is easy to see that

$$\lambda \ = \ -lnZ(\mu_1,\ \mu_2, ...,\ \mu_k) \tag{11}$$

$$m_r \ = \ \frac{\delta}{\delta \mu_r} lnZ(\mu_1,\ \mu_2, ...,\ \mu_k) \tag{12}$$

or

$$m_r \ = \ \sum_{i=1}^{n} [m_r - f_r(x_i)] e^{\sum_{j=1}^{k} \ \mu_j \ f_j(x_i)} = 0 \ , \ \ r = 1, 2, ..., k \tag{13}$$

Equation (13) represents k equations for k unknown variables $\mu_1,\ \mu_2, ...,\ \mu_k$. When it is solved, from Equation (11) λ is calculated, and then from Equation (9) $q_1,\ q_2, ...,\ q_n$ are determined , and finally, from $p_i = e^{-q_i}$ the probabilities $p_1,\ p_2, ...,\ p_n$ are calculated.

Substitution $t_j = e^{\mu_j}, \ \ j = 1, 2, ...,\ k$ can be introduced. Then Equations (11) and (13) become:

$$\lambda \ = \ 1 - ln[\sum_{i=1}^{n} \Pi_{j=1}^{k} t_j^{f_j(x_i)}] \tag{14}$$

$$\sum_{i=1}^{n} [m_r - f_r(x_i)] \Pi_{j=1}^{k} t_j^{f_j(x_i)} \ = \ 0, \qquad r = 1, 2, ..., k \tag{15}$$

There is an algorithm to solve this system. However, the function that is to be minimized is not convex even in the simplest case when there is only one constraint: expected value. The standard Newton-Rapson procedure will not work. But the Jacobian matrix for this system is symmetric and positive definite. This gives a scalar potential function which is strictly convex and whose minimum is easy to find. The use of the second order Taylor expansion is recommended. However, after much experience with the algorithm, our impression is that it is not even worth trying to find the exact value for α that determines how far to go along a certain direction, let alone inverting the Jacobian matrix every time. For our software system we developed a heuristic that performs well.

B. Selection Principle

The previous model has constraints $p_i > 0,\ i = 1, 2, ..., n$. This may be too strong since the probabilities need only to be nonnegative. To make $p_i \geq 0$, $p_i = q_i^2$ can be introduced instead of $p_i = e^{-q_i}$, which was used before. In that case, the problem becomes to maximize

$$H(q_1,\ q_2, ...,\ q_n) \ = \ -2\sum_{i=1}^{n} q_i^2 ln(q_i) \tag{16}$$

under the conditions

$$\sum_{i=1}^{n} q_i^2 \ = \ 1 \tag{17}$$

$$\sum_{i=1}^{n} q_i^2 \ f_r(x_i) \ = \ m_r, \qquad r = 1, 2, ..., k \tag{18}$$

Lagrange multipliers are introduced:

$$F(q_1,\ q_2, ...,\ q_n) \ = \ -2\sum_{i=1}^{n} q_i^2 \ ln(q_i) \ + \ \lambda\sum_{i=1}^{n} q_i^2 \tag{19}$$

$$+ \ \sum_{r=1}^{k} \mu_r \sum_{i=1}^{n} q_i^2 \ f_r(x_i)$$

Partial derivatives should be zero:

$$\frac{\delta F}{\delta q_i} \ = \ -2q_i[2ln(q_i)+1-\lambda-\sum_{r=1}^{k}\mu_r f_r(x_i)] = 0,\ i = 1, 2, ..., n \tag{20}$$

Now, the selection has to be made: any q_i can be zero.

$$q_i = 0 \ \ or \ \ q_i \ = \ e^{\left(\ -1+\lambda+\sum_{r=1}^{k} \ \mu_r \ f_r(x_i) \ \right)^{0.5}}, \ \ i = 1, 2, ..., n \tag{21}$$

When it is decided which q_i are to be zero, the remaining equations will give as many equations as there are unknown variables. The partition function is equal as in the previous model, and the whole discussion repeats. The only difference is that summations are not carried for all $i = 1$ to n, but only for those i for which $q_i \neq 0$.

This new model is used only to show how the case p_i=0 for some i can be included. In practice, we have to decide which p_i will be zero. We can do it in advance and consider a model that has only $n - m$ probabilities (if m probabilities are selected to be zero). If we select too many probabilities to be zero, the system may become over-determined.

