Bounds and Approximations on the Reliability of Large Networks

Tamás Szántai, József Bukszár and Edith Kovács

***Abstract*— We present methods for calculating bounds and approximations on the reliability of large networks. We regard the most reliable few paths leading from the start node to the terminal node of the network and then calculate lower and upper bounds on the probability that at least one of these paths is permeable. For this purpose we will use the so called multitree upper and hypermultitree lower bounds on the probability of union of events, developed earlier by J. Bukszár. Chow-Liu's dependence tree approximation of multivariate discrete probability distributions was generalized earlier by T. Szántai and E. Kovács to higher order dependence tree approximations, called k-th order t-cherry junction tree approximations. We will show how the Chow-Liu's dependence tree and the more general t-cherry junction tree graph structures can be used for bounding and approximating the network reliability. Some possible applications and numerical results will also be presented.**

I. INTRODUCTION

Consider a compact, directed and acyclic network $(\mathscr{N},\mathscr{A})$. Assume that $\mathscr{N}=\{c_1,\ldots,c_n\}$ is the set of nodes, and $\mathscr{A}\subset\mathscr{N}\times\mathscr{N}$ is the set of arcs. Without restricting generality, we may assume that there is exactly one node such that no arc leads into it and there is exactly one node such that no arc goes out of it. These two nodes will be called start and terminal nodes, respectively. Let us suppose that c_1 is the single start node and c_n is the single terminal node. Let us suppose that each arc is alive with probability p independently of each other. The probability of the event that one can get from the start node c_1 to the terminal node c_n along living arcs is called the reliability of the network. The goal of the paper is to find bounds (both lower and upper) and approximations on the reliability of large sized networks.

One possible way to do this is to determine all paths leading from the start node c_1 to the terminal node c_n. Denote these paths by $P_1,\ldots,P_N$ and the event that all arcs along the path P_i are alive denote by $A_i, i=1,\ldots,N$. With these notations the reliability of the network equals to the probability value:

$$P(A_1\cup\cdots\cup A_N).$$

As the probability of the event that all arcs are living along a few paths (say not more than five of them) can be calculated quickly even if there exist a number of paths leading from the start node to the terminal node we will investigate procedures bounding or approximating the probability of union of events using these probabilities only. In Section II we summarize those results which bound or approximate the probability of union of events using only the probabilities of single events and of intersections of two different events. In Section III we summarize higher order bounding and approximating techniques for the probability of the union of events. These procedures use some intersection probabilities according to more than two but not too many (say less than five, depending on the number of the existing paths in the reliability network) events.

II. PROBABILITY BOUNDS AND APPROXIMATIONS BASED ON MAXIMUM WEIGHT SPANNING TREES

Graph theory has plenty applications in the calculation of bounds and approximations of the probability of the union of events.

In 1968 Chow and Liu were presented a method for the approximation of a discrete multivariate joint probability distribution, see [1]. They defined the so called dependence tree. This is a spanning tree of a complete, weighted graph, where the nodes are random variables and each link between two variables is weighted by the information content (also called mutual information) of the pair of random variables. They determined the heaviest spanning tree via Prim or Kruskal algorithm. To this spanning tree a fraction-type probability distribution was assigned. The nominator contains the product of the two dimensional marginals of the random variables connected, and in the denominator the product of the univariate marginals at the power of the node-degree (the number of edges that contain the given node) minus 1.

Hunter ([2]), Worsley ([3]), and later Block, Costigan and Sampson ([4]) use weighted graph-structures. The nodes are events, the edges are weighted by the probability of the occurence of the connected events. Using the heaviest spanning tree Hunter and Worsley gave upper bound for the probability of the union of events.

Block, Costigan and Sampson gave a probability upper bound using a product type probability earned from the heaviest spanning tree. Their method is applicable in special cases, where the distribution fullfils the MTP2 property. All these bounding and approximating procedures use only the probabilities of single events and probabilities of intersections of some pairs of the events.

T. Szántai is with Institute of Mathematics, Budapest University of Technology and Economics, 1111 Műegyetem rkp. 3, Budapest, Hungary szantai@math.bme.hu

J. Bukszár is with the Center for Biomarker Research and Personalized Medicine, School of Pharmacy, Virginia Commonwealth University, P.O. Box 980126, Richmond, VA 23298-0126, USA jbukszar@vcu.edu

E. Kovács is with the Department of Methodology, ÁVF College of Management of Budapest, 1117 Villányi út 13-17, Budapest, Hungary kovacs.edith@avf.hu

ISBN 978-963-311-370-7

III. HIGHER ORDER PROBABILITY APPROXIMATIONS

This section contains two main streams. One is dedicated to the calculation of lower and upper bounds for the reliability of a network, the other gives an approximation of the realiability of the network. The latter gives the approximation of realiability as a realisation of a special kind of discrete probability distribution.

