EXTREMAL PROBABILITY BOUNDS IN COMBINATORIAL OPTIMIZATION*

DIVYA PADMANABHAN†, SELIN DAMLA AHIPASAOGLU‡, ARJUN RAMACHANDRA§, AND KARTHIK NATARAJAN¶

Abstract. In this paper, we compute the tightest possible bounds on the probability that the optimal value of a combinatorial optimization problem in maximization form with a random objective exceeds a given number, assuming only knowledge of the marginal distributions of the objective coefficient vector. The bounds are "extremal" since they are valid across all joint distributions with the given marginals. We analyze the complexity of computing the bounds assuming discrete marginals and identify instances when the bounds are computable in polynomial time. For compact 0/1 V-polytopes, we show that the tightest upper bound is weakly NP-hard to compute by providing a pseudopolynomial time algorithm. On the other hand, the tightest lower bound is shown to be strongly NP-hard to compute for compact 0/1 V-polytopes by restricting attention to Bernoulli random variables. For compact 0/1 H-polytopes, for the special case of PERT networks arising in project management, we show that the tightest upper bound is weakly NP-hard to compute by providing a pseudopolynomial time algorithm. The results in the paper complement existing results in the literature for computing the probability with independent random variables.

Key words. Probability Bounds, Combinatorial Optimization, PERT

AMS subject classifications. 90-08 , 90C05, 90C27

1. Introduction.

In this paper, we are interested in the random combinatorial optimization problem of the form:

$$Z(\tilde{\mathbf{c}}) = \max \quad \tilde{\mathbf{c}}'\mathbf{x} \qquad \text{s.t.} \quad \mathbf{x} \in \mathcal{X} \subseteq \{0,1\}^n, \tag{1.1}$$

where $\tilde{\mathbf{c}} = (\tilde{c}_1, \ldots, \tilde{c}_n)$ is an n-dimensional random vector and $\mathcal{X}$ is a subset of the set $\{0,1\}^n$. Our main goal is to compute bounds on the probability that the random optimal value $Z(\tilde{\mathbf{c}})$ is greater than or equal to a fixed number r when the marginal distributions of the random variables $\tilde{c}_i$ for $i \in [n]$ are specified. Throughout the paper, given a nonnegative integer n, we let $[n]$ denote the set $\{1,\ldots,n\}$ and given integers $n_1 \leq n_2$, we let $[n_1, n_2]$ denote the set $\{n_1, \ldots, n_2\}$. We assume each random variable $\tilde{c}_i$ is discrete with marginal probabilities specified as $\mathbb{P}(\tilde{c}_i = c_{ik}) = p_{ik}$ for $k \in [0,K]$ and support given by $\mathcal{C}_i = \{c_{i0}, \ldots, c_{iK}\}$ where the values are ordered as $c_{i0} < \ldots < c_{iK}$. The marginal probabilities satisfy $\sum_k p_{ik} = 1$ for all $i \in [n]$ and $p_{ik} \geq 0$ for all $i \in [n]$ and $k \in [0,K]$. Let Θ denote the set of all joint distributions on $\tilde{\mathbf{c}}$ consistent with the marginal distributions:

$$\Theta = \left\{ \theta \in \mathbb{P}(\prod_{i=1}^{n} \mathcal{C}_i) : \mathbb{P}_\theta(\tilde{c}_i = c_{ik}) = p_{ik}, \text{ for } i \in [n], k \in [0,K] \right\},$$

*Submitted: August 2021, Revised: April 2022

†School of Mathematics and Computer Science, Indian Institute of Technology, Ponda-403401, Goa. Email: divya@iitgoa.ac.in

‡Mathematical Sciences, University of Southampton, Highfield Southampton SO17 1BJ. Email: sda1u20@soton.ac.uk

§Engineering Systems and Design, Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372. Email: arjun_ramachandra@sutd.edu.sg

¶Engineering Systems and Design, Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372. Email: karthik_natarajan@sutd.edu.sg

where $\mathbb{P}(\prod_{i=1}^n \mathcal{C}_i)$ is the set of all joint distributions supported on the set $\mathcal{C}_1 \times \ldots \times \mathcal{C}_n$. Given a fixed value r, we are interested in computing the following extremal probability bounds:

$$\text{(Upper bound) } U(r) = \max_{\theta \in \Theta} \mathbb{P}_\theta(Z(\tilde{\mathbf{c}}) \geq r),$$

$$\text{(Lower bound) } L(r) = \min_{\theta \in \Theta} \mathbb{P}_\theta(Z(\tilde{\mathbf{c}}) \geq r).$$

A related probability of interest to compute is when the random combinatorial optimization problem has mutually independent random variables in the objective coefficient vector. Specifically, let θ_{ind} be the joint distribution:

$$\mathbb{P}_{\theta_{ind}}(\tilde{c}_1 = c_{1k_1}, \ldots, \tilde{c}_n = c_{nk_n}) = p_{1k_1} \times \ldots \times p_{nk_n}, \text{ for } k_1 \in [0, K], \ldots, k_n \in [0, K],$$

where $\theta_{ind} \in \Theta$. The probability for the independent distribution is given as:

$$\text{(Independence) } I(r) = \mathbb{P}_{\theta_{ind}}(Z(\tilde{\mathbf{c}}) \geq r),$$

where $U(r) \geq I(r) \geq L(r)$. We discuss the complexity of computing $U(r), L(r)$ and $I(r)$ in this paper.

1.1. Applications. Our interest in studying these probability bounds are motivated from the applications discussed next.

(a) In simple settings, the extremal probability bounds discussed in this paper reduce to well known probability bounds. For example, consider computing an upper bound on the probability of occurrence of at least one of the n events $E_1, \ldots, E_n$. If only the probabilities of occurrence of each individual event is known, Boole's union bound given by $\min(\sum_i \mathbb{P}(E_i), 1)$ is tight. This bound arises as a special case of the framework above, by defining the Bernoulli random variables as $\tilde{c}_i = 1$ if E_i occurs and $\tilde{c}_i = 0$ otherwise, setting $Z(\tilde{\mathbf{c}}) = \sum_i \tilde{c}_i$ and $r = 1$. Bounds on the sum of random variables when only the marginal distributions are given has been extensively studied in the risk, insurance and finance settings; see Chapter 4 in [38].

(b) In the context of Program Evaluation and Review Technique (PERT) networks, the distribution of the completion time of a project needs to be estimated where the project is composed of several activities with random activity times [13]. Planning decisions are made taking into account the distribution of the project completion time. In this setting, $Z(\tilde{\mathbf{c}})$ is the optimal value of a longest path problem on a directed acyclic graph where the arc length vector $\tilde{\mathbf{c}}$ denotes the random activity duration vector. The probability of the completion time exceeding a deadline r is a relevant measure of the performance of the project (higher the probability, worse the performance). Much of the literature has looked at computing this probability under the assumption of independence or limited dependence among the activity durations [15, 12, 19, 4, 30]. However in PERT networks, there is evidence of significant dependence occurring among the activity durations when the resources are shared across activities or when adverse events affect all activities [35]. This motivates the interest in the computation of extremal probability bounds.

(c) In the context of reliability, the probability of a system being functional is characterized in terms of the probabilities of the subcomponents being operational. Extremal probability bounds then provide an estimate of the robustness of the system to dependence among the subcomponents; see the book of [26]. For example, the s-t reliability measure (probability that there exists at least one operational path from node s to node t in a graph) is computed by assuming each edge (i, j) on the graph is associated with a Bernoulli random variable $\tilde{c}_{ij}$ where $\tilde{c}_{ij} = 1$ if the arc is operational and 0 if it fails and formulating $Z(\tilde{\mathbf{c}})$ as a minimum s-t cut problem with $r = 1$.

1.2. Existing Results and Contributions of This Paper. Evaluating $Z(\mathbf{c})$ is already NP-hard for the class of deterministic combinatorial optimization problems. In this paper we focus on combinatorial optimization problems where the convex hull of the feasible region has a compact representation and $Z(\mathbf{c})$ is computable in polynomial time. Two representations we consider are described next:

(a) V-polytope: The convex hull of the set $\mathcal{X} \subseteq \{0,1\}^n$, denoted by $\text{conv}(\mathcal{X})$, is given by a convex combination of a set of P points:

$$\begin{aligned} \text{conv}(\mathcal{X}) &= \text{conv}\{\mathbf{x}^1, \ldots, \mathbf{x}^P\}, \\ &= \left\{ \sum_{j=1}^{P} \lambda_j \mathbf{x}^j : \sum_{j=1}^{P} \lambda_j = 1, \lambda_j \geq 0, \text{ for } j \in [P] \right\}, \end{aligned} \tag{1.2}$$

where $\mathbf{x}^1, \ldots, \mathbf{x}^P \in \{0,1\}^n$. In this representation, P is typically exponential in n and so (1.2) is only useful when P is allowed to be part of the input size specification. The size of the input instance for computing $U(r)$ or $L(r)$ in this case is given by:

$$\text{Size of input } = O(\max(K, P) n \max(\log_2 U_1, \log_2 U_2)),$$

where $K+1$ is an upper bound on the size of any marginal support, n is the number of random variables, P is the number of extreme points in the V-polytope, U_1 and U_2 are the maximum numerical values among the integers in the ratio representation of the rational numbers p_{ik} and c_{ik} across all i and k. The logarithmic dependence of the input size on the magnitude of the input probabilities and the support points arises since $O(\log_2 U)$ binary digits are needed to represent a positive integer U. For $P = 1$ and $\mathbf{x} = \mathbf{1}_n$ (the vector of all ones), we get $Z(\tilde{\mathbf{c}}) = \sum_i \tilde{c}_i$. Even for the sum of random variables, computing $U(r)$ and $L(r)$ have been shown to be NP-hard for two point marginal distributions [25] using a reduction from the partition problem. Computing $I(r)$ with two point marginal distributions has also shown to be #P-hard [22] using a reduction from the problem of counting the number of feasible solutions to a 0-1 knapsack problem. In special cases, the bounds are efficiently computable. These include the sum of $n = 2$ random variables [28, 39] where simple formulas exist for arbitrary distributions and for the sum of n random variables with $K = 1$ (Bernoulli random variables) [37]. Many other bounds, not necessarily tight have also been proposed in the literature (see Chapter 4 in [38] for several such bounds). We add to this stream of results by showing that for compact 0/1 V-polytopes, the upper bound $U(r)$ is in fact weakly NP-hard to compute by providing a pseudopolynomial time algorithm. Specifically, we show that when the random variables take values $c_{ik} = k$ for $k \in [0, K]$, it is possible to compute $U(r)$ by solving a linear program that is of polynomial size in K, n, P and $\log_2(U_1)$. The key aspect of this result is that dependence on the parameter U_2 is overcome. Furthermore for Bernoulli random variables, we provide further reduction in the polynomial size of the linear program for computing $U(r)$. On the other hand, we show the lower bound $L(r)$ is strongly NP-hard to compute. Specifically, we show that it is not possible to compute $L(r)$ in polynomial time in the input size even when the random variables are Bernoulli, unless P = NP. We also provide a #P-hardness result for independent Bernoulli random variables in this representation.