III. The Network Design Problem

Computer networks consist of computers, called nodes, and communication lines, called links, that interconnect them. The network design problem is:

- For given locations of nodes, traffic matrix (offered traffic for each pair of nodes) and cost matrix (cost to transfer a message for each pair of nodes)
- With performance constraints: reliability, delay (time that a message spend in the network), throughput
- Find values for variables: topology (which nodes will be connected directly with a line and which will have to communicate indirectly, using other nodes as intermediate stations), line capacities (how much traffic will each link be able to carry), flow assignment - routing (which paths messages between any pair of nodes will follow)
- Minimize the cost (of building and maintaining the whole network).

Other formulations of the problem are: minimize delay for the given cost or maximize throughput for given cost and delay. It has been shown that all these problems are similar and that the same techniques can be applied. Different aspects of the network design problem, particularly routing and link capacity were investigated [9], [10], [11]. More recent results are in [12] and [13] and the latest survey on topology [14].

This problem is intractable if full and exact solution is required. Networks can have many hundreds of nodes (computers). Fortunately, experience has shown that network design can be done hierarchically (or bi-level [15]) and still be near optimal. An example is a network for a country. First, we can decide where to put trunks between major cities, then connect small cities to nearest major cities, then make local networks inside the cities. This approach allows us to work with networks of at most 50 nodes at a time. This is a great help, but the problem is still intractable.

The network design problem, that was for many decades investigated with emphasis on wide area networks, is recently revitalized with application to mobile ad hoc networks [16], [17], [18]. The other refinements of the problem and areas of current research are radio networks where the goal is changed to covering maximum area [19] and quality of service over heterogeneous networks [20].

IV. Adjustment of the Network Design Problem for the MEM

The network design problem has to be fitted to the model described in the Section II. Let us consider a n-node network with given traffic matrix $t_{i,j}$, line capacity C and total traffic T. It is possible to apply MEM if analysis is started with totally interconnected network of n nodes. Initial feasible routing is then trivial. Some lines will be dropped later in the process of improving utilization or reducing the cost.

To apply the maximum entropy method, it has to be decided what will be the variables of the system. It may be desirable to have as variables the traffic along different lines; that is what should be made as equal as possible. However, these variables are too coarse. From them the routing can not be determined. The more serious problem is that there are no natural constraints on these variables. This forces us to select as variables of the system something finer: the traffic of a particular message type (message types are distinguished by the source and destination for a message) on a particular line [8].

The number of different message types is $n(n-1)$ (from each node to every other node, except itself). The number of different lines is also $n(n-1)$. There is a variable for each pair (message-type, line) so the total number of variables is $n^2(n-1)^2$. Constraints that enforce feasible routing can be determined as follows. For each node there is an equation for each message type. The total number of equations is then $n^2(n-1)$, plus the equation that establishes that the sum of all probabilities is equal to 1. In this case, the last condition is equivalent to the requirement that the total network traffic is equal to some given constant within a certain range. The equations will express the following conditions: for each transit node the flow-in is equal to the flow-out for each message type separately. For the source nodes and the sink nodes, equation is balanced by the required load for particular message type.

The matrix for this system is large, but fortunately very sparse. The total number of the elements in the system is$n^4(n-1)^3$ (the number of equations times the number of variables). The density of the matrix is then calculated as $\frac{2}{n^2(n-1)}$. The density approaches zero with the cube of the number of nodes, which means that is inappropriate or impossible to keep such a matrix in the memory. For example, for $n = 20$ there are 144,000 variables with 7,600 equations and density is only 0.003%. We implemented an algorithm for calculating matrix values.

V. THE COST FUNCTION

Among all possible topologies and associated routings we want to select one that is optimal in some sense, usually the combination of network cost and delay. In determining which line to keep and which to eliminate, an appropriate cost function is needed [9]. There is no unique best cost function because the network can be viewed from at least two different points: network manager's and user's. From the network manager's point of view a line that is expensive to install is expensive, but from the user's point of view a line that is introducing long delays is expensive. This two criteria are always contradictory. The best solution is usually some compromise between these two extreme positions. A line cost can be defined as a weighted sum (or some other function) of the installation cost and the total delay on that line. General form of the cost function can be $C = C_I + KD$. The network cost is the sum of line costs. When the weight coefficient K is set to zero, delays are ignored and when it is set to some very large value only delays are considered. The second cost component, total delay, is a dynamic component and it has to be recalculated after each rerouting.