In [5] and [6] J. Bukszár together with A. Prekopa and T. Szántai introduced the concepts of cherry-tree, t-cherrytree, multitree, and hypermultitree. These are special type of graph and hypergraph structures useful for giving lower and upper bounds for the probability of the union of events.

Let $A_1,\dots,A_n$ be arbitrary events in a probability space (Ω,A,P). The aim is finding an upper bound for $P(A_1\cup\ldots\cup A_n)$ based on some of the probabilities $P(A_{k_1}\cap\ldots\cap A_{k_i})$, where $1\le k_1\le\ldots\le k_i\le n$, $i=1,\dots,d$. The bounds of this type are called d-th order upper bounds. For $m=d-1$, in [7] a special hypergraph structure with n vertices called m-multitree was introduced. To each m-multitree an upper bound can be assigned. (For $m=1$ one obtains the Hunter–Worsley bound, for $m=2$ one obtains the cherry tree bound.)

Definition 2.1 ([8]) Let m be a positive integer. An *m-multicherry* is a hypergraph of the form $(V,\varepsilon_2,\dots,\varepsilon_{m+1})$, where $V=\{v_1,\dots,v_{m+1}\}$ is the set of vertices and for each $i=2,\dots,m+1$ the family of hyperedges ε_i is the set of all subsets of ε_i^{m+1} is the set of all subsets of $\{v_1,\dots,v_{m+1}\}$ containing i vertices with v_{m+1} included, i.e., $\varepsilon_i=\{H|v_{m+1}\in H\subset\{v_1,\dots,v_{m+1}\},|H|=i\}$. The vertex v_{m+1} is called *the dominating vertex* of the m-multicherry. The m-multicherry with dominating vertex v_{m+1} and with non-dominating vertices $v_1,\dots,v_m$ is denoted by $(\{v_1,\dots,v_m\},v_{m+1})$.

Definition 2.2 ([8]) Let m be a positive integer. An *m-multitree* is a hypergraph of the form $(V,\varepsilon_2,\dots,\varepsilon_{m+1})$, where V is the set of vertices and ε_i's are sets of hyperedges containing i vertices. An m-multitree is recursively defined by the following two rules.

(i) The smallest m-multitree $\Delta=(V,\varepsilon_2,\dots,\varepsilon_{m+1})$ has m vertices and ε_i is the family of all subsets of V containing i vertices (here $\varepsilon_{m+1}=\emptyset$).

(ii) From an m-multitree $\Delta=(V,\varepsilon_2,\dots,\varepsilon_{m+1})$ we can obtain a new m-multitree $\Delta'=\left(V',\varepsilon_2',\dots,\varepsilon_{m+1}'\right)$ by adjoining an m-multicherry $(\{v_1,\dots,v_m\},v_{m+1})$, where $v_1,\dots,v_m\in V$ and v_{m+1} is a new vertex (i.e., $v_{m+1}\notin V$). More precisely $V'=V\cup\{v_{m+1}\}$, $\varepsilon_i'=\varepsilon_i\cup\{H|v_{m+1}\in H\subset\{v_1,\dots,v_{m+1}\},|H|=i\}$.

Let $\{A_1,\dots,A_n\}$ be a set of n events and let $V=\{1,\dots,n\}$ the set of indices.

Definition 2.3 ([8]) The weight of an m-multitree according to the set $\{A_1,\dots,A_n\}$ of events is given by the following formula:

$$\begin{aligned} w(\Delta^m) &= \sum_{\{l_1,l_2\}\in\varepsilon_2} P\left(A_{l_1}\cap A_{l_2}\right)\\ &- \sum_{\{l_1,l_2,l_3\}\in\varepsilon_3} P\left(A_{l_1}\cap A_{l_2}\cap A_{l_3}\right)\\ &+ \cdots\\ &+ (-1)^{m+1}\sum_{\{l_1,\dots,l_{m+1}\}\in\varepsilon_{m+1}} P\left(A_{l_1}\cap\ldots\cap A_{l_{m+1}}\right). \end{aligned}$$

Theorem 2.1 If $w(\Delta)$ is the weight of an arbitrary m-multitree according to the set of events $\{A_1,\dots,A_n\}$ then

$$P(A_1\cup\ldots\cup A_n)\le\sum_{i=1}^{n}P(A_i)-w(\Delta^m).$$

It is obvious from the inequality of the theorem, that it is important to achieve the heaviest m-multitree possible.