(b) H-polytope: The convex hull of the set $\mathcal{X} \subseteq \{0,1\}^n$ is given by:

$$\text{conv}(\mathcal{X}) = \{\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}, \tag{1.3}$$

where the matrix $\mathbf{A}$ is of size $m \times n$ and $\mathbf{b}$ is a vector of length m. In this representation, the size of the input instance for computing $U(r)$ or $L(r)$ is given by:

$$O(\max(K, m)n \max(\log_2 U_1, \log_2 U_2, \log_2 U_3)),$$

where in addition to the other parameters, U_3 is the maximum numerical value among the integers in the ratio representation of the rational numbers in the matrix $\mathbf{A}$ and vector $\mathbf{b}$. An example of a combinatorial optimization problem with a compact 0/1 H-polytope representation is a PERT network where computing $Z(\mathbf{c})$ is possible in polynomial time. In PERT networks, the extreme points are characterized by the s-t paths in the network which can be exponentially large. The V-polytope representation is not useful in this setting. However $Z(\mathbf{c})$ can be computed efficiently using a linear program which grows polynomially in the size of the network characterized by the number of nodes and edges in the graph, rather than the number of paths in the graph. Computing $I(r)$ is however known to be NP-hard for PERT networks even when the activity durations are Bernoulli random variables [19]. For certain classes of reliability problems, polynomial time computable bounds $U(r)$ and $L(r)$ have been proposed in the literature [48, 45]. However these formulations make use of the equivalence of separation and optimization [17] to prove polynomial time complexity bounds without providing compact formulations that are easy to implement in practice.
We add to the stream of results in H-polytopes by showing that that for PERT networks a polynomial sized linear program (LP) can be used to compute the tightest upper bound $U(r)$ when the activity durations are restricted to take values in $[0, K]$. In turn, this shows that for PERT networks, the upper bound $U(r)$ is weakly NP-hard. This provides the maximum (worst case) probability of the random project completion time exceeding a given deadline.

A related area of research is distributionally robust chance constraints [47, 21] wherein the constraints of an optimization problem are required to be satisfied with high probability. The difference of this line of research from our work is that we instead focus on computing the tail probabilities of the objective value of an uncertain optimization problem.

The structure of the paper is as follows. In Section 2 and Section 3 respectively, we provide results for the V-polytope and the H-polytope. Numerical results provided in Section 4 compare various probability bounds in random walks and PERT networks. We also show applications in models exhibiting limited dependence.

2. Bounds for the V-Polytope.

2.1. Upper Bound. We begin by developing a pseudopolynomial time algorithm for computing $U(r)$ for 0/1 V-polytopes. The bound is computed using a linear program. For the analysis, we assume that the support of each random variable $\tilde{c}_i$ is contained in $\mathcal{C}_i = [0, K]$. Under this restriction on support, we are looking for algorithms with running time polynomial in K, n, P and $\log_2(U_1)$ thereby dropping the explicit dependence on the size of the input required to represent the marginal support values c_{ik}. The support of the random vector is contained in $[0, K]^n$ which is of size $O(K^n)$. Let us first write an exponential sized LP to compute $U(r)$ (see [20]):

$$\begin{aligned} U(r) = \max \quad & \sum_{\mathbf{c}\in[0,K]^n} \theta(\mathbf{c})\mathbb{1}_{\{Z(\mathbf{c})\geq r\}} \\ \text{s.t.} \quad & \sum_{\mathbf{c}\in[0,K]^n} \theta(\mathbf{c}) = 1, \\ & \sum_{\mathbf{c}\in[0,K]^n : c_i = k} \theta(\mathbf{c}) = p_{ik}, \text{ for } i \in [n], k \in [0, K], \\ & \theta(\mathbf{c}) \geq 0, \text{ for } \mathbf{c} \in [0, K]^n, \end{aligned}$$

where $\mathbb{1}_{\{Z(\mathbf{c})\geq r\}} = 1$ if $Z(\mathbf{c}) \geq r$ and 0 otherwise and the decision variables are the joint probabilities $\theta(\mathbf{c}) = \mathbb{P}(\tilde{\mathbf{c}} = \mathbf{c})$ for $\mathbf{c} \in [0, K]^n$. The primal linear program has a polynomial number of constraints but an exponential number of variables. From strong duality, $U(r)$ is the optimal value of the corresponding dual linear program,

$$U(r) = \min \ \lambda + \sum_{i=1}^{n}\sum_{k=0}^{K} \alpha_{ik} p_{ik}$$

$$\text{s.t. } \lambda + \sum_{i=1}^{n}\sum_{k=0}^{K} \alpha_{ik} \mathbb{1}_{\{c_i=k\}} \geq 1, \text{ for } Z(\mathbf{c}) \geq r, \mathbf{c} \in [0, K]^n, \tag{2.1}$$

$$\lambda + \sum_{i=1}^{n}\sum_{k=0}^{K} \alpha_{ik} \mathbb{1}_{\{c_i=k\}} \geq 0, \text{ for } \mathbf{c} \in [0, K]^n, \tag{2.2}$$

where the decision variables are λ and α_{ik} for $i \in [n]$ and $k \in [0, K]$. The dual linear program has a polynomial number of variables but an exponential number of constraints. By the equivalence of separation and optimization [17], a polynomial time algorithm to solve the underlying separation problem for the dual linear program implies the existence of a polynomial time algorithm to compute $U(r)$. We now show that the separation problems corresponding to the constraints (2.1) and (2.2) can be solved efficiently and develop a compact linear program to compute $U(r)$. We highlight that the approach detailed next, gives the tightest possible upper bound given any input marginal distributions.

THEOREM 2.1. *Let $\mathcal{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^P\} \subseteq \{0, 1\}^n$. Given the marginal distributions of the random vector $\tilde{\mathbf{c}}$ as $\mathbb{P}(\tilde{c}_i = k) = p_{ik}$ for $k \in [0, K]$ and $i \in [n]$, the tightest upper bound is computable by solving the linear program:*

$$\begin{aligned}
U(r) = \max \quad & \sum_{\mathbf{x}\in\mathcal{X}} a_{\mathbf{x}} \\
s.t. \quad & \sum_{\mathbf{x}\in\mathcal{X}} a_{\mathbf{x}} + b = 1, \\
& h_{ik} + \sum_{\mathbf{x}\in\mathcal{X}} \left[\sum_{l=k}^{nK} \delta_{ikl\mathbf{x}} x_i + \sum_{l=0}^{nK} \delta_{ikl\mathbf{x}} (1 - x_i) \right] = p_{ik}, \ \textit{for } i \in [1, n], k \in [0, K], \\
& \sum_{k\in[0,K]} h_{ik} = b, \ \textit{for } i \in [1, n], \\
& \sum_{l\in[r,nK]} \tau_{l\mathbf{x}} = a_{\mathbf{x}}, \ \textit{for } \mathbf{x} \in \mathcal{X}, \\
& \tau_{lx} = \sum_{k\in[0,K]} \delta_{nkl\mathbf{x}} (1 - x_n) + \sum_{k=0}^{\min(l,K)} \delta_{nkl\mathbf{x}} x_n \ \textit{for } l \in [r, nK], \ \textit{for } \mathbf{x} \in \mathcal{X}, \\
& \sum_{k=0}^{\min(K,l)} \delta_{ikl\mathbf{x}} x_i + \sum_{k=0}^{K} \delta_{ikl\mathbf{x}} (1 - x_i) = \sum_{k=0}^{M} \delta_{i+1,k,l+k,\mathbf{x}} x_{i+1} + \sum_{k=0}^{\min(K,l)} \delta_{i+1,k,l,\mathbf{x}} (1 - x_{i+1}) \\
& \qquad \textit{for } i \in [2, n], \ \textit{for } l \in [0, nK], \ \textit{for } \mathbf{x} \in \mathcal{X}, \\
& \delta_{1,l,l,\mathbf{x}} x_i = \sum_{k=0}^{M} \delta_{2,k,l+k,\mathbf{x}} x_2 + \sum_{k=0}^{K} \delta_{2,k,l,\mathbf{x}} (1 - x_2) \ \textit{for } l \in [1, K], \ \textit{for } \mathbf{x} \in \mathcal{X}, \\
& \sum_{k\in[0,K]} \delta_{1,k,0,\mathbf{x}} (1 - x_1) + \delta_{1,0,0,\mathbf{x}} x_1 = \sum_{k=0}^{K} \delta_{2,k,k,\mathbf{x}} x_2 + \sum_{k=0}^{K} \delta_{2,k,0,\mathbf{x}} (1 - x_2) \ \textit{for } \mathbf{x} \in \mathcal{X}, \\
& \delta_{nkl\mathbf{x}} = 0 \ \textit{for } k \in [0, K], \ \textit{for } l \in [0, r - 1], \ \textit{for } \mathbf{x} \in \mathcal{X}, \\
& \delta_{1,k,l,\mathbf{x}} = 0 \ \textit{for } k \in [0, K], \ \textit{for } l \in [1, nK], \ \textit{for } \mathbf{x} \in \mathcal{X} \ \textit{with } x_1 = 0, \\
& \delta_{1,k,l,\mathbf{x}} = 0 \ \textit{for } k \in [0, K], \ \textit{for } l \in [0, nK] \setminus k, \ \textit{for } \mathbf{x} \in \mathcal{X} \ \textit{with } x_1 = 1, \\
& \boldsymbol{\delta}, \mathbf{h}, \mathbf{a}, b, \boldsymbol{\tau} \geq 0.
\end{aligned}$$

where $M = \min(nK - l, K)$. *Specifically the linear program is solvable in time polynomial in* K*,* n*,* P *and* $\log_2(U_1)$*.*

Proof. We derive the LP by reformulating constraints (2.1) and (2.2).
Step (1): Reformulating constraints (2.1): We can rewrite constraint (2.1) as: $\lambda + W(\boldsymbol{\alpha}) \geq 1$, where $W(\boldsymbol{\alpha})$ is the optimal value of the following 0-1 integer program,

$$\begin{aligned} W(\boldsymbol{\alpha}) = \min \quad & \sum_{i=1}^{n}\sum_{k=0}^{K} \alpha_{ik} y_{ik} \\ \text{s.t.} \quad & \max_{\mathbf{x}\in\mathcal{X}} \sum_{i=1}^{n} \left(\sum_{k=0}^{K} k y_{ik}\right) x_i \geq r, \\ & \sum_{k=0}^{K} y_{ik} = 1, \text{ for } i \in [n],\ y_{ik} \in \{0,1\}, \text{ for } i \in [n], k \in [0,K]. \end{aligned}$$

This is obtained by defining the binary variable y_{ik} as $\mathbb{1}_{\{c_i = k\}}$. Towards further simplification, for any $\mathbf{x} \in \mathcal{X}$, $\boldsymbol{\alpha} \in \mathbb{R}^{n\times(K+1)}$ and $r \in [0, nK]$, define $G(\boldsymbol{\alpha}, \mathbf{x})$ as the optimal value of the following 0-1 integer program:

$$\begin{aligned} G(\boldsymbol{\alpha}, \mathbf{x}) = \min \quad & \sum_{i=1}^{n}\sum_{k=0}^{K} \alpha_{ik} y_{ik} \\ \text{s.t.} \quad & \sum_{i=1}^{n} \left(\sum_{k=0}^{K} k y_{ik}\right) x_i \geq r,\ \sum_{k=0}^{K} y_{ik} = 1, \text{ for } i \in [n], \\ & y_{ik} \in \{0,1\}, \text{ for } i \in [n], k \in [0,K]. \end{aligned}$$

Then we have that $\lambda + W(\boldsymbol{\alpha}) \geq 1 \iff \lambda + G(\boldsymbol{\alpha}, \mathbf{x}) \geq 1$ for $\mathbf{x} \in \mathcal{X}$. We next propose a dynamic programming reformulation to compute $G(\boldsymbol{\alpha}, \mathbf{x})$. The dynamic programming solution further aids in developing our linear program. Let $f_{i,l,\mathbf{x}}$ denote the optimal value of the subproblem making optimal assignments for the variables y_{jk} for $j \in [i]$:

$$\begin{aligned} f_{i,l,\mathbf{x}} = \min \quad & \sum_{j=1}^{i}\sum_{k=0}^{K} \alpha_{jk} y_{jk} \\ \text{s.t.} \quad & \sum_{j=1}^{i} \left(\sum_{k=0}^{K} k y_{jk}\right) x_j = l, \\ & \sum_{k=0}^{K} y_{jk} = 1, \text{ for } j \in [i],\ y_{jk} \in \{0,1\}, \text{ for } j \in [i], k \in [0,K]. \end{aligned}$$

Thus $f_{i,l,x}$ denotes the optimal value of the objective when every random variable $c_j, j \in [1,n]$ must be assigned a value from $[0,K]$ and the partial sum $\sum_{j=1}^{i} c_j x_j = l$. However there is a cost α_{jk} for assigning $c_j = k$. Also variable j contributes to the partial sum only if $j \leq i$ and $x_j = 1$. The objective is to minimize the cost of the overall assignment to all random variables.