It is easy to see that two extremes do not give reasonable results. If only delays are considered, the best network will always be totally interconnected network. Removing any line will increase delays. But some very expensive line may carry very little traffic and the removal of such line would significantly decrease line costs and only marginally increase delays. Such solution would be overlooked if line costs are not considered.

The other extreme is when only line costs are considered. The best network in that case is the minimum spanning tree. Interesting case is a network that forms a ring when costs are considered. Each node has two neighbors to which it can be connected by inexpensive lines. Connections to any other node is considerably more expensive. The minimum spanning tree for such a network is an open ring. That is the solution if only line-costs are considered. It is easy to see that a closed ring is much better solution. By adding that last line that will close the ring, the cost will not increase dramatically, but the average path length will be almost halved and delay will be much smaller. If delay is included, even with a small weight coefficient, in the cost function, the line that closes the ring would not be dropped.

The cost function can dynamically change and that is the reason that robust solution is needed. Evolutionary algorithms [21] that may have very good properties can be to slow for such dynamic adjustments.

VI. NDP ALGORITHM BASED ON MEM

An algorithm is presented here that uses guided MEM to get a robust solution for the NDP. It first gives the maximum entropy solution (routing) for the system described in Section IV. It was mentioned that some numerical method is needed to solve nonlinear system that defines MEM solution. There is an algorithm to solve this system. However, the function that is to be minimized is not convex even in the simplest case when there is only one constraint: expected value. The standard Newton-Rapson procedure will not work. But the Jacobian matrix for this system is symmetric and positive definite. This gives a scalar potential function which is strictly convex and whose minimum is easy to find. The use of the second order Taylor expansion is recommended. However, after much experience with the algorithm, our impression is that it is not even worth trying to find the exact value for α that determines how far to go along a certain direction, let alone inverting the Jacobian matrix every time. For our software system we developed a heuristic that performs well.

The other problem that was mentioned is that matrix for the system is very large, but fortunately very sparse. A function is implemented that calculates the value of matrix element without need to store that element.

After the initial solution is obtained some refinements are done. It is never a good idea to have traffic of certain messages from A to B and from B to A, for any pair of nodes A and B. The maximum entropy method avoids such situations but it can not make any probability exactly zero. In the second pass we eliminate one half of the variables. For each message type and each pair of nodes we keep traffic only in one direction. For the direction where it was near zero, we cancel it. After that we have routing and can do something about topology (we start with a totally interconnected network), for example to exclude lines that carry little traffic.

The algorithm is applied to a simple three node network as an example. Tables I and II give results for total traffic 90, full duplex and $\epsilon = 0.0001$. The column 3 is the initial MEM solution when all lines are included, column 4 is refinement when opposing traffic is canceled, column 5 excludes lines where traffic is less than 4% and the last column excludes lines where traffic is less than 9%.

TABLE I
TRAFFIC DISTRIBUTION, TOTAL LOAD 90

Line	Offered	All lin.	1-way	Tr. 4%	Tr. 9%
(1,2)	15.0	18.2	16.6	18.7	10.0
(1,3)	5.0	11.9	13.7	10.0	10.0
(2,1)	15.0	18.2	16.6	18.7	10.0
(2,3)	10.0	14.8	14.7	16.3	25.0
(3,1)	5.0	11.9	13.7	10.0	10.0
(3,2)	10.0	14.8	14.7	16.3	25.0

TABLE II
ROUTING, TOTAL LOAD 90

Mess.	Line	All l.	1-way	Tr. 4%	Tr. 9%
(1,2)	(1,2)	23.5	16.6	18.7	10.0
(1,2)	(1,3)	7.5	13.4	11.3	20.0
(1,2)	(2,1)	0.2			
(1,2)	(2,3)	0.8			
(1,2)	(3,1)	0.8			
(1,2)	(3,2)	7.5	13.4	11.3	20.0
(1,3)	(1,2)	4.3	6.3	10.0	10.0
(1,3)	(1,3)	7.8	3.7		
(1,3)	(2,1)	1.4			
(1,3)	(2,3)	4.3	6.3	10.0	10.0
(1,3)	(3,1)	0.8			
(1,3)	(3,2)	1.4			
(2,3)	(1,2)	1.0			
(2,3)	(1,3)	6.1	10.3	8.7	
(2,3)	(2,1)	6.1	10.3	8.7	
(2,3)	(2,3)	15.3	9.7	11.2	20.0
(2,3)	(3,1)	1.0			
(2,3)	(3,2)	0.4			