In [7] an algorithm is given for finding heavy m-multitrees. The m-multitree bound the algorithm provides us may not be the heaviest one, however, by extending of an r-multitree to an $(r+1)$-multitree the algorithm improves on the.

Later in [8] J. Bukszár has introduced the hypermultitree in order to give lower and upper bounds for the probability of the union of n events. These bounds are a generalization of Tomescu's lower and upper bounds ([12]).

The definition of the (h,m)-hipermultitree uses the definition of m-multitree, and is given recursively.

Definition 2.4 ([8]) Let $h\ge 0$ and $m\ge 1$ be arbitrary integers. An *(h,m)-hypermultitree* is a hypergraph of the form $(V,{}_h\varepsilon_2,\dots,{}_h\varepsilon_{m+1})$, where V is the set of vertices and ${}_h\varepsilon_i$'s are sets of hyperedges containing $h+i$ vertices. An (h,m)-hypermultitree is defined recursively by the following rules.

(i) The $(0,m)$-hypermultitrees are the same as the m-multitrees.

(ii) The smallest (h,m)-hypermultitree $\Delta=(V,{}_h\varepsilon_2,\dots,{}_h\varepsilon_{m+1})$ has $h+m$ vertices and ${}_h\varepsilon_i$ consists of all subsets of V containing $h+i$ vertices (here ${}_h\varepsilon_{m+1}=\emptyset$.

(iii) From an (h,m)-hypermultitree $\Delta=(V,{}_h\varepsilon_2,\dots,{}_h\varepsilon_{m+1})$ we can obtain a new (h,m)-hypermultitree in the following manner. Let $\Gamma=\left(V,{}_{h-1}\varepsilon_2^*,\dots,{}_{h-1}\varepsilon_{m+1}^*\right)$ be an arbitrary $(h-1,m$-hypermultitree with the same set of vertices as in Δ. By joining a new vertex v to Δ and the hyperedges of Γ extended by v, we obtain the new (h,m)-hypermultitree $\Delta'=\left(V,{}_h\varepsilon_2',\dots,{}_h\varepsilon_{m+1}'\right)$, i.e.,

$$V'=V\cup\{v\}\qquad {}_h\varepsilon_i'={}_h\varepsilon_i\cup\{E\cup\{v\}|E\in{}_{h-1}\varepsilon_i^*\}$$

The (h,m)-hypermultitrees are generalizations of Tomescu's hypertrees, which are the $(h,1)$-hypermultitrees in our definition.

Definition 2.5 [8] The weight of the $\Delta^{h,m} = (V,_h\varepsilon_2,\ldots,_h\varepsilon_{m+1})$,$(h,m)$-hypermultitree, according to the set of events $\{A_1,\ldots,A_n\}$ is

$$\begin{aligned} w\left(\Delta^{h,m}\right) &= \sum_{\{i_1,\ldots,i_{h+2}\}\in_h\varepsilon_2} P\left(A_{i_1}\cap\ldots\cap A_{i_{h+2}}\right) \\ &- \sum_{\{i_1,\ldots,i_{h+3}\}\in_h\varepsilon_3} P\left(A_{i_1}\cap\ldots\cap A_{i_{h+3}}\right) \\ &+ \ldots \\ &+ (-1)^{m+1} \sum_{\{i_1,\ldots,i_{m+1}\}\in_h\varepsilon_{m+1}} P\left(A_{i_1}\cap\ldots\cap A_{i_{h+m+1}}\right) \end{aligned}$$

Theorem 2.2 [8] If $\Delta^{h,m} = (V,_h\varepsilon_2,\ldots,_h\varepsilon_{m+1})$ is an arbitrary (h,m)-hypermultitree, according to the set of events $\{A_1,\ldots,A_n\}$ then the following inequalities hold:

- if h is even, then:

$$P\left(\bigcup_{i=1}^{n} A_i\right) \leq \sum_{k=1}^{h+1}(-1)^{k-1}S_k - w\left(\Delta^{h,m}\right);$$

- if h is odd, then:

$$P\left(\bigcup_{i=1}^{n} A_i\right) \geq \sum_{k=1}^{h+1}(-1)^{k-1}S_k + w\left(\Delta^{h,m}\right),$$

where $S_k = \sum_{1\leq i_1\leq\ldots\leq i_k\leq n} P\left(A_{i_1}\cap\ldots\cap A_{i_k}\right)$.

Remark 2.1 For the special case $m = 1$ we obtain the Tomescu bounds, for $h = 0$ and $m = 1$ we obtain Hunter's and Worsley's bound.