For the base case of the DP, we must have $f_{1,k,\mathbf{x}} = \alpha_{1,k}$ for $k \in [0,K]$ provided $x_1 = 1$, since $\sum_{j=1}^{1} c_j x_j = k$ only if $x_1 = 1$. Wheareas if $x_1 = 0$, $f_{1,0,\mathbf{x}} = \min_{k\in[0,K]} \alpha_{1,k}$ as $\sum_{j=1}^{1} c_j x_j = 0$ always. For $i > 1$, if $x_i = 1$, then $f_{i,l,\mathbf{x}}$ will take the smallest possible value of $f_{i-1,l-k,\mathbf{x}} + \alpha_{ik}$ out of all possible values of $k \in [0,K]$, and $y_{ik} = 1$ for the corresponding k. Whereas if $x_i = 0$, $f_{i,l,\mathbf{x}}$ will take the smallest possible value of $f_{i-1,l,\mathbf{x}} + \alpha_{ik}$. So we have:

$$f_{i,l,\mathbf{x}} = \min_{k\in[0,K]} (f_{i-1,l,\mathbf{x}} + \alpha_{ik})(1 - x_i) + \min_{k\in[0,K]} (f_{i-1,l-k,\mathbf{x}} + \alpha_{ik})\, x_i.$$

Finally, the optimal objective is $\min_{r\le l\le nK} f_{n,l,x}$. Putting together the dynamic programming recursion gives us the following linear program:

$$
\begin{aligned}
G(\boldsymbol{\alpha},\mathbf{x}) = \max \quad & t_{\mathbf{x}}\\
\text{s.t.} \quad & f_{n,l,\mathbf{x}} - t_{\mathbf{x}} \ge 0 \text{ for } l\in[r,nK],\\
& f_{i-1,l-k,\mathbf{x}} + \alpha_{ik} - f_{i,l,\mathbf{x}} \ge 0,\ \text{for } i\in[2,n], l\in[0,nK],\\
& \qquad\qquad k\in[0,\min(l,K)],\ \text{for } \mathbf{x}\in\mathcal{X} \text{ with } x_i=1\\
& f_{i-1,l,\mathbf{x}} + \alpha_{ik} - f_{i,l,\mathbf{x}} \ge 0,\ \text{for } i\in[2,n], l\in[0,nK],\\
& \qquad\qquad k\in[0,K],\ \text{for } \mathbf{x}\in\mathcal{X} \text{ with } x_i=0,\\
& -f_{1,k,\mathbf{x}} + \alpha_{1,k} = 0 \text{ for } k\in[0,K] \text{ for } \mathbf{x}\in\mathcal{X} \text{ with } x_1=1,\\
& -f_{1,0,\mathbf{x}} + \alpha_{1,k} \ge 0 \text{ for } k\in[0,K] \text{ for } \mathbf{x}\in\mathcal{X} \text{ with } x_1=0.
\end{aligned} \tag{2.3}
$$

Forcing $\lambda + G(\boldsymbol{\alpha},\mathbf{x})$ to be greater than 1, provides the following equivalent reformulation of the constraint (2.1):

$$
\begin{aligned}
& \lambda + t_{\mathbf{x}} \ge 1\\
& f_{n,l,\mathbf{x}} - t_{\mathbf{x}} \ge 0 \text{ for } l\in[r,nK],\\
& f_{i-1,l-k,\mathbf{x}} + \alpha_{ik} - f_{i,l,\mathbf{x}} \ge 0,\ \text{for } i\in[2,n], l\in[0,nK],\\
& \qquad\qquad k\in[0,\min(l,K)],\ \text{for } \mathbf{x}\in\mathcal{X} \text{ with } x_i=1\\
& f_{i-1,l,\mathbf{x}} + \alpha_{ik} - f_{i,l,\mathbf{x}} \ge 0,\ \text{for } i\in[2,n], l\in[0,nK],\\
& \qquad\qquad k\in[0,K],\ \text{for } \mathbf{x}\in\mathcal{X} \text{ with } x_i=0,\\
& -f_{1,k,\mathbf{x}} + \alpha_{1,k} = 0 \text{ for } k\in[0,K] \text{ for } \mathbf{x}\in\mathcal{X} \text{ with } x_1=1,\\
& -f_{1,0,\mathbf{x}} + \alpha_{1,k} \ge 0 \text{ for } k\in[0,K],\ \text{for } \mathbf{x}\in\mathcal{X} \text{ with } x_1=0.
\end{aligned}
$$

Step (2): Reformulating constraints (2.2): Enforcing (2.2) boils down to ensuring,

$$
\lambda + \min\left\{\sum_{i=1}^{n}\sum_{k=0}^{K}\alpha_{ik}y_{ik} : \sum_{k=0}^{K} y_{ik} = 1,\ \text{for } i\in[n], y_{ik}\in\{0,1\},\ \text{for } i\in[n], k\in[0,K]\right\} \ge 0.
$$

It is easy to see that the optimal value of the optimization problem is attained by $y_{ik} = 1$ for $k = \text{argmin}_{k\in[0,K]}\,\alpha_{ik}$ for all $i\in[n]$. Thus the constraint can be reformulated as, $\lambda + \max\{\sum_{i=1}^{n} v_i : \alpha_{ik} - v_i \ge 0,\ \text{for } i\in[1,n], k\in[0,K]\} \ge 0$. Then integrating all the constraints together gives us the following linear program:

$$
\begin{aligned}
\min \quad & \lambda + \sum_{i=1}^{n}\sum_{k=0}^{K}\alpha_{ik}p_{ik}\\
\text{s.t.} \quad & \lambda + t_{\mathbf{x}} \ge 1,\\
& f_{n,l,\mathbf{x}} - t_{\mathbf{x}} \ge 0 \text{ for } l\in[r,nK],\\
& f_{i-1,l-k,\mathbf{x}} + \alpha_{ik} - f_{i,l,\mathbf{x}} \ge 0,\ \text{for } i\in[2,n], l\in[0,nK],\\
& \qquad\qquad k\in[0,\min(l,K)],\ \text{for } \mathbf{x}\in\mathcal{X} \text{ with } x_i=1\\
& f_{i-1,l,\mathbf{x}} + \alpha_{ik} - f_{i,l,\mathbf{x}} \ge 0,\ \text{for } i\in[2,n], l\in[0,nK],\\
& \qquad\qquad k\in[0,K],\ \text{for } \mathbf{x}\in\mathcal{X} \text{ with } x_i=0,\\
& -f_{1,k,\mathbf{x}} + \alpha_{1,k} = 0 \text{ for } k\in[0,K] \text{ for } \mathbf{x}\in\mathcal{X} \text{ with } x_1=1,\\
& -f_{1,0,\mathbf{x}} + \alpha_{1,k} \ge 0 \text{ for } k\in[0,K],\ \text{for } \mathbf{x}\in\mathcal{X} \text{ with } x_1=0,\\
& \lambda + \sum_{i=1}^{n} v_i \ge 0,\\
& \alpha_{ik} - v_i \ge 0,\ \text{for } i\in[1,n], k\in[0,K].
\end{aligned}
$$

The dual of this LP is the tight reformulation in the theorem. The LP has a total of $O(n^2K^2P)$ variables and $O(n^2K^2P)$ constraints. The maximum value from the input univariate distributions p_{ik} requires $\log_2(U_1)$ bits for representation. Hence the linear program is solvable in time polynomial in n, K, P and $\log_2(U_1)$. ∎

When P is polynomial in n, this is a polynomial sized linear program in comparison to the original primal linear program which has $O(K^n)$ variables. All the variables in the linear program can be interpreted as probabilities of appropriate events. Let θ^* be a distribution that attains the objective $U(r)$. Then, for example, $a_{\mathbf{x}}$ can intuitively be thought of as the probability $\mathbb{P}_{\theta^*}(Z(\tilde{\mathbf{c}}) \geq r, \mathbf{x} \in \text{argmax}_{\bar{x} \in \mathcal{X}} \tilde{\mathbf{c}}'\bar{\mathbf{x}})$, $\delta_{ikl\mathbf{x}}$ may be interpreted as $\mathbb{P}_{\theta^*}(Z(\tilde{\mathbf{c}}) \geq r, \mathbf{x} \in \text{argmax}_{\bar{x} \in \mathcal{X}} \tilde{\mathbf{c}}'\bar{\mathbf{x}}, \tilde{c}_i = k, \sum_{j=1}^{i} \tilde{c}_j x_j = l)$, $h_{ik} = \mathbb{P}_{\theta^*}(Z(\tilde{\mathbf{c}}) < r, \tilde{c}_i = k)$. The constraints on δ then capture the fact that if $\sum_{j=1}^{i} \tilde{c}_j x_j = l$ then $\sum_{j=1}^{i+1} \tilde{c}_j x_j = l + k'$ if $\tilde{c}_{i+1} = k'$ and $x_{i+1} = 1$. However if $x_{i+1} = 0$, then the partial sum is preserved upto $i+1$ so that $\sum_{j=1}^{i+1} \tilde{c}_j x_j = l$.

COROLLARY 2.2. *Suppose* $\mathbf{a}^*, b^*, \boldsymbol{\delta}^*, \mathbf{h}^*, \boldsymbol{\tau}^*$ *denote an optimal solution to the LP in Theorem 2.1. A distribution that meets the value* $U(r)$ *can be constructed as follows:*

1. *Generate a Bernoulli random variable* $\tilde{z} = 1$ *with probability* $\sum_{\mathbf{x} \in \mathcal{X}} a^*_{\mathbf{x}}$.
2. *If* $\tilde{z} = 1$,
   - (a) *Generate* $\mathbf{x} \in \mathcal{X}$ *with probability* $a^*_{\mathbf{x}} / \sum_{\mathbf{x} \in \mathcal{X}} a^*_{\mathbf{x}}$.
   - (b) *If* $x_1 = 0$, *generate* $\tilde{c}_1 = k$ *with probability* $\delta_{1,k,0,x} / \sum_{k'} \delta_{1,k',0,x}$
     If $x_1 = 1$, *generate* $\tilde{c}_1 = k$ *with probability* $\delta_{1,k,k,x} / \sum_{k'} \delta_{1,k',k',x}$
   - (c) *Set* $l = \tilde{c}_1 x_1$
   - (d) *For* $i \in [2, n]$:
     - i. *If* $x_i = 0$, *generate* $\tilde{c}_i = k$ *with probability* $\delta_{i,k,l,x} / \sum_{k'} \delta_{i,k',l,x}$,
       If $x_i = 1$, *generate* $\tilde{c}_i = k$ *with probability* $\delta_{i,k,l+k,x} / \sum_{k'} \delta_{i,k',l+k',x}$,
     - ii. $l = l + \tilde{c}_i x_i$
3. *If* $\tilde{z} = 0$, *for* $i \in [n]$, *generate* $\tilde{c}_i = k$ *with probability* h^*_{ik}/b^*.