The total offered load is for this example is 60. The shortest path is of length 1 and the longest path is of length 2. That means that minimal total traffic is 60 and maximal total traffic is 120 (without cycles). These two cases have unique solutions (for total traffic 60 everything goes along the shortest path and for total traffic 120 everything goes along the longest path) and we do need the maximum entropy method for that. We would not be able to get maximum entropy solutions for these cases since many probabilities are zero and the maximum entropy method can not force any probability to zero. But if we put 60.1 or 119.9 for the total traffic, we get very reasonable results. Tables III i IV show how MEM successfully routes traffic near extreme points along shortest (columns 3 and 4) or longest (columns 5 and 6) path:

TABLE III
TRAFFIC 60.1 I 119.9

Line	Offered	Min all	min-1-w	Max all	max-1-w
(1,2)	15.0	15.0	15.0	15.2	15.0
(1,3)	5.0	5.0	5.0	24.8	25.0
(2,1)	15.0	15.0	15.0	15.2	15.0
(2,3)	10.0	10.0	10.0	20.0	19.9
(3,1)	5.0	5.0	5.0	24.8	25.0
(3,2)	10.0	10.0	10.0	20.0	19.9

TABLE IV
ROUTING FOR TRAFFIC 60.1 I 119.9

Mess.	Line	min all	min-1-w	max all	max-1-w
(1,2)	(1,2)	30.0	30.0	0.4	
(1,2)	(1,3)	0.0		29.7	29.9
(1,2)	(3,2)	0.0		29.7	29.9
(1,3)	(1,2)	0.0		10.0	10.0
(1,3)	(1,3)	10.0	10.0	0.1	
(1,3)	(2,3)	0.0		10.0	10.0
(2,3)	(1,3)	0.0		19.9	20.1
(2,3)	(2,1)	0.0		19.9	20.1
(2,3)	(2,3)	20.0	20.0	0.2	

We said before that our goal is to make traffic along all lines as equal as possible. We can keep the constraints and include additional equations that will force the traffic on all lines to be exactly equal. This is exactly what we wanted. The problem is that there is a range for total traffic where this is possible. It is obvious that the total traffic that is close to its extreme values will not permit equal traffic on all lines (provided that all loads are not equal). For the previous case the limit where traffic on all lines can be made equal is when the total traffic is 75. At that point some probabilities become zero, and if we drop the total traffic below 75 we can not get equal traffic on all lines any more.

The better approach is to drop the requirement (which can not be satisfied any more) that traffic on all lines must be equal and introduce new variables that will represent traffic on different lines. They are connected to old variables and will be included as additional constraints. Since traffic on each line is a variable now, these variables will be made as equal as possible by the MEM. The problem is that they are not the only variables. Since we really want to make them equal, we can give them larger weight coefficients. This works remarkably well and a weight coefficient of 10 or 20 gives very nice solutions.

VII. THE GUIDED MEM

For many problems initial adjustment for the MEM application requires that variables of the system be determined in such a way that a feasible solution is obtained. This may not be a desirable solution for the optimization, but constraints have to be satisfied first.

It is possible to modify the MEM model and include a mechanism to guide the process of optimization. Once the necessary constrains are satisfied, artificial variables can be introduced that will guide the optimization process in the desirable direction.

MEM guidance will be demonstrated on an example, similar to Brandeis Dice Problem.

A die, possibly irregular, is considered. The number of spots that shows up when the die is tossed defines a random variable with possible outcomes and corresponding probabilities:

$$X = [1, 2, 3, 4, 5, 6]$$

$$P_{(6)} = [p_1, p_2, p_3, p_4, p_5, p_6]$$

The constraint that the sum of the probabilities is 1 is always present and in usual terminology not counted as an additional constraint. Without any (additional) constraints the expected value $E(X)$ is 3.5 and the solution for the probabilities is an uniform distribution: $p_i = 0.167, \quad i = 1, 2, ...6$.

For a single constraint EX=4.4 there is one (additional) constraint:

$$1p_1 + 2p_2 + 3p_3 + 4p_4 + 5p_5 + 6p_6 = 4.4$$

and the MEM solution is:

$$P_{(6)} = [0.063, 0.087, 0.121, 0.169, 0.234, 0.325]$$

As expected, the probabilities density is shifted towards larger outcomes since expected value shifted in that direction.