In [8] Bukszár gives an algorithm that uses Theorem 2.2, in order to achieve lower and upper bounds for the union of events. The algorithm that finds a heavy $(1,m)$-hypermultitree consists of two phases. In the first phase a heavy $(1,1)$-hypermultitree is constructed and in the second phase $(1,r)$-hypermultitree is extended to $(1,r+1)$-hypermultitree recursively $(r = 1,\ldots,m-1)$ in order to obtain better bounds.

Now we give an approximation for the union of events, using first a method for approximation of a discrete joint probability distribution. The approximation which uses the two dimensional marginals only was introduced by Chow and Liu in [1], then the method which uses the 3-, 4-, 5-order marginals was recently introduced in [9] and [10].

In order to give an approximation for $P(A_1\cup\ldots\cup A_n)$,we need to define the following characteristic variables:

$$X_i = \begin{cases} 1 \text{ if } A_i \text{takes place} \\ 0 \text{ if } \overline{A_i} \text{takes place} \end{cases}.$$

We denote the one-dimensional marginals of the joint probability distribution $P(\mathbf{X}) = P(X_1,\ldots,X_n)$ by $P(X_i)$, $i = 1,\ldots,n$ and the set of the indices by $V = \{1,\ldots,n\}$.

The problem is now finding a good approximation $P_{app}(\mathbf{X})$ for $P(\mathbf{X})$, which uses only low dimensional marginals and then to express the probability of the realization of $(X_1 = 1,\ldots,X_n = 1)$ from it . This requires the calculation of $P_{app}\left(\bigcap_{i=1}^{n}(X_i = 1)\right)$. Using this $P_{app}(A_1\cup\ldots\cup A_n)$ can be easily calculated.

For higher order marginals corresponding to a set of indices $A\subset V$ we use the popular notation $P(\mathbf{X}_A)$.

Our approach needs the definition of junction tree, t-cherry tree, and t-cherry junction tree, and for the optimalization of fitting the Kullback-Leibler divergence as loss function.

Let $X = \{X_i\}_{i=1,\ldots,n}$ be a set of discrete finite random variables over the same $(\Omega,\mathscr{A},P)$.

Throughout this part we will use the following popular notation:

$$\sum_{\mathbf{x}} P(\mathbf{X}) = \sum_{i_1=1}^{m_1}\ldots\sum_{i_n=1}^{m_n} P\left(X_1 = x_1^{i_1},\ldots,X_n = x_n^{i_n}\right).$$

where $x_k^{i_k}$, $i_k = 1,\ldots,m_k$ are the possible values of the random variable X_k, $k = 1,\ldots,n$. Apply similar notation for products, too.

The junction tree, also called Markov tree is a special tree with nodes containing random variables.

A tree which fulfills the following properties is called junction tree over **X.**

Definition 2.6 ([10])

1) Each node of the tree consists of a subset of X, called *cluster*. To each cluster we assign the joint marginal probability distribution of its random variables.
2) Each edge connecting two clusters of the tree consists of a subset of X given by the intersection of the connected clusters, called *separator*. To each separator we assign the joint marginal probability distribution of its random variables.
3) If two clusters contain a random variable, then all clusters on the path between these two clusters contain this random variable (running intersection property).
4) The union of all clusters is X.

The junction tree provides a joint probability distribution of $\mathbf{X}$:

$$P(\mathbf{X}) = \frac{\prod_{\mathbf{X}_C\in\mathscr{C}} P(\mathbf{X}_C)}{\prod_{\mathbf{X}_S\in\mathscr{S}} P(\mathbf{X}_S)^{\nu_s-1}}, \qquad (1)$$

where $\mathscr{C}$ is the set of clusters and $\mathscr{S}$ is the set of separators, $P(\mathbf{X}_C)$ and $P(\mathbf{X}_S)$ are the marginal probability distribution associated to the random vector, with variables from the set C and S, respectively ν_s is the number of those clusters which contain all of the variables involved in S.

Theorem 2.3 [10] The Kullback- Leibler divergence between the approximating probability distribution associated to a junction tree structure given by formula (1) and the real distribution $P(\mathbf{X})$ is given by:

$$KL\left(P_{app}(\mathbf{X}), P(\mathbf{X})\right) = -H(\mathbf{X}) - \left[\sum_{\mathbf{X}_C \in \mathscr{C}} I(\mathbf{X}_C) - \sum_{\mathbf{X}_S \in \mathscr{S}} (\nu_S - 1) I(\mathbf{X}_S)\right] + \sum_{i=1}^{n} H(X_i), \tag{2}$$

where $H(\mathbf{X})$ denotes the entropy of the random vector $\mathbf{X}$ and $I(\mathbf{X}_C)$ $(I(\mathbf{X}_S))$ denotes the information content (see [11]) of the components of the random vector $\mathbf{X_C}$ $(\mathbf{X_S})$.