A proof sketch is provided in Appendix A. We will provide some insights on the extremal distribution constructed with an example PERT network.

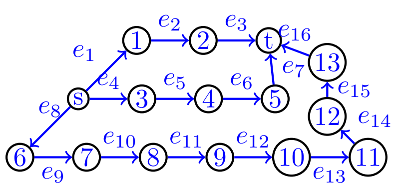


Fig. 1: An example network

Path	Edges	Len
1	(s,1), (1,2), (2,t)	3
2	(s,3), (3,4), (4,5), (5,t)	4
3	(s,6), $(i, i+1) \forall i \in [6, 12]$, (13,t)	9

Table 1: Paths in Example 2

We consider the network in Figure 1 with the s-t paths described in Table 1 and use $K = 2$. The network has 15 vertices and 16 edges. Each s-t path corresponds to an extreme point $\mathbf{x} \in \mathcal{X}$. All marginal distributions are identical with $p_{ek} = 1/3$ for all $e \in E, k \in [0, 2]$. For achieving a value of $Z(\tilde{\mathbf{c}}) > 8$, path 3 is critical. This is captured by the variables in our formulation wherein $a_1 = a_2 = 0$ and all possible mass gets concentrated on a_3 whenever $r > 8$. In particular for $r = 18$, the maximum value of $\mathbb{P}(Z(\tilde{\mathbf{c}}) \geq 18) = 1/3$ and for all edges e in path 3, $\delta_{e,0,18,\mathbf{x}}$ and $\delta_{e,1,18,\mathbf{x}}$ turn out to be zero. Our variables $\delta_{e,k,l,\mathbf{x}} = \mathbb{P}(\tilde{c}_e = k, \sum_{j \leq e} c_j x_j = l, \tilde{\mathbf{x}} = \mathbf{x})$. Indeed, if path 3 must have a total length of 18, then none of the edges in the path can be allowed to have a length of 0 or 1. Thus all edges in path 3 end up with mass concentrated on a support of 2 so that $\delta_{16,2,18,\mathbf{x}} = \mathbb{P}(\sum_i \tilde{c}_i x_i = 18, \tilde{c}_{16} = 2, \tilde{\mathbf{x}} = \text{Path 3}) = 1/3$ and is as large as it can get. Similarly in this context we note that the $h_{ek} = \mathbb{P}(\tilde{c}_e = k, Z(\tilde{\mathbf{c}}) < r)$ variables are such that no edge in path 3 is allowed to take a length of 2. In particular $h_{e0} = h_{e1} = 1/3$ and $h_{e2} = 0$ for all edges in path 3. This makes intuitive sense as such an assignment prevents $Z(\tilde{\mathbf{c}}) = 18$ and $\mathbf{h}$ variables capture the probabilities of assignments if the total length from all s-t paths is less than 18.

Further structural insights on the extremal distribution are not obvious to deduce as the extremal distribution varies greatly with r. For example, in the case of sums of random variables, where $\mathcal{X} = \{\mathbf{1}_n\}$, if $r = n$, it is known that the comonotonic distribution is an extremal distribution whereas if $r = 1$ the counter-monotone distribution serves as an extremal distribution [38].

2.1.1. Reduced Formulations for Bernoulli Random Variables. In the scenario where the random variables take support in $\{0, 1\}$, we show that the size of the linear program in Theorem 2.1 can be reduced by employing an alternative approach to tackle the separation problem: $\min \left\{\sum_{i=1}^n \alpha_i c_i : Z(\mathbf{c}) \geq r, \mathbf{c} \in \{0,1\}^n\right\}$.

THEOREM 2.3. *Let $\mathcal{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^P\} \subseteq \{0,1\}^n$. Given the marginal distributions of the Bernoulli random vector $\tilde{\mathbf{c}}$ as $\mathbb{P}(\tilde{c}_i = 1) = 1 - \mathbb{P}(\tilde{c}_i = 0) = p_i$ for $i \in [n]$, the tightest upper bound is computable by solving the linear program:*

$$
\begin{aligned}
U(r) = \max \quad & \sum_{\mathbf{x}\in\mathcal{X}} a_{\mathbf{x}} \\
s.t \quad & \sum_{\mathbf{x}\in\mathcal{X}} a_{\mathbf{x}} + b = 1, \\
& h_i + \sum_{\mathbf{x}\in\mathcal{X}} g_{i,\mathbf{x}} = p_i, \ \textit{for } i \in [n], \\
& h_i \leq b, \ \textit{for } i \in [n], \\
& g_{i,\mathbf{x}} \leq a_{\mathbf{x}}, \ \textit{for } i \in [n], \mathbf{x} \in \mathcal{X}, \\
& r a_{\mathbf{x}} - \sum_{i: x_i = 1} g_{i,\mathbf{x}} \leq 0, \ \textit{for } \mathbf{x} \in \mathcal{X}, \\
& \mathbf{a} \geq 0, b \geq 0, \mathbf{g} \geq 0, \mathbf{h} \geq 0.
\end{aligned}
$$

In particular $U(r)$ can be computed in time polynomial in n, P and $\log(U_1)$.

Proof. Constraint (2.2) in the exponential sized dual linear program for Bernoulli random variables can be rewritten as follows:

$$
\begin{aligned}
& \lambda + \min\left\{\sum_{i=1}^n \alpha_i c_i : c_i \in \{0,1\}^n\right\} \geq 0 \\
\iff \quad & \lambda + \min\left\{\sum_{i=1}^n \alpha_i c_i : 0 \leq c_i \leq 1, \ \text{for } i \in [n]\right\} \geq 0 \\
\iff \quad & \lambda + \sum_{i=1}^n -\eta_i \geq 0, \alpha_i + \eta_i \geq 0, \ \text{for } i \in [n], \boldsymbol{\eta} \geq 0,
\end{aligned}
$$

where the first equivalence follows from the 0/1 extreme points of the unit hypercube and the second equivalence is from LP duality. Constraint (2.1) can be rewritten as follows,

$$
\begin{aligned}
& \min\left\{\sum_{i=1}^n \alpha_i c_i : \max_{\mathbf{x}\in\mathcal{X}} \mathbf{c}^\top \mathbf{x} \geq r, \mathbf{c} \in \{0,1\}^n\right\} \geq 1 - \lambda \\
\iff \quad & \min\left\{\sum_{i=1}^n \alpha_i c_i : \mathbf{c}^\top \mathbf{x} \geq r, 0 \leq c_i \leq 1, \ \text{for } i \in [n]\right\} \geq 1 - \lambda, \ \text{for } \mathbf{x} \in \mathcal{X},
\end{aligned}
$$

where the equivalence holds by disaggregating the constraints and using the observation that for each $\mathbf{x} \in \mathcal{X} \subseteq \{0,1\}^n$, the constraint $\mathbf{c}^\top \mathbf{x} \geq r$ has a totally unimodular structure. Note that while this totally unimodular structure arises with binary support, it breaks down for more general discrete support. Further dualizing the linear program for each $\mathbf{x} \in \mathcal{X}$ and enforcing the constraints gives the equivalent formulation:

$$
\begin{aligned}
& \lambda + r\Delta_{\mathbf{x}} - \sum_{i=1}^n \gamma_{i,\mathbf{x}} \geq 1, \ \text{for } \mathbf{x} \in \mathcal{X}, \ \alpha_i - \Delta_{\mathbf{x}} x_i + \gamma_{i,\mathbf{x}} \geq 0, \ \text{for } i \in [n], \mathbf{x} \in \mathcal{X}, \\
& \Delta_{\mathbf{x}} \geq 0, \ \text{for } \mathbf{x} \in \mathcal{X}, \gamma_{i,\mathbf{x}} \geq 0 \text{ for } i \in [n], \mathbf{x} \in \mathcal{X}.
\end{aligned}
$$

Putting the reformulations together in place of the dual constraints (2.2) and (2.1) in the exponential sized dual linear program gives:

$$\begin{aligned}
\min \quad & \lambda + \sum_{i=1}^{n} \alpha_i p_i \\
\text{s.t.} \quad & \lambda + \sum_{i=1}^{n} -\eta_i \geq 0,\ \alpha_i + \eta_i \geq 0,\ \text{for } i \in [n], \\
& \lambda + r\Delta_{\mathbf{x}} - \sum_{i=1}^{n} \gamma_{i,\mathbf{x}} \geq 1,\ \text{for } \mathbf{x} \in \mathcal{X}, \\
& \alpha_i - \Delta_{\mathbf{x}} x_i + \gamma_{i,\mathbf{x}} \geq 0,\ \text{for } i \in [n], \mathbf{x} \in \mathcal{X}, \\
& \Delta_{\mathbf{x}} \geq 0,\ \text{for } \mathbf{x} \in \mathcal{X}, \gamma_{i,\mathbf{x}} \geq 0,\ \text{for } i \in [n], \mathbf{x} \in \mathcal{X}, \eta_i \geq 0,\ \text{for } i \in [n].
\end{aligned}$$

The formulation in the theorem is the dual of the above LP with $O(nP)$ variables and $O(nP)$ constraints. Hence its time complexity is polynomial in n, P and $\log(U_1)$. ∎

While the above LP has $O(nP)$ variables and $O(nP)$ constraints, in comparison, the linear program in Theorem 2.1 applied to Bernoulli random variables has $O(n^2P)$ variables and $O(n^2P)$ constraints.

Next we describe the construction of the extremal distribution using the optimal solution of the linear program in Theorem 2.3. We first observe that the following interpretation for the variables as probabilities ensure that all the constraints in the LP are satisfied: $a_{\mathbf{x}} = \mathbb{P}(\sum_{i=1}^n \tilde{c}_i x_i \geq r)$, $b = \mathbb{P}(Z(\tilde{\mathbf{c}}) < r)$, $g_{i,x} = \mathbb{P}(\sum_{i=1}^n \tilde{c}_i x_i \geq r, \tilde{c}_i = 1)$, $h_i = \mathbb{P}(Z(\mathbf{c}) < r, \tilde{c}_i = 1)$. Let $\mathbf{g}_{\mathbf{x}} = [g_{1,x}, \ldots, g_{n,x}]$ and suppose $\mathcal{C}_{\mathbf{x},r} = \{\mathbf{c} \in \{0,1\}^n : \sum_{i=1}^n c_i x_i \geq r\}$ and $\bar{\mathcal{C}}_r = \{\mathbf{c} \in \{0,1\}^n : Z(\mathbf{c}) < r\}$. All components of the vector $\mathbf{g}_{\mathbf{x}}/a_{\mathbf{x}}$ lie in the interval $[0,1]$ and also we have $\sum_i g_{i,\mathbf{x}} x_i \geq r a_{\mathbf{x}}$. Therefore the vector $\mathbf{g}_{\mathbf{x}}/a_{\mathbf{x}}$ can be expressed as convex combination of the extreme points of the set $\{\mathbf{c} \in \{0,1\}^n : \sum_{i=1}^n c_i x_i \geq r\}$. Thus the following representations exist:

- For all $\mathbf{x} \in \mathcal{X}, \mathbf{g}_{\mathbf{x}}/a_{\mathbf{x}} = \sum_{\mathbf{c} \in \mathcal{C}_{\mathbf{x},r}} \lambda_{\mathbf{c},\mathbf{x}} \mathbf{c}$ with $\sum_{\mathbf{c} \in \mathcal{C}_{\mathbf{x},r}} \lambda_{\mathbf{c},\mathbf{x}} \mathbf{c} = 1$.
- $\mathbf{h}/b = \sum_{\mathbf{c} \in \bar{\mathcal{C}}_r} \gamma_{\mathbf{c}} \mathbf{c}$ with $\sum_{\mathbf{c} \in \bar{\mathcal{C}}_r} \gamma_{\mathbf{c}} \mathbf{c} = 1$.