If the elementary probabilities were not the goal of equalization but some coarser variables, additional constraint can be included. If, for example, the goal is to to make $p_x = p_1+p_2+p_3$ equal to $p_y = p_4 + p_5 + p_6$, a system of two constraints can be used:

$$1p_1 + 2p_2 + 3p_3 + 4p_4 + 5p_5 + 6p_6 = 4.4$$

$$1p_1 + 1p_2 + 1p_3 - 1p_4 - 1p_5 - 1p_6 = 0$$

In this case it is possible to have a solution that will satisfy both constraints:

$$P_{(6)} = [0.004, 0.042, 0.454, 0.004, 0.042, 0.454] \qquad (22)$$

The problem with this approach is that it limited to cases when the guidance goal (in this case the total equalization of p_x and p_y) is possible. However, the main advantage of the MEM method is its ability to push towards the guidance goal even when exact goal satisfaction is not possible.

This can be illustrated on the previous example, but with changed requirement that $E(X) = 4.6$. It is easy to see that the constraint

$$p_1 + p_2 + p_3 = p_4 + p_5 + p_6$$

can not be satisfied. The maximum value for $E(X)$ is reached when probabilities density is pushed toward higher values:

$$P_{(6)} = [0, 0, 0.5, 0, 0, 0.5]$$

The value for $E(X)$ is in that case equal to 4.5. For any higher value of $E(X)$ exact equalization (which is the second constraint) is not possible.

To make the sums $p_1+p_2+p_3$ and $p_4+p_5+p_6$ as equal as possible, new variables are introduced: $p_6 = p_x = p_1+p_2+p_3$ and $p_7 = p_y = p_4 + p_5 + p_6$. Two new constraints that define these new probabilities are added. The fact that new variables are mentioned as constraints will make them participate in the equalization process.

Care mast be taken about normalization. New probabilities (p_7 and p_8) are not independent from the old ones and the sum of all probabilities becomes 2. Considering that the sum of all probabilities has to be 1 and that the sum of old probabilities (only old probabilities participate in the first constraint) is only 0.5, the first constraint has to be redefined.

Three constraints now become:

$$1p_1 + 2p_2 + 3p_3 + 4p_4 + 5p_5 + 6p_6 + 0p_7 + 0p_8 = 2.3$$

$$1p_1 + 1p_2 + 1p_3 + 0p_4 + 0p_5 + 0p_6 - 1p_7 + 0p_8 = 0$$

$$0p_1 + 0p_2 + 0p_3 + 1p_4 + 1p_5 + 1p_6 + 0p_7 - 1p_8 = 0$$

and the corresponding MEM solution is:

$$P_{(8)} = [0.020, 0.039, 0.076, 0.055, 0.106, 0.205, 0.135, 0.365]$$

or, when only $P_{(6)}$ is denormalized:

$$P_{(6)} = [0.040, 0.078, 0.152, 0.109, 0.211, 0.409]$$

This solution represents smooth extrapolation of the previous case. All constraints are satisfied. Expected value is 4.6. However, p_7 and p_8 are not equal since that was not the requirement any more. These variables were mentioned in the system of constraints so they participate in the process of equalization, but only to some extent. In this case (after denormalization), $p_7 = 0.270$ and $p_8 = 0.730$. This is far from being equal, the ratio p_8/p_7 is 2.7. We can make them closer to being equal by forcing them to contribute more significantly in the optimization process. This can be accomplished by redefining them in such a way that the larger mass of the probability is concentrated in them. If the constraints $p_6 = p_1 + p_2 + p_3$ and $p_7 = p_4 + p_5 + p_6$ are replaced with $p_6 = 9p_1 + 9p_2 + 9p_3$ and $p_7 = 9p_4+9p_5+9p_6$ only the 10% of the probability mass will remain in the old probabilities and 90% will be concentrated in the new probabilities. This will make new probabilities more significant in the equalization process, but the first constraint has to be redefined to reflect the fact that old probabilities, that define it, now contribute 10 times less. The new set of constraint is:

$$1p_1 + 2p_2 + 3p_3 + 4p_4 + 5p_5 + 6p_6 + 0p_7 + 0p_8 = 0.46$$

$$9p_1 + 9p_2 + 9p_3 + 0p_4 + 0p_5 + 0p_6 - 1p_7 + 0p_8 = 0$$

$$0p_1 + 0p_2 + 0p_3 + 9p_4 + 9p_5 + 9p_6 + 0p_7 - 1p_8 = 0$$

The corresponding MEM solution is:

$$P_{(8)} = [0.001, 0.007, 0.031, 0.002, 0.010, 0.049, 0.348, 0.552]$$

or, when only $P_{(6)}$ is denormalized:

$$P_{(6)} = [0.014, 0.066, 0.307, 0.022, 0.104, 0.487]$$

New probabilities p_7 and p_8 are now closer to being equal since ratio p_8/p_7 is 1.6.

We can push this process further in that direction by making old probabilities contain only 2% of the probability mass, which is equivalent of making new probabiities 50 times more important.

The new set of constraints is now:

$$1p_1 + 2p_2 + 3p_3 + 4p_4 + 5p_5 + 6p_6 + 0p_7 + 0p_8 = 0.092$$

$$49p_1 + 49p_2 + 49p_3 + 0p_4 + 0p_5 + 0p_6 - 1p_7 + 0p_8 = 0$$

$$0p_1 + 0p_2 + 0p_3 + 49p_4 + 49p_5 + 49p_6 + 0p_7 - 1p_8 = 0$$

The corresponding MEM solution is:

$$P_{(8)} = [0.000, 0.000, 0.009, 0.000, 0.000, 0.010, 0.444, 0.536]$$

or, when only $P_{(6)}$ is denormalized:

$$P_{(6)} = [0.001, 0.019, 0.434, 0.001, 0.023, 0.523]$$

New probabilities p_7 and p_8 are now even closer to being equal since ratio p_8/p_7 improved to 1.2.

For significance of new probabilities equal to 100, the corresponding probabilities are $P_{(8)} = $ [0.000000, 0.000038, 0.004603, 0.000000, 0.000044, 0.005314, 0.459509, 0.530491], $P_{(6)} = $ [0.0000, 0.0038, 0.4603, 0.0000, 0.0044, 0.5314] and ratio $p_8/p_7 = 1.15$.

The process that is described shows that it is possible to adjust MEM for some constrained optimization problem and then guide it in the desired direction, but there is no universal way how to do it, each problem has to be investigated separately.

VIII. Network Example for Guided MEM

Tables V and VI represent the same example as before, total traffic 90, but with constraints where traffic on all lines is made equal, lines are introduced as variables, and lines are introduced as variables with weight 4.

TABLE V
LINES AS VARIABLES, TOTAL TRAFFIC 90

Line	Offered	Old	Eq.	Var.	Weight 4
(1,2)	15.0	16.6	15.0	16.1	15.6
(1,3)	5.0	13.7	15.0	14.0	14.4
(2,1)	15.0	16.6	15.0	16.1	15.6
(2,3)	10.0	14.7	15.0	14.9	15.0
(3,1)	5.0	13.7	15.0	14.0	14.4
(3,2)	10.0	14.7	15.0	14.9	15.0

The limit where we can force equal traffic on all lines is when total traffic is 75 for this case. Then the traffic for some

TABLE VI
ROUTING, LINES AS VARIABLES, TOT 90

Mess.	Line	Old	Eq.	Var.	Weight 4
(1,2)	(1,2)	16.6	15.0	16.1	15.5
(1,2)	(1,3)	13.4	15.0	13.9	14.5
(1,2)	(3,2)	13.4	15.0	13.9	14.5
(1,3)	(1,2)	6.3	5.0	6.0	5.6
(1,3)	(1,3)	3.7	5.0	4.0	4.4
(1,3)	(2,3)	6.3	5.0	6.0	5.6
(2,3)	(1,3)	10.3	10.0	10.1	10.0
(2,3)	(2,1)	10.3	10.0	10.1	10.0
(2,3)	(2,3)	9.7	10.0	9.9	9.9

messages on some lines drops to zero. Tables VII and VIII show what the second method can do in that case. The third column gives old results, the column 4 results of modified algorithm when there is a constraint that traffic on all lines be equal, column 5 when lines are introduced as variables, column 6 when these lines variables have weight 4, column 7 with weight 9 and the last column with weight 19. If the total traffic drops below 75, the fourth column can not be calculated any more, but the remaining columns continue to smoothly abandon the uniform distribution.