In the formula (2) the first and the last term do not depend on the junction tree structure. The difference between the sums of information contents depends only on the structure of the junction tree. Therefore it is worthy to introduce the following definition:

Definition 2.7 ([10]) The difference

$$\sum_{\mathbf{X}_C \in \mathscr{C}} I(\mathbf{X}_C) - \sum_{\mathbf{X}_S \in \mathscr{S}} (\nu_S - 1) I(\mathbf{X}_S)$$

is called the weight of the junction tree.

The heavier the weight of the junction tree the better the approximation.

The problem is to define a junction tree family which makes it possible to search for a good approximation. For this purpose we use the t-cherrytree graph structure [5] over the set V of indices, we then associate the so called k-th order t-cherry junction tree with.

The recursive construction of the k-th order t-cherry tree:

Definition 2.8 ([10])

(i) The complete graph of (k-1) nodes from V represent the smallest k-th order t-cherry tree.

(ii) By connecting a new vertex from V, with all vertices of $(k-1)$ dimensional complete subgraph of the existing k-th order t-cherry tree, we obtain a new k-th order t-cherry tree.

(iii) Each k-th order t-cherry tree can be obtained from (i) by successive application of (ii).

Definition 2.9 ([10]) The set of vertices of the $(k-1)$ dimensional complete subgraph used in step (ii) of Definition 2.8 is called hyper edge of the k-th order t-cherry tree.

Definition 2.10 ([10]) The set of vertices of a hyper edge together with a new vertex is called hypercherry of the k-th order t-cherry tree.

The set of hyper edges of the k-th order t-cherry tree let be denoted by ε_{k-1}.

The set of hyper cherries of the k-th order t-cherry tree let be denoted by $\mathscr{C}_k$.

The set of vertices V, the set of hyper edges $\varepsilon_{k-1}$and the set of hypercherries $\mathscr{C}_k$ define the $\Delta_k = (V, \varepsilon_{k-1}, \mathscr{C}_k)$ k-th order t-cherry tree.

Definition 2.11 ([10]) The *k-th order t-cherry junction tree* is defined in the following way.

1) By using Definition 2.8 we construct a k-th order t-cherry tree over $V:\ \Delta_k = (V, \varepsilon_{k-1}, \mathscr{C}_k)$.
2) To each hyper cherry $(\{i_1, \ldots, i_{k-1}\}, i_k)$ we order a cluster set containing the variables $\left\{X_{i_1}, \ldots, X_{i_{k-1}}, X_{i_k}\right\}$.
3) To each hyper edge $\{i_1, \ldots, i_{k-1}\}$ we assign a separator set containing the variables $\{X_{i_1}, \ldots,\ X_{i_{k-1}}\}$ (edge of the junction tree).

The junction tree provides a joint probability distribution of $\mathbf{X}$:

$$P_k(\mathbf{X}) = \frac{\prod\limits_{\left(X_{i_1}, \ldots. X_{i_k}\right) \in \mathscr{C}} P\left(X_{i_1}, \ldots. X_{i_k}\right)}{\prod\limits_{\left(X_{j_1}, \ldots, X_{j_{k-1}}\right) \in \mathscr{S}} P\left(X_{j_1}, \ldots, X_{j_{k-1}}\right)^{\nu_{j_1, \ldots, j_{k-1}} - 1}}, \tag{3}$$

where $\mathscr{C}$ is the set of clusters and $\mathscr{S}$ is the set of separators, $P\left(X_{i_1}, \ldots. X_{i_k}\right)$ and $P\left(X_{j_1}, \ldots, X_{j_{k-1}}\right)$ are the marginal probability distributions. $\nu_{j_1, \ldots, j_{k-1}}$ is the number of those clusters which contain all of the variables $X_{j_1}, \ldots,\ X_{j_{k-1}}$.

We emphasize that for $k = 2$ we obtain the so called Chow–Liu dependence tree [1]. The best fitting Chow–Liu dependence tree can be found by Prim's or Kruskal's algorithm.

In [9] we use a greedy algorithm for for searching for a heavy t-cherry juntion tree, and in [10] we give a theorem and an algorithm which makes possible to improve the approximation given by a k-th order t-cherry junction tree to a $(k+1)$-th order t-cherry juntion tree.