We now construct a distribution that meets the objective $U(r)$ in Theorem 2.3.

COROLLARY 2.4. *Given an optimal solution of the linear program denoted by $\mathbf{a}^*, b, \boldsymbol{g}^*, \boldsymbol{h}^*$, an extremal distribution is constructed using the following mixture distribution:*

1. *Generate a Bernoulli random variable $\tilde{z} = 1$ with probability $\sum_{\mathbf{x} \in \mathcal{X}} a^*_{\mathbf{x}}$.*
2. *If $\tilde{z} = 1$,*
   (a) *Generate $\mathbf{x} \in \mathcal{X}$ with probability $a^*_{\mathbf{x}} / \sum_{\mathbf{x} \in \mathcal{X}} a^*_{\mathbf{x}}$.*
   (b) *Generate $\mathbf{c} \in \mathcal{C}_{\mathbf{x},r}$ with probability $\lambda_{c,r}$.*
3. *If $\tilde{z} = 0$, for $i \in [n]$, generate $\mathbf{c} \in \bar{\mathcal{C}}_r$ with probability γ_c.*

We now consider an application of our techniques to the sum of random variables.

2.2. Application to the Sum of Random Variables. The computation of probability bounds for the sum of dependent random variables has received much attention in the literature. In particular, there have been many upper and lower bounds developed with general marginal distributions (discrete or continuous) in the works of [39, 14, 34, 44, 43, 7] and the references therein. These bounds are typically generated by choosing appropriate dual feasible solutions and are guaranteed to be tight in special cases [38]. Given the hardness results for computing these bounds, it is of interest to find instances where the tight bounds are polynomial time computable.

We now discuss the application of Theorem 2.1 to computing bounds for sums of dependent random variables with discrete marginal distributions. Let $S(r, K)$ denote the following probability bound:

$$S(r,K) = \max\left\{\mathbb{P}_\theta\left(\sum_{i=1}^n \tilde{c}_i \geq r\right) : \mathbb{P}_\theta(\tilde{c}_i = k) = p_{ik},\ \text{for } k\in[K], i\in[n], \theta\in\mathbb{P}([0,K]^n)\right\}.$$

For Bernoulli random variables with $K=1$ where $p_{i0} = 1-p_i$ and $p_{i1} = p_i$, the tightest upper bound for $r=1$ is given by Boole's union bound: $S(1,1) = \min(\sum_{i=1}^n p_i, 1)$. For more general values of $r \in [n]$, the tightest upper bound for the sum of dependent Bernoulli random variables was computed in closed form by [37]:

$$S(r,1) = \min\left(\left(\min_{t\in[0,r-1]} \sum_{i=1}^{n-t} \frac{p_{(i)}}{r-t}\right), 1\right), \tag{2.4}$$

where the marginal probabilities $p_1, p_2, \dots, p_n$ are ordered as $p_{(1)} \leq p_{(2)} \leq \dots \leq p_{(n)}$. For the sum of discrete random variables with support in $[0,K]$, directly applying Theorem 2.1 brings us to the following corollary which shows that the tightest bound is computable in polynomial time. This adds to the stream of literature on identifying instances where the tightest upper bound is computable in polynomial time.

Corollary 2.5. *Given the marginal distributions of the random vector* $\tilde{\mathbf{c}}$ *as* $\mathbb{P}(\tilde{c}_i = k) = p_{ik}$ *for* $k\in[0,K]$ *and* $i\in[n]$*, the tightest upper bound on the sum exceeding a value* r *is computable by solving the linear program:*

$$\begin{aligned}
S(r,K) = \max \quad & a \\
s.t. \quad & a + b = 1, \\
& h_{ik} + \sum_{l=k}^{nK} \delta_{i,k,l} = p_{ik},\ \textit{for } i\in[n], k\in[0,K], \\
& \sum_{k=0}^{K} h_{ik} = b,\ \textit{for } i\in[n], \\
& a = \sum_{l=r}^{nK} \tau_l,\ \tau_l = \sum_{k=0}^{\min(K,l)} \delta_{n,k,l},\ \textit{for } l\in[r,nK], \\
& \delta_{n,k,l} = 0,\ \textit{for } l\in[0,r-1], k\in[0,\min(K,l)], \\
& \delta_{1,k,k} = \sum_{k'=0}^{\min(K,nK-k)} \delta_{2,k',k'+k},\ \textit{for } k\in[0,K], \\
& \delta_{2,k,k'+k} = 0,\ \textit{for } k\in[0,K], k'\in[K+1,nK-k], \\
& \sum_{k=0}^{\min(K,l)} \delta_{i,k,l} = \sum_{k'=0}^{\min(K,nK-l)} \delta_{i+1,k',l+k'}, \\
& \qquad \textit{for } i\in[2,n-1], l\in[0,nK], \\
& a, b, \mathbf{h}, \boldsymbol{\delta}, \boldsymbol{\tau} \geq 0.
\end{aligned}$$

The linear program is polynomial sized in n*,* K *and* $\log(U_1)$*.*

Next we describe the construction of the extremal distribution using the optimal solution of the linear program in Corollary 2.5. Given an optimal solution of the linear program denoted by $a^*, b^*, \mathbf{h}^*, \boldsymbol{\delta}^*, \boldsymbol{\tau}^*$, an extremal distribution is constructed using the following mixture distribution:

1. Generate a Bernoulli random variable $\tilde{z}$ with probability a^*.
2. If $\tilde{z} = 1$,
   (a) Generate $\tilde{c}_1 = k$ with probability $\delta_{1,k,k}/a$.

(b) For each i in $[2, n]$, generate $\tilde{c}_i$ as follows:

$$\mathbb{P}(\tilde{c}_i = k \;\Big|\; \sum_{j=1}^{i} \tilde{c}_i = l) = \frac{\delta_{i,k,l+k}}{\displaystyle\sum_{k' \in [0,K]} \delta_{i-1,k',l}}, \text{ for } l \in [0, iK].$$

3. If $\tilde{z} = 0$, generate $\tilde{c}_i = k$ with probability h_{ik}/b independently across $i \in [n]$.

It can be verified that θ^* is the extremal distribution where the optimal decision variables can be interpreted as: $a^* = \mathbb{P}_{\theta^*}(\sum_i \tilde{c}_i \geq r)$, $b^* = \mathbb{P}_{\theta^*}(\sum_i \tilde{c}_i < r)$. Additionally, $h^*_{ik} = \mathbb{P}_{\theta^*}(\tilde{c}_i = k, \sum_{j=1}^n \tilde{c}_j < r)$, $\tau^*_l = \mathbb{P}_{\theta^*}(\sum_{i=1}^n \tilde{c}_i \geq r, \sum_{i=1}^n \tilde{c}_i = l)$ and $\delta_{i,k,l} = \mathbb{P}_{\theta^*}(\tilde{c}_i = k, \sum_{j=1}^i \tilde{c}_i = l, \sum_{l=1}^n \tilde{c}_l = n)$.

2.2.1. Weighted Probability Bounds. In this section, we show that the results in Corollary 2.5 can be extended to compute tight weighted probability bounds of sums of discrete random variables as the optimal value of a compact linear program. Such bounds are useful in modeling scenarios where some of the variables are extremally dependent (assuming only knowledge of the marginal distributions), while the rest are mutually independent and the two sets of variables are independent of each other (see Subsection 4.1 for a numerical example). We can thus offset the inherent conservatism in the extremally dependent and mutually independent models by introducing a limited degree of independence into the model. Denote by $\mathbf{w} = (w_1, w_2, \ldots, w_{nK}),\ w_i \in \mathbb{R},\ i \in [nK]$ a vector of pre-specified weights. We are interested in computing the following tight upper bound on the weighted sum of the tail probabilities

$$\max_{\theta \in \Theta} \sum_{l=0}^{nK} w_l \mathbb{P}_\theta \left(\sum_{i=1}^n \tilde{c}_i \geq l \right).$$

Note that without loss of generality, we can ignore $\ell = 0$ and consider $\mathbb{P}_\theta(\sum_{i=1}^n \tilde{c}_i = l)$ for $\ell \in [nK]$ instead of tail probabilities by a suitable transformation of weights. Denote by $S(\mathbf{w}, K)$ the following upper bound:

$$S(\mathbf{w}, K) = \max_{\theta \in \Theta} \sum_{l=1}^{nK} w_l \mathbb{P}_\theta \left(\sum_{i=1}^n \tilde{c}_i = l \right),$$

where we are given the marginal distributions of the discrete random vector $\tilde{\mathbf{c}}$ as $\mathbb{P}(\tilde{c}_i = k) = p_{ik}$ for $k \in [0, K]$ and $i \in [n]$. We next prove the result for sums of Bernoulli random variables ($K = 1$) which can then be extended to the case $K \geq 2$.

THEOREM 2.6. *Given the marginal distributions of a Bernoulli random vector* $\tilde{\mathbf{c}}$ *as* $\mathbb{P}(\tilde{c}_i = 1) = p_i$ *for* $i \in [n]$*, the tightest upper bound* $S(\mathbf{w}, 1)$ *is computable via the LP:*

$$\begin{aligned} S(\mathbf{w}, 1) = \quad \max \quad & \sum_{l=0}^n \tau_l w_l \\ s.t. \quad & \sum_{l=0}^n \tau_l = 1, \\ & \sum_{l=0}^n \delta_{li} = p_i, && \textit{for } i \in [n], \\ & \tau_l \geq \delta_{li}, && \textit{for } i \in [n],\ \textit{for } l \in [n], \\ & \sum_{i=1}^n \delta_{li} = l\tau_l, && \textit{for } l \in [n], \\ & \tau_l \geq 0, && \textit{for } l \in [n], \\ & \delta_{li} \geq 0, && \textit{for } i \in [n],\ \textit{for } l \in [n]. \end{aligned} \tag{2.5}$$

Proof. The tight bound $S(\mathbf{w}, 1)$ can be computed as the optimal value of the following exponential sized linear program:

$$\begin{aligned}
& \max && \sum_{l=0}^{n} w_l \sum_{\mathbf{c}\in[0,1]^n : \sum_{t=1}^{n} c_t = \ell} \theta(\mathbf{c}) && \\
& \text{s.t.} && \sum_{\mathbf{c}\in[0,1]^n : c_i = 1} \theta(\mathbf{c}) = p_i, && \text{for } i \in [n], \\
& && \sum_{\mathbf{c}\in[0,1]^n} \theta(\mathbf{c}) = 1, && \\
& && \theta(\mathbf{c}) \geq 0 && \text{for } \mathbf{c} \in [0,1]^n.
\end{aligned} \tag{2.6}$$