TABLE VII
DIFFERENT WEIGHT COEFFICIENTS

Line	Offer	Old	Eq.	Wght4	Wght9	Wght19
(1,2)	15.0	16.0	12.5	13.9	13.1	12.7
(1,3)	5.0	9.1	12.5	11.0	11.7	12.3
(2,1)	15.0	16.0	12.5	13.9	13.1	12.7
(2,3)	10.0	12.4	12.5	12.7	12.6	12.5
(3,1)	5.0	9.1	12.5	11.0	11.7	12.3
(3,2)	10.0	12.4	12.5	12.7	12.6	12.5

TABLE VIII
ROUTING FOR DIFFERENT WEIGHT COEFFICIENTS

Mess.	Line	Old	Eq.	Wght4	Wght9	Wght19
(1,2)	(1,2)	23.6	19.9	20.3	18.4	16.0
(1,2)	(1,3)	6.5	10.0	9.2	10.4	11.8
(1,2)	(3,2)	6.5	10.0	9.1	10.1	11.0
(1,3)	(1,2)	3.4	0.1	2.2	2.1	3.0
(1,3)	(1,3)	6.6	9.9	7.4	6.9	5.3
(1,3)	(2,3)	3.4	0.1	2.2	2.2	3.3
(2,3)	(1,3)	5.1	5.0	5.4	6.1	7.4
(2,3)	(2,1)	5.1	5.0	5.3	5.7	6.3
(2,3)	(2,3)	14.9	14.9	14.1	13.0	10.7

IX. Conclusion

The network design problem is suitable for the maximum entropy method application since the routing problem is an underdetermined one. Also, since it is intuitively clear that an optimal network should not have overloaded or underutilized links, the maximum entropy constraint gives a starting topology and routing with smoothly distributed traffic that is robust to changes in cost function. Such optimization is useful in ad-hoc and wireless networks where cost function and consequently, topology and routing have often to be quickly adjusted. An algorithm presented here has a number of automatic features that, step by step improve solution, but also a number of parameters that can be adjusted for specific cases to help the optimization process. Further research can include quantitative analysis of robustness of this MEM solution considering different realistic cost functions.

REFERENCES

[1] Ding YS, Zhang TL, Gu Q, et al., Using Maximum Entropy Model to Predict Protein Secondary Structure with Single Sequence *Protein and Peptide Letters* Volume 16, Issue 5, pp. 552-560, 2009

[2] Xuan Wang, Lu Li, Lin Yao, Anwar, W., A Maximum Entropy Approach to Chinese Pin Yin-To-Character Conversion. *2006 IEEE International Conference on Systems, Man, and Cybernetics*, 2006, Taipei, Taiwan

[3] Teh, Chee Siong Lim, Chee Peng, A probabilistic SOM-KMER model for intelligent data analysis, *WSEAS Transactions on Systems* Vol. 5, no. 4, pp. 825-832. Apr. 2006

[4] Zhengmao Ye, Habib Mohamadian1, Yongmao Ye, Practical Approaches on Enhancement and Segmentation of Trimulus Color Image with Information Theory Based Quantitative Measuring, *WSEAS Transactions on Signal Processing*, Issue 1, Volume 4, January 2008, pp. 12-20

[5] Heric, Dusan; Zazula, Damjan, Reconstruction of Object Contours Using Directional Wavelet Transform, *WSEAS Transactions on Computers*. Vol. 4, no. 10, pp. 1305-1312. Oct. 2005

[6] Ciavolino E, Dahlgaard JJ, Simultaneous Equation Model based on the generalized maximum entropy for studying the effect of management factors on enterprise performance, *Journal of Applied Statistics* Volume 36, Issue 7, pp. 801-815, 2009

[7] Aladdin Shamilov, Generalized entropy optimization problems and the existence of their solutions, *Physica A: Statistical Mechanics and its Applications*, Volume 382, Issue 2, August 2007, pp. 465-472

[8] Tuba, Milan: Maximum Entropy Method and Underdetermined Systems Applied to Computer Network Topology and Routing, in *Recent Advances in Applied Informatics and Communications*, WSEAS Press 2009, pp. 127-132

[9] Tuba, Milan: Cost Function for Communication Links in Computer Networks, *Bulletins for Applied Mathematics (BAM)*, LXXIII-1028/94, pp. 115-122, Budapest, 1994

[10] Tuba, Milan: A Mathematical Model for Routing Comparison in Computer Networks, *Bulletins for Applied Mathematics (BAM)*, LXXXVI-A 1565/98, Arad, July 1998, pp. 493-503

[11] Tuba, Milan: Parameters for the Internet Optimization on the Local Level, *Applied & Computing Mathematics*, Vol. II, pp. 139-142, Kosice, 1997

[12] Tuba, Milan: Relation between Static and Dynamic Optimization in Computer Network Routing, *Recent Advances in Artificial Intelligence, Knowledge Engineering and Data Bases*, WSEAS Press 2009, pp. 484-489

[13] Tuba, Milan: Computer Network Routing Based on Imprecise Routing Tables, *WSEAS Transactions on Communications*, Issue 4, Volume 8, April 2009, pp. 384-393

[14] Abd-El-Barr M: Topological network design: A survey, *Journal of Network and Computer Applications* Vol. 32, Issue 3, 2009, pp. 501-509

[15] Obreque C, Donoso M, Gutierrez G, Marianov V: A branch and cut algorithm for the hierarchical network design problem, *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH*, Vol. 200 Issue 1, Jan. 2010, pp. 28-35

[16] Karavetsios, P., Economides, A.: Performance Comparison of Distributed Routing Algorithms in Ad Hoc Mobile Networks, *WSEAS Transactions on Communications*, Vol. 3, Issue 1, 2004, pp. 317-321

[17] Sokullu, R., Karaca, O.: Comparative Performance Study of ADMR and ODMPR in the Context of Mobile Ad Hoc Networks and Wireless Sensor Networks, *International Journal of Communications*, Issue 1, Volume 2, 2008, pp. 45-53

[18] Kumar, D., Bhuvaneswaran, R.: ALRP: Scalability Study of Ant Based Local Repair Routing Protocol for Mobile Ad Hoc Networks, *WSEAS Transactions on Computer Research*, Vol. 3, Issue 4, Apr 2008, pp. 224-233

[19] Mendes SP, Molina G, Vega-Rodriguez MA, Miguel A, Gomez-Pulido JA, Saez Y, Miranda G, Segura C, Alba E, Isasi P, Leon C, Sanchez-Perez JM: Benchmarking a Wide Spectrum of Metaheuristic Techniques for the Radio Network Design Problem, *IEEE Transactions on Evolutionary Computation* Vol. 13, Issue 5, Oct. 2005, pp. 1133-1150

[20] Awan I, Al-Begain K: An analytical study of quality of service provisioning for multi service mobile IP networks using adaptive buffer management, *11th International Conference on Analytical and Stochastic Modeling Techniques and Applications*, 2004, Proceedings, pp. 166-172

[21] Watcharasitthiwat K, Wardkein P: Reliability optimization of topology communication network design using an improved ant colony optimization, *Computers & Electrical Engineering* Vol. 35, Issue 5, Sep. 2009, pp. 730-747

Milan Tuba received B. S. in Mathematics, M. S. in Mathematics, M. S. in Computer Science, M. Ph. in Computer Science, Ph. D. in Computer Science from University of Belgrade and New York University.

From 1983 to 1994 he was in the U.S.A. first as a graduate student and teaching and research assistant at Vanderbilt University in Nashville and Courant Institute of Mathematical Sciences, New York University and later as an Assistant Professor of Electrical Engineering at Cooper Union Graduate School of Engineering, New York. During that time he was the founder and director of Microprocessor Lab and VLSI Lab, leader of scientific projects and supervisor of many theses. From 1994 he was Associate professor of Computer Science and Director of Computer Center at University of Belgrade, Faculty of Mathematics, and from 2004 also a Professor of Computer Science and Dean of the College of Computer Science, Megatrend University Belgrade. He was teaching more than 20 graduate and undergraduate courses, from VLSI Design and Computer Architecture to Computer Networks, Operating Systems, Image Processing, Calculus and Queuing Theory. His research interest includes mathematical, queuing theory and heuristic optimizations applied to computer networks, image processing and combinatorial problems. He is the author of more than 100 scientific papers and a monograph. He is coeditor or member of the editorial board or scientific committee of number of scientific journals and conferences.

Prof. Tuba is member of the ACM since 1983, IEEE 1984, New York Academy of Sciences 1987, AMS 1995, SIAM 2009. Participated in many WSEAS Conferences with plenary lectures and articles in Proceedings and Transactions.