For finding a good approximation searching through the family of all k-th order junction trees, we need to maximize the weight of this special junction tree. For the calculation of the information gains we need the probabilities of all possible realizations of each of the k-th order marginals. Under the assumption that the random variables $(X_i)_{i=1\ldots n}$ are the characteristic variables, each of the k-th order marginals have 2^k realizations. In practice we use k less then or equal 5. For finding these probabilities we need to know all probabilities of type $P\left(A_{i_1} \cap \ldots \cap A_{i_k}\right)$, where $\{i_1, \ldots, i_k\}$ are all k-element subsets of V. It is obvious that any $P\left(A_{j_1} \cap \ldots \cap A_{j_l}\right)$, $l < k$ can be calculated by a simple summation.

The probabilities of type $P\left(\overline{A}_i \cap \ldots \cap \overline{A}_l \cap A_{l+1} \cap \ldots \cap A_m\right)$ where $m \leq k$, $l = 1, \ldots, m-1$ can be expressed by the following recursion:

For $m = 2, \ldots, k$

1. For $l = 1$:

$$\begin{aligned} P\left(\overline{A}_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_m}\right) &= P\left(A_{i_2} \cap \ldots \cap A_{i_m}\right) \\ &- P\left(A_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_m}\right); \end{aligned}$$

2. For $l = 2, \ldots, m$:

$$\begin{aligned} &P\left(\overline{A}_{i_1} \cap \ldots \cap \overline{A}_{i_{l-1}} \cap \overline{A}_{i_l} \cap A_{i_{l+1}} \cap \ldots \cap A_{i_m}\right) \\ = \; &P\left(\overline{A}_{i_1} \cap \ldots \cap \overline{A}_{i_{l-1}} \cap A_{i_{l+1}} \cap \ldots \cap A_{i_m}\right) \\ - \; &P\left(\overline{A}_{i_1} \cap \ldots \cap \overline{A}_{i_{l-1}} \cap A_{i_l} \cap A_{i_{l+1}} \cap \ldots \cap A_{i_m}\right). \end{aligned}$$

IV. APPLICATIONS AND NUMERICAL EXAMPLES

A. *Applications*

The paper ”Predicting Protein Complex Membership Using Probabilistic Network Reliability” [13] deals with the problem of identifying to which protein complex, with a known core set of proteins, a given candidate protein belongs. The authors formulate their probabilistic method in the following way:

”The probability that a given candidate protein is in the same protein complex with a known core set of proteins may be expressed as the probability that there exists a path of direct and stable protein interactions between that candidate and some member of the complex. How can we estimate this probabilty? This problem is analogous to one previously considered problem in theory of communication networks: the two terminal network reliability problem”.

Their method ranks protein candidates according to the probability of their connection to the core complex via any path. They provide a weighted graph model of the protein interaction network, in which nodes represent proteins and the edges are weighted with the probability that the linked proteins interact stably and directly [14]. The bounding and approximating methodology presented in this paper can be applied to this problem.

B. *Numerical examples*

All bounding and approximating techniques presented in this paper will be illustrated on the reliability of a network. For very small, toy networks we can calculate the exact reliability of the network and compare the approximations and bounds to these exact values. For large networks first we determine the most reliable paths from the source node to the terminal node of the network. Then if the number of these paths is too large we can make a preliminary Monte Carlo simulation according to these paths to choose those which are permeable for high percentage of the simulation runs. If at least one of these paths is permeable we accept that the network itself is permeable. The reliability of the network will be estimated by the probability that the network is permeable.

Consider the randomly generated network of Figure 1 and suppose that each arc is permeable with probability p independently of each other.

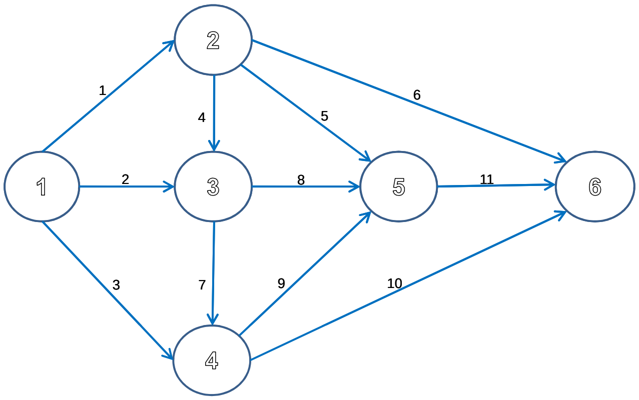

Fig. 1. Randomly generated network with 6 nodes and 11 arcs

In the network of Figure 1 there exist 10 paths leading from node 1 to node 6. The path–arc incidence matrix of these paths looks like:

	Arcs										
	1	2	3	4	5	6	7	8	9	10	11
Path 1	1	0	0	0	0	1	0	0	0	0	0
Path 2	0	1	0	0	0	0	0	1	0	0	1
Path 3	1	0	0	1	0	0	0	1	0	0	1
Path 4	0	1	0	0	0	0	1	0	0	1	0
Path 5	1	0	0	1	0	0	1	0	0	1	0
Path 6	0	1	0	0	0	0	1	0	1	0	1
Path 7	1	0	0	1	0	0	1	0	1	0	1
Path 8	1	0	0	0	1	0	0	0	0	0	1
Path 9	0	0	1	0	0	0	0	0	0	1	0
Path 10	0	0	1	0	0	0	0	0	1	0	1

If A_i denotes the event that the i-th path is permeable, $i = 1, \ldots, 10$ then we can calculate all of the 1023 probability terms involved in the Poincare formula and get the following expression for the union of events $A_1, \ldots, A_{10}$:

$$\begin{aligned} P(A_1 \cup \cdots \cup A_{10}) &= 2p^2 + 4p^3 - p^4 - 17p^5 + 4p^6 + 40p^7 \\ &- 55p^8 + 32p^9 - 9p^{10} + p^{11}. \end{aligned}$$

Figures 2 – 5. show to what extent the bounds and approximations are better when we use higher order bounds and approximations. All diagrams on the vertical axis give the reliability of the network given in Figure 1. The horizontal axis corresponds to the common p value expressing the probability that any arc of the network is alive independently of all others.

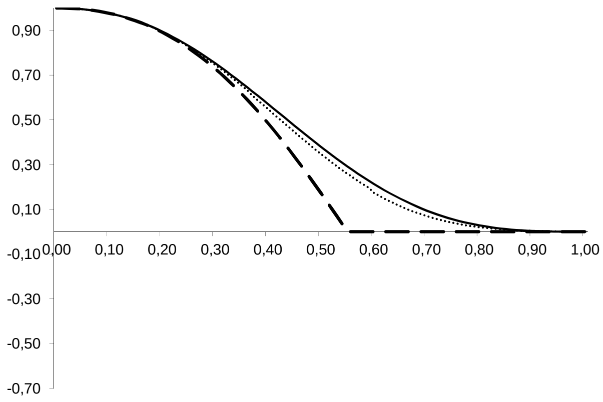

Fig. 2. Second order lower bound (Hunter and Worsley): dashed line
Second order approximation (Kovács and Szántai): dotted line
True probability value: continuous line

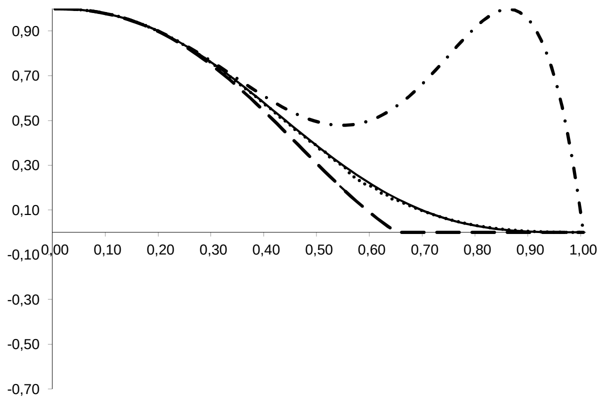

Fig. 3. Third order (2-multitree) lower bound: dashed line
Third order approximation (Kovács and Szántai): dotted line
True probability value: continuous line
Third order ((1,1)-hypermultitree) upper bound: dot-and-dash line

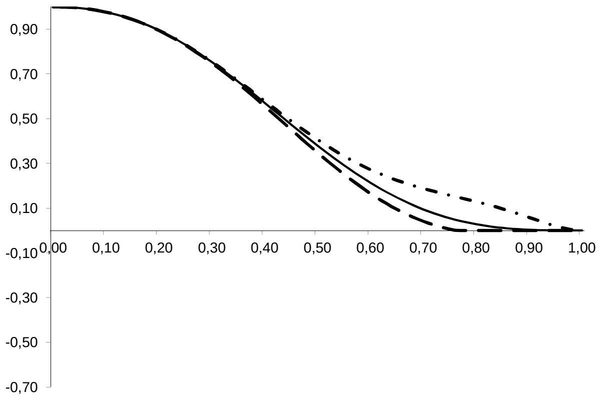

Fig. 4. Fourth order (3-multitree) lower bound: dashed line
True probability value: continuous line
Fourth order ((1,2)-hypermultitree) upper bound: dot-and-dash line