An optimal solution of this linear program always exists with a finite optimal value. Note that when $\mathbf{w} = (\mathbf{0}_{r-1}, \mathbf{1}_{n-r+1})$ (zeros up to index $r-1$ and ones thereafter), the objective function in (2.6) reduces to the tail probability bounds $S(r, 1)$ considered in Subsection 2.2. We next derive a compact reformulation of (2.6) by considering the linear relaxation of its dual separation problem, similar to the proof of Theorem 2.3 with $\mathcal{X} = \{\mathbf{1}_n\}$. The dual of the linear program (2.6) can be written as:

$$\begin{aligned}
& \min && \sum_{i=1}^{n} \alpha_i p_i + \lambda \\
& \text{s.t.} && \sum_{i=1}^{n} \alpha_i c_i + \lambda \geq w_l, \quad \text{for } \mathbf{c} \in [0,1]^n : \textstyle\sum_{i=1}^{n} c_i = l, \ \text{for } l \in [n].
\end{aligned} \tag{2.7}$$

The dual linear program (2.7) has 2^n constraints, which can be divided into n sets of $\binom{n}{l}$ constraints for $l \in [n]$. Similar to the steps followed in the derivation of the reduced formulation for Bernoulli variables in Theorem 2.3, for each $l \in [n]$, the set of $\binom{n}{l}$ constraints corresponding to the scenarios $\mathbf{c} \in [0,1]^n : \sum c_i = l$ can be rewritten as follows:

$$\begin{aligned}
& \lambda + \left\{ \min \sum_{i=1}^{n} \alpha_i c_i : \mathbf{c} \in [0,1]^n, \ \sum_{i=1}^{n} c_i = l \right\} \geq w_l, && \text{for } l \in [n] \\
\iff \ & \lambda + \left\{ \min \sum_{i=1}^{n} \alpha_i c_i : 0 \leq c_i \leq 1, \ \text{for } i \in [n], \ \sum_{i=1}^{n} c_i = l \right\} \geq w_l, && \text{for } l \in [n] \\
\iff \ & \lambda + \left\{ \begin{aligned} & \max && \sum_{i=1}^{n} u_{li} + l v_l \\ & \text{s.t.} && u_{li} + v_l \leq \alpha_i, \quad \text{for } i \in [n], \\ & && u_{li} \leq 0, \quad \text{for } i \in [n], \end{aligned} \right\} \geq w_l, && \text{for } l \in [n],
\end{aligned} \tag{2.8}$$

where the first equivalence follows from the totally unimodular structure of the constraint matrix and the second equivalence is from linear programming duality. Since an optimal solution to the primal (2.6) exists, by strong duality, the dual (2.7) must also have an optimal solution. Consequently there must exist a feasible solution to the linear program in the last equivalence of (2.8) and the constraint sets corresponding to each $l \in [0, n]$ in (2.7) can be replaced by the following polynomial-sized set of constraints:

$$\left\{ \begin{aligned} & \lambda + \sum_{i=1}^{n} u_{li} + l v_l \geq w_l, && \\ & u_{li} + v_l \leq \alpha_i, && \text{for } i \in [n], \\ & u_{li} \leq 0, && \text{for } i \in [n], \end{aligned} \right\}, \quad \text{for } l \in [n]. \tag{2.9}$$

Thus the compact version of the dual (2.7) can be written as:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n} \alpha_i p_i + \lambda \\
\text{s.t.} \quad & \lambda - \sum_{i=1}^{n} u_{li} + l v_l \geq w_l, && \text{for } l \in [n], \\
& v_l - u_{li} \leq \alpha_i, && \text{for } i \in [n], \text{ for } l \in [n], \\
& u_{li} \geq 0, && \text{for } i \in [n], \text{ for } l \in [n].
\end{aligned} \tag{2.10}$$

Finally, dualizing (2.10) leads to the compact linear program (2.5) with $O(n^2)$ variables and constraints. ∎

It is straightforward to generalize the result in Theorem 2.6 to compute the tight bound on the weighted probability of sums of discrete random variables $S(\mathbf{w}, K)$ by a combination of techniques used in the proofs of Corollary 2.5 and Theorem 2.6.

We note interesting connections with state of the art if the weight vector were to be intepreted as a function so that $w(l) = w_l$. The objective then can be viewed as,

$$S(\mathbf{w}, K) = \max \sum_{l=1}^{nK} w_l \mathbb{E}[\mathbb{1}[\sum_{i=1}^{n} \tilde{c}_i = l]] = \max \mathbb{E}[\sum_{l=1}^{nK} w_l \mathbb{1}[\sum_{i=1}^{n} \tilde{c}_i = l]] = \max \mathbb{E}\left[w\left(\sum_{i=1}^{n} \tilde{c}_i\right)\right]$$

In a special case where the function $w(.)$ is convex, it is well known that the comonotone distribution attains the maximum value [38]. However on the other hand when $w(.)$ is concave, the analysis appears to be harder and the theory of joint mixability [42] has been used to solve some special cases in this context.

2.3. Hardness Results for the Lower Bound and Independence. In this section, we show both $L(r)$ and $I(r)$ are not computable in polynomial time for compact 0/1 V-polytopes unless P = NP. The hardness results are shown using a reduction from the independent set problem in graphs. An independent set in an undirected graph $G = (V, E)$ is a subset of the vertices such that no two vertices are adjacent to one another. The decision and optimization version of this problem are known to be NP-hard while counting the number of independent sets is known to be #P hard [16]. The next theorem shows computing the lower bound $L(r)$ is NP-hard.

THEOREM 2.7. *Let $\mathcal{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^P\} \subseteq \{0,1\}^n$. Given the marginal distributions of the Bernoulli random vector $\tilde{\mathbf{c}}$ as $\mathbb{P}(\tilde{c}_i = 1) = 1 - \mathbb{P}(\tilde{c}_i = 0) = p_i$ for $i \in [n]$, computation of the lower bound $L(r)$ is NP-hard and cannot be computed in time polynomial in the input size unless P = NP.*

Proof. The relevant separation problem to be solved to compute $L(r)$ is:

$$\max \left\{ \sum_{i=1}^{n} \alpha_i c_i : \mathbf{c}'\mathbf{x}^j \leq r - 1, \text{ for } j \in [P], c_i \in \{0,1\}, \text{ for } i \in [n] \right\}, \tag{2.11}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^n$ is given. This is NP-hard to solve. To see this, consider a graph $G = (V, E)$ on n nodes. Given an undirected graph $G = (V, E)$, let $n = |V|$ and $P = |E|$. Define the set $\mathcal{X}$ as the set of incidence vectors of the graph:

$$\mathcal{X} = \{\mathbf{x}^e; e \in E\} \subseteq \{0,1\}^n,$$

where for any $e = (i, j) \in E$, we let $x_i^e = 1$, $x_j^e = 1$ and $x_k^e = 0$ for all $k \neq i, j$. Setting $\alpha_i = 1$ for all i and $r = 2$ in (2.11) solves the maximum independent set problem. Since the separation problem is NP-hard to solve, the optimization problem is NP-hard to solve and computing $L(r)$ is NP-hard. ∎

Remark: A special case where the lower bound $L(r)$ is efficiently computable is in the case of sums of discrete random variables. This case turns out to be so due to an appropriate transformation to the upper bound $U(.)$ as detailed below.

$$\min_{\theta\in\Theta}\mathbb{P}_\theta(\sum_{i=1}^n \tilde{c}_i \geq r) = 1-\max_{\theta\in\Theta}\mathbb{P}_\theta(\sum_{i=1}^n \tilde{c}_i \leq r-1) = 1-\max_{\theta\in\Theta}\mathbb{P}_\theta(\sum_{i=1}^n (K-\tilde{c}_i) \geq nK-r+1)$$
$$= 1-\max_{\theta\in\bar{\Theta}}\mathbb{P}_\theta(\sum_{i=1}^n \tilde{d}_i \geq nK-r+1),$$

where $\tilde{d}_i$ is a random variable such that $\mathbb{P}(\tilde{d}_i = j) = \mathbb{P}(\tilde{c}_i = K-j) = p_{i,K-j}$, and $\bar{\Theta}$ consists of all distributions consistent with the marginal distributions of $\tilde{\mathbf{d}}$.

We next discuss hardness results for computing the probabilities with independent random variables. The next theorem from [22] shows that computing the probability of the sum of independent discrete random variables is #P-hard.

Theorem 2.8. *[22] Let $\tilde{c}_i$ be a two point random variable with $\mathbb{P}(\tilde{c}_i = a_i) = 1-\mathbb{P}(\tilde{c}_i = 0) = p_i$ for $a_i \in \mathbb{Z}_+$. Computing the probability $I(r) = \mathbb{P}_{\theta_{ind}}(\sum_{i=1}^n \tilde{c}_i \geq r)$ is #P-hard.*

The hardness in Theorem 2.8 was shown using a reduction from the counting version of the knapsack problem. The hardness result in their construction arises from the support of the random variables. When the random variables have restricted support such as Bernoulli, the sum is a Poisson Binomial random variable for which the probability is computable in polynomial time through recursions [9]. We next show however that for $Z(\tilde{\mathbf{c}})$ given as the optimal value of a maximization problem over a compact 0/1 V-polytope, computing the probability under the assumption of independence is hard even when the random variables are Bernoulli.

Theorem 2.9. *Let $\mathcal{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^P\} \subseteq \{0,1\}^n$. Given the marginal distributions of the Bernoulli random vector $\tilde{\mathbf{c}}$ as $\mathbb{P}(\tilde{c}_i = 1) = 1-\mathbb{P}(\tilde{c}_i = 0) = p_i$ for $i \in [n]$, computation of the probability $\mathbb{P}_{\theta_{ind}}(Z(\tilde{\mathbf{c}}) \geq r)$ is #P-hard and cannot be computed in time polynomial in the input size unless P = NP.*

Proof. We will do a reduction from counting the number of independent sets in a graph. Given an undirected graph $G = (V,E)$, let $n = |V|$ and $P = |E|$. Define the set $\mathcal{X}$ as the set of incidence vectors of the graph:

$$\mathcal{X} = \{\mathbf{x}^e; e \in E\} \subseteq \{0,1\}^n,$$

where for any $e = (i,j) \in E$, we let $x_i^e = 1$, $x_j^e = 1$ and $x_k^e = 0$ for all $k \neq i,j$. Let $\mathbb{P}(\tilde{c}_i = 1) = 1-\mathbb{P}(\tilde{c}_i = 0) = 1/2$ and $r = 2$. Then:

$$\mathbb{P}_{\theta_{ind}}\left(\max_{e\in E}\tilde{\mathbf{c}}'\mathbf{x}^e \geq 2\right) = 1-\mathbb{P}_{\theta_{ind}}\left(\max_{e\in E}\tilde{\mathbf{c}}'\mathbf{x}^e \leq 1\right)$$
$$= 1-\mathbb{P}_{\theta_{ind}}(\tilde{c}_i + \tilde{c}_j \leq 1 \text{ for } (i,j) \in E)$$
$$= 1-\frac{\text{No. of independent sets in G}}{2^n}.$$

Since computing the number of independent sets is #P-hard, so is computing $I(r)$. ∎

2.4. Extensions with Sparse Bivariate Information. While Theorem 2.9 establishes the hardness of computation of the tail probability $\mathbb{P}_{\theta_{ind}}(Z(\tilde{\mathbf{c}}) \geq r)$, we observe the family Θ can be further restricted with additional constraints so as to bring the bound $U(r)$ closer to $I(r)$. For example if one were to incorporate knowledge of sparse bivariate information, the polynomial solvability result of Theorem 2.1 would carry forward. For simplicity, we will give some pointers to this extension by restricting our attention to the case of Bernoulli random variables.

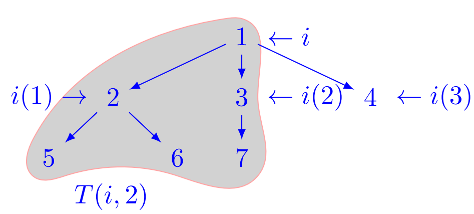
Let $\tilde{\mathbf{c}}$ be a Bernoulli random vector with known univariate marginal distributions and sparse bivariate marginals. In particular consider an undirected graph $G = (V, E)$ over the $|V| = n$ vertices. Assume that the bivariate distributions $\mathbb{P}(\tilde{c}_i = q_i, \tilde{c}_j = q_j)$ are known for $q_i \in \{0,1\}, q_j \in \{0,1\}$, for $(i,j) \in E$. Consider a family of distributions,

$$\Theta_{bv} = \left\{\theta \in \mathbb{P}(\{0,1\}^n) : \ \mathbb{P}_\theta\left(\tilde{c}_i = 1, \tilde{c}_j = 1\right) = p_{ij} \text{ for } (i,j) \in E, \ \ \mathbb{P}_\theta\left(\tilde{c}_i = 1\right) = p_i \text{ for } i \in V\right\}$$

Suppose $U_{bv}(r)$ denotes the largest value of the tail probability out of all disributions in Θ_{bv}, $U_{bv}(r) = \max_{\theta \in \Theta_{bv}} \mathbb{P}(Z(\tilde{\mathbf{c}}) \geq r)$. When G represents the complete graph over n vertices,Θ_{bv} represents the set of all joint distributions consistent with all pairs (a total of $n(n-1)/2$) of given bivariate distributions. This case is unfortunately computationally hard even for the case where $Z(\tilde{\mathbf{c}})$ denotes a sum of random variables. Checking for existence of a feasible joint distribution is itself hard [33].

In the case where the graph G is a tree, Θ_{bv} contains all joint distributions consistent with $n-1$ of the bivariate distributions (as specified by the edges in the tree). We focus on this sparsity structure of bivariate information. A special case where $Z(\tilde{\mathbf{c}}) = \sum_{i=1}^{n} \tilde{c}_i$ is studied in [32], wherein a tight linear programming formulation is proposed. These techniques can be used along with our approach proposed in Section 2 to obtain tightest possible bounds for $U_{bv}(r)$ when the underlying graph G is a tree and $Z(\tilde{\mathbf{c}})$ represents any linear optimization problem over a 0/1 polytope.

Fig. 2: Suppose $n = 7$ and the set of known bivariate marginal distributions corresponds to the set $E = \{(1,2),(2,5),(2,6),(3,7),(1,3),(1,4)\}$. The figure gives the corresponding directed tree with all arcs pointing away from the root node 1. The degrees of the various nodes are $d_1 = 3, d_2 = 2, d_3 = 1, d_4 = 0, d_5 = 0, d_6 = 0, d_7 = 0$. Let i denote the root node labelled 1. Assuming a non-decreasing order on the node labels, the three children of i are denoted as $i(1), i(2)$ and $i(3)$ (nodes 2,3 and 4 respectively). The sub-tree of i induced by the first two children is denoted by $T(i,2)$ and is shaded. The number of nodes in $T(i,2)$ is denoted by $N(i,2)$. $N(i,2) = 6$ here. The set of vertices in $T(i,2)$ is $V(i,2) = \{1,2,3,5,6,7\}$.



We now focus on a directed rooted tree representation of the graph where node 1 is designated as the root and the arcs are directed away from the root node. Assume an arbitrary but fixed ordering of the remaining nodes. We let d_i denote the outdegree of node i and denote the sth child of node i (as per the ordering fixed a-priori) as $i(s)$. We denote by $T(i,s)$ the sub-tree rooted at i consisting of the first s sub-trees of i where $V(i,s)$ is the set of vertices in $T(i,s)$ and $N(i,s)$ is the cardinality of this set. For ease of understanding, the notations are illustrated in Figure 2.

THEOREM 2.10. *For a tree graph G, the bound $U_{bv}(r)$ can be computed using a linear program polynomial sized in n and the number of extreme points P.*

$$
\begin{aligned}
\min_{\boldsymbol{\eta},\boldsymbol{\gamma},\boldsymbol{\Delta},\boldsymbol{\tau},\boldsymbol{\chi},\mathbf{f}} \quad & \lambda + \sum_{i=1}^{n} \alpha_i p_i + \sum_{(i,j)\in E} \beta_{ij} p_{ij} \\
s.t. \quad & \lambda - \sum_{(i,j)\in E} (\Delta_{ij} + \chi_{ij}) - \sum_{i=1}^{n} \tau_i \geq 0 \\
& \sum_{j:(i,j)\in E} (\Delta_{ij} - \eta_{ij}) + \sum_{j:(j,i)\in E} (\Delta_{ji} - \gamma_{ji}) + \tau_i + \alpha_i \geq 0 \;\; \text{for } i \in [n] \\
& \eta_{ij} + \gamma_{ij} - \Delta_{ij} + \chi_{ij} + \beta_{ij} \geq 0 \;\; \text{for } (i,j) \in E \\
& \lambda + z_{\mathbf{x}} \geq 1 \;\; \text{for } \mathbf{x} \in \mathcal{X} \\
& f_{1,d_1,0,t,\mathbf{x}} - z_{\mathbf{x}} \geq 0 \;\; \text{for } t \in [r,n], \mathbf{x} \in \mathcal{X} \\
& f_{1,d_1,1,t,\mathbf{x}} - z_{\mathbf{x}} \geq 0 \;\; \text{for } t \in [r,n], \mathbf{x} \in \mathcal{X} \\
& f_{i,s,0,0,\mathbf{x}} = 0 \quad \text{for } i \in [n], \text{ for } s \in [0,d_i], \mathbf{x} \in \mathcal{X} \\
& f_{i,s,1,x_1,\mathbf{x}} - \alpha_i = 0 \;\; \text{for } i \in [n], \text{ for } s \in [0,d_i], \mathbf{x} \in \mathcal{X} \\
& \text{For each internal node } i \text{ and each } \mathbf{x} \in \mathcal{X}: \\
& \quad f_{i(1),d_{i(1)},0,t,\mathbf{x}} - f_{i,1,0,t,\mathbf{x}} \geq 0 \;\; \text{for } t \in [0, N(i,1)-2] \\
& \quad f_{i(1),d_{i(1)},1,t,\mathbf{x}} - f_{i,1,0,t,\mathbf{x}} \geq 0 \;\; \text{for } t \in [x_{i(1)}, N(i,1) - 2 + x_{i(1)}] \\
& \quad f_{i(1),d_{i(1)},0,t-x_i,\mathbf{x}} - f_{i,1,1,t,\mathbf{x}} + \alpha_i \geq 0 \;\; \text{for } t \in [x_i, N(i,1) - 2 + x_i] \\
& \quad f_{i(1),d_{i(1)},1,t-x_i,\mathbf{x}} - f_{i,1,1,t,\mathbf{x}} + \alpha_i + \beta_{i,i(i)} \geq 0 \\
& \qquad\qquad \text{for } t \in [x_i + x_{i(1)}, N(i,1) - 2 + x_i + x_{i(1)}] \\
& \text{For each internal node } i \text{ with out-degree at least 2, and each } \mathbf{x} \in \mathcal{X}: \\
& \quad f_{i,s-1,0,t-a,\mathbf{x}} + f_{i(s),d_{i(s)},0,a,\mathbf{x}} - f_{i,s,0,t,\mathbf{x}} \geq 0 \\
& \qquad \text{for } s = [2,d_i], t = [0,(N(i,s)-2)], a = [a^1_{min}, a^1_{max}] \\
& \quad f_{i,s-1,0,t-a,\mathbf{x}} + f_{i(s),d_{i(s)},1,a,\mathbf{x}} - f_{i,s,0,t,\mathbf{x}} \geq 0 \\
& \qquad \text{for } s = [2,d_i], t = [x_{i(s)}, (N(i,s) - 2 + x_{i(s)})], a = [a^2_{min}, a^2_{max}] \\
& \quad f_{i,s-1,1,t-a,\mathbf{x}} + f_{i(s),d_{i(s)},0,a,\mathbf{x}} - f_{i,s,1,t,\mathbf{x}} \geq 0 \\
& \qquad \text{for } s = [2,d_i], t = [x_i, (N(i,s) - 2 + x_i)], a = [a^3_{min}, a^3_{max}] \\
& \quad f_{i,s-1,1,t-a,\mathbf{x}} + f_{i(s),d_{i(s)},1,a} - f_{i,s,1,t,\mathbf{x}} + \beta_{i,i(s)} \geq 0 \\
& \qquad \text{for } s = [2,d_i], t = [x_i + x_{i(s)}, (N(i,s) - 2 + x_i + x_{i(s)})], \\
& \qquad\quad a = [a^4_{min}, a^4_{max}] \\
& \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\Delta}, \boldsymbol{\tau}, \boldsymbol{\chi} \geq 0
\end{aligned}
$$

where $a^1_{min} = \max(0, t - (N(i,s-1)-1)), a^1_{max} = \min(N(i(s), d_{i(s)}) - 1 + x_{i(s)}, t)$, $a^2_{min} = \max(x_{i(s)}, t - (N(i,s-1)-1)), a^2_{max} = \min(N(i(s), d_{i(s)}) - 1 + x_{i(s)}, t)$, $a^3_{min} = \max(0, t - (N(i,s-1) - 1 + x_i)), a^3_{max} = \min(N(i(s), d_{i(s)} - 1), t - x_i)$, $a^4_{min} = \max(x_{i(s)}, t-(N(i,s-1)-1+x_i)), a^4_{max} = \min(N(i(s), d_{i(s)}-1+x_{i(s)}), t-x_i)$.

Steps to reconstruct the proof are provided in Appendix B.

3. Bounds for the H-Polytope: PERT Networks. In this section, we consider combinatorial optimization problems with a known compact H-polytope representation. While the formulations in the previous section can be used for V-polytope representations, the complexity of the formulations depend on P and can be cumbersome in applications where P is large. It is therefore desirable to have compact formulations under known H-polytope representations. We will now show that for PERT networks represented with a H-polytope, the upper bound $U(r)$ is efficiently computable in polynomial time in n and K.

PERT networks are widely used in project planning and management across various settings such as construction projects, software planning projects and facility maintenance projects. A PERT network is denoted by a directed acyclic graph (DAG) $G = (V,E)$ where V is the set of vertices and E is the set of edges. The start node is denoted by $s \in V$ and the terminal node is denoted by $t \in V$. The arcs represent activities in the project and nodes represent events in an activity on arc framework

[13]. The network structure captures precedence relationships among the activities. Each activity (i, j) is associated with a random time duration c_{ij} to complete it. For fixed c_{ij}, $(i, j) \in E$, the completion time of the project is computed as the longest path from node s to t. This is formulated as the 0-1 integer program:

$$
\begin{aligned}
Z^{\mathrm{pert}}(\mathbf{c}) = \max \quad & \textstyle\sum_{(i,j)\in E} c_{ij} x_{ij} \\
\text{s.t} \quad & \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = \begin{cases} 1, & \text{if } i = s, \\ -1, & \text{if } i = t, \\ 0, & \text{otherwise,} \end{cases} \\
& x_{ij} \in \{0, 1\}, \quad \text{for } (i,j) \in E.
\end{aligned}
$$

The total unimodularity of the constraint matrix ensures that the LP relaxation exactly solves the integer program and $Z^{pert}(\mathbf{c})$ is polynomial time computable. There is a large stream of literature on uncertain PERT networks [46, 36] and computing the distribution and the expected value of $Z^{\mathrm{pert}}(\tilde{\mathbf{c}})$ with independent activity durations. Evaluating both the distribution and the expected value are known to be #P-hard [19] and not polynomial time computable even in the number of values that the project duration takes. Several approximations and bounds have been proposed (see [15, 12, 23]). In special cases, the computation of the distribution and the expected value are known to be possible in polynomial time with independent distributions. Specifically, for the class of series parallel graphs with activity durations supported in $[0, K]$, the worst case probability and expectation bounds can be computed in polynomial time. For more general graphs, prior works of [12, 23] have also constructed approximations by using transformations to series parallel graphs.

Applying the formulation in Theorem 2.1 requires enumeration of the P extreme points which in the setting of PERT networks, corresponds to the s-t paths in the network. The previous formulation is hence useful only when the number of s-t paths does not grow rapidly. We next propose a tight formulation that does not require the enumeration of the s-t paths. Specifically the result implies that for extremal dependence, the worst-case probability is polynomial time computable for general DAG under the assumption of restricted support in $[0, K]$ while for independent distributions, such a result is possible only for restricted graphs like series parallel graphs.

THEOREM 3.1. *Consider a PERT network $G = (V, E)$ with $|E| = n$ and s and t denoting the source and terminal nodes respectively. Given the marginal distributions of the activity duration vector $\tilde{\mathbf{c}}$ as $\mathbb{P}(\tilde{c}_{ij} = k) = p_{ijk}$ for $(i, j) \in E$, $k \in [0, K]$ and $r \in [0, nK]$, the tightest upper bound on the probability of the project completion time taking a value greater than or equal to r is the optimal value of the linear program:*

$$
\begin{aligned}
U^{pert}(r) = \max \quad & a \\
s.t. \quad & a + b = 1, \\
& \textstyle\sum_{k=0}^{K} h_{ijk} = b, \ \textit{for } (i,j) \in E, \\
& \textstyle\sum_{k=0}^{K} g_{ijk} + \sum_{k=0}^{K}\sum_{l=k}^{nK} \delta_{ij,k,l} = a, \ \textit{for } (i,j) \in E, \\
& \textstyle h_{ijk} + g_{ijk} + \sum_{l=k}^{nK} \delta_{ij,k,l} = p_{ijk}, \ \textit{for } (i,j) \in E, k \in [0, K], \\
& \textstyle a = \sum_{l=r}^{nK} \tau_l, \\
& \textstyle \tau_l = \sum_{i:(i,t)\in E} \sum_{k=0}^{\min(l,K)} \delta_{it,k,m}, \ \textit{for } l \in [r, nK], \\
& \textstyle \sum_{j:(i,j)\in E} \sum_{k=0}^{\min(K,nK-l)} \delta_{ij,k,l+k} = \sum_{j:(j,i)\in E} \sum_{k=0}^{\min(K,l)} \delta_{ji,k,l}, \\
& \qquad \textit{for } i \in V \setminus \{s, t\}, l \in [0, nK], \\
& \textstyle \sum_{i:(i,t)\in E} \sum_{k=0}^{\min(l,K)} \delta_{it,k,l} = 0, \ \textit{for } l \in [0, r-1], \\
& \textstyle \sum_{i:(s,i)\in E} \sum_{k=0}^{\min(K,nK-l)} \delta_{si,k,l+k} = 0, \ \textit{for } l \in [1, nK], \\
& a, b \geq 0, \tau_l \geq 0, \ \textit{for } l \in [r, nK], \ h_{ijk}, g_{ijk} \geq 0, \ \textit{for } (i,j) \in E, k \in [0, K], \\
& \delta_{ij,k,l} \geq 0, \ \textit{for } (i,j) \in E, k \in [0, K], l \in [k, nK].
\end{aligned}
$$

Hence, $\max_{\theta\in\Theta}\mathbb{P}_\theta(Z^{pert}(\tilde{\mathbf{c}})\geq r)=U^{pert}(r)$ *and* $U^{pert}(r)$ *is computable in time polynomial in* $|V|,|E|,K$ *and* $\log(U_1)$*.*

Proof. The approach will, as before, involve developing a compact formulation for the separation problem in (2.1). We will make use of the structure of the s-t flow polytope in order to derive the reduced formulation. Given λ and $\boldsymbol{\alpha}$, the constraint (2.1) is equivalent to:

$$\lambda+\underbrace{\min\left\{\sum_{(i,j)\in E}\sum_{k=0}^{K}\alpha_{ijk}\mathbb{1}_{\{c_{ij}=k\}}:Z^{pert}(\mathbf{c})\geq r, c_{ij}\in[0,K],\text{ for }(i,j)\in E\right\}}_{\text{Sep}(\boldsymbol{\alpha})}\geq 1. \tag{3.1}$$

This problem looks at assigning a length from the set $[0,K]$ to each edge c_{ij} where the cost of assigning length k to c_{ij} is α_{ijk}. In particular, we want to compute a minimum cost assignment of the lengths to c_{ij} in such a way that the longest path from node s to t has a length at least r. This is equivalent to ensuring the existence of a s-t path with length at least r. The costs $\boldsymbol{\alpha}_{ij}=(\alpha_{ijk};k\in[0,K])$ can be viewed as a mapping from $[0,K]$ to $\mathbb{R}$, albeit without any structural assumptions such as monotonicity, non-negativity etc. Observe that for each edge $(i,j)\in E$, we will always incur a cost of at least $q_{ij}=\min_{k\in[0,K]}\alpha_{ijk}$. We focus on minimizing the updated costs $v_{ijk}=\alpha_{ijk}-q_{ij}\geq 0$. In particular for $k^*\in\text{argmin}_{k\in[0,K]}\alpha_{ijk}$, we have $v_{ijk^*}=0$. The optimization problem $\text{Sep}(\boldsymbol{\alpha})$ in (3.1) can therefore be split up as follows:

$$\text{Sep}(\boldsymbol{\alpha})=\sum_{(i,j)\in E}q_{ij}+\left\{\begin{array}{ll}\min & \displaystyle\sum_{(i,j)\in E}\sum_{k=0}^{K}\overbrace{(\alpha_{ijk}-q_{ij})}^{v_{ijk}}\mathbb{1}_{\{c_{ij}=k\}}\\ \text{s.t} & Z^{pert}(\mathbf{c})\geq r, c_{ij}\in[0,K]\text{ for }(i,j)\in E\end{array}\right\}. \tag{3.2}$$

We will now focus on finding an assignment to $\mathbf{c}$ so as to solve the optimization problem in the second term in Equation (3.2). Observe that we want to minimize the updated costs $\mathbf{v}$ subject to the constraint $Z(\mathbf{c})\geq r$. For this, we propose a set of dynamic programming recursions as follows.

Let $f_{l,i}$ denote the best value of the objective in the optimization problem (3.2) when there exists a path from s to i with of length exactly l. The computation of $f_{l,i}$ gives a minimum cost assignment such that some path from s to i has a length of exactly l. Since a PERT network is described by a DAG, there exists an ordering of the vertices by means of a topological sort. Denote such an ordering by O_{top}. The base case of the dynamic program is given by the computation of $f_{0,s}$ for the source node s. Clearly $f_{0,s}=0$ as the assignment $c_{ij}=\text{argmin}_{k\in S}v_{ijk}$ incurs a total cost of 0 and any path from s to itself has a length of 0 trivially. Next we describe the induction step. For any node j, let the value of $f_{l,i}$ be known for all nodes i such that $(i,j)\in E$, $l\in[0,nK]$. This is possible when we fill the columns of the matrix f in the order given by O_{top}. The following relations hold,

$$f_{l,j}=\min_{i:(i,j)\in E}\min_{k\in[0,K]}\left(f_{l-k,i}+v_{ijk}\right),\text{ for }l\in[k,nK]$$

This holds since if a path of length l exists from s to j and an edge (i,j) on this path is assigned a value of k, then the path from s to i must have a length of $l-k$. The optimal value of the objective must therefore choose the minimum value generated out of all possible assignments for all incoming arcs (i,j) to node j. Finally the objective function in (3.2) requires that the assignment produces a path of length of at least r

from s to t. Let z denote the objective value of the optimization problem in (3.2). Then, $z = \min_{l\in[r,nK]} f_{l,t}$. Putting all the dynamic programming recursions together,

$$
\begin{aligned}
\mathrm{Sep}(\boldsymbol{\alpha}) = \max_{\mathbf{q},\mathbf{f},z} \quad & \sum_{(i,j)\in E} q_{ij} + z \\
\text{s.t.} \quad & q_{ij} \le \alpha_{ijk},\ \text{for } (i,j)\in E, k\in[0,K], \\
& f_{0,s} = 0,\ z \le f_{l,t},\ \text{for } l\in[r,nK], \\
& f_{l,j} \le f_{l-k,i} + \alpha_{ijk} - q_{ij},\ \text{for } (i,j)\in E, k\in[0,K], l\in[k,nK].
\end{aligned}
$$

Now, forcing $\lambda + \mathrm{Sep}(\boldsymbol{\alpha}) \ge 1$ gives a set of linear inequalities reformulating (2.1). Constraint (2.2) can be reformulated in the same manner as in proof of Theorem 2.1. Combining the reformulations of (2.1) and (2.2) gives us,

$$
\begin{aligned}
\min \quad & \lambda + \sum_{(i,j)\in E}\sum_{k\in[0,K]} \alpha_{ijk} p_{ijk} \\
\text{s.t.} \quad & \lambda + \sum_{(i,j)\in E} d_{ij} \ge 0,\ \alpha_{ijk} - d_{ij} \ge 0,\ \text{for } (i,j)\in E,\ \text{for } k\in[0,K], \\
& \lambda + z + \sum_{(i,j)\in E} q_{ij} \ge 1,\ \alpha_{ijk} - q_{ij} \ge 0,\ \text{for } (i,j)\in E, k\in[0,K], \\
& f_{0,s} = 0,\ f_{l,t} - z \ge 0,\ \text{for } l\in[r,nK], \\
& f_{l-k,i} + \alpha_{ijk} - q_{ij} - f_{l,j} \ge 0,\ \text{for } (i,j)\in E, k\in[0,K], l\in[k,nK].
\end{aligned}
$$

Further taking the dual of this linear program gives us the formulation in the theorem.

The LP thus obtained with $O(|E|^2K^2 + |V||E|K)$ constraints and $O(|E|^2K^2)$ variables is therefore polynomially sized in $|V|$, $|E|$ and K. The maximum value from the input univariate distributions p_{ijk} requires $\log_2(U_1)$ bits for representation. Hence the linear program is solvable in time polynomial $|V|, |E|, K$ and $\log_2(U_1)$. ∎

The techniques used in deriving Theorem 2.1 and Theorem 3.1 rely on dynamic programming. However by making further use of the problem structure, we are able to obtain a further reduced formulation in Theorem 3.1 for PERT networks.

4. Numerical Results.

We will now provide numerical results from various formulations. The following formulations were implemented using gurobipy [18].

(a) Upper bound $U(r)$: Tightest upper bounds via the LPs in Theorems 2.1 and 3.1.

(b) Markov bound: Markov's inequality gives us an upper bound for any distribution $\theta \in \Theta$ and positive value of r as, $\mathbb{P}_\theta(Z(\tilde{\mathbf{c}}) \ge r) \le \min\left(\max_{\theta\in\Theta} \mathbb{E}_\theta[Z(\tilde{\mathbf{c}})]/r, 1\right)$. To compute the maximum expected value when $\mathcal{X}$ is represented with a V-polytope, we can use existing results in the literature [29] to get