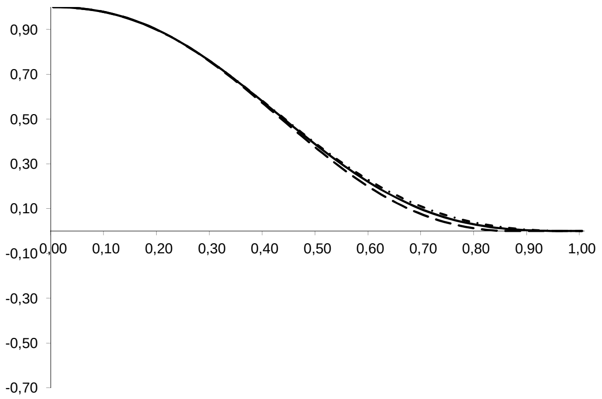

Fig. 5. Fifth order (4-multitree) lower bound: dashed line
True probability value: continuous line
Fifth order ((1,3)-hypermultitree) upper bound: dot-and-dash line

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

We can conclude that the hypermultitree and multitree bounds get tighter as we increase their order. For the randomly generated moderate sized network it was sufficient to increase the order of the bounds up to five to obtain tight bounds.

In the paper we developed a new approximation technique for the probability of the union of events. This technique is based on the approximation of discrete joint probability distributions by the Chow-Liu's spanning tree and for higher order approximations by the Kovács-Szántai's t-cherry junction tree. The advantage of the new approximation is that on the randomly generated network we obtained good second order approximation and very good third order approximation.

B. Future Works

For large networks one can determine the most reliable paths from the source node to the terminal node in the following way. To each arc of the network we assign the minus logarithm of the probability value that the arc is living. Then we find the k shortest paths in the network. There are effective algorithms for this purpose, for example see [15]. Then if the number of these paths is too large, say more than one hundred, we perform a Monte Carlo simulation according to these paths to choose those of them which are permeable in high percentage of the simulation runs. Leave only these paths and if at least one of these paths is permeable accept that the network itself is permeable. We hope that for relatively large sized networks this way one can get useful approximations and bounds on the reliability of the network.

References

[1] C.K. Chow and C.N. Liu, Approximating Discrete Probability Distribution with Dependence Tree, *IEEE Transactions on Informational Theory*, vol. 14, 1968, pp 462-467.

[2] D. Hunter, Bounds for the Probability of a Union, *Journal of Applied Probability*, vol. 13, 1976, pp 597-603.

[3] K. J. Worsley, An Improved Bonferroni Inequality and Applications, *Biometrika*, vol. 69, 1982, pp 297-302.

[4] H. W. Block, T. Costigan and A.R. Sampson, Second Order Bonferroni-type, Product-type and Setwise Probability Inequalities, *Stochastic Orders and Decision under Risk*, IMS Lecture Notes – Monograph Series, 1991, pp 74-94.

[5] J. Bukszár and A. Prékopa, Probability Bounds with Cherry Trees, *Mathematics of Operational Research*, vol. 26, 2001, pp 174-192.

[6] J. Bukszár and T. Szántai, Probability Bounds Given by Hypercherry Trees, *Optimization Methods and Software*, vol. 17, 2002, pp 409-422.

[7] J. Bukszár, Upper Bounds for the Probability of a Union by Multitrees, *Advances in Applied Probability*, vol. 33, 2001, pp 437-452.

[8] J. Bukszár, Hypermultitrees and Sharp Bonferroni Inequalities, *Mathematical Inequalities & Applications*, vol. 6, no. 4, 2003, pp 727-743.

[9] E. Kovács and T. Szántai, On the Approximation of Discrete Multivariate Probability Distribution using the new concept of t-cherry junction tree, in: Proceedings of the IFIP/IIASA/GAMM Workshop on Copyng with Uncertainty, IIASA, Laxenburg, 2007, accepted.

[10] T. Szántai and E. Kovács, Hypergraphs as Means of Discovering the Dependence Structure of a Discrete Multivariate Probability Distribution, *Annals of Operations Research*, 2009, accepted.

[11] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, Wiley Interscience, New York; 1991.

[12] I. Tomescu, Hypertrees and Bonferroni inequalities, *Journal of Combinatorial Theory*, vol. 41, 1986, pp 209-217.

[13] S. Asthana, O.D. King,F.D. Gibbons and F.P. Roth, Predicting Protein Complex Membership Using Probabilistic Network Reliability, *Genom. Res.*, vol. 14, 2004, pp 1170-1175.

[14] A.M. Edwards, B. Kus, R. Jansen, D. Greenbaum, J. Greenblatt and M. Gerstein, Bridging Structural Biology and Genomics: Assessing Protein Interaction Data with Known Complexes, *Trends. Genet.*, vol. 18, 2002, pp 529-536.

[15] D. Eppstein, Finding the k Shortest Paths, *SIAM Journal on Computation*, vol. 28, 1999, 652-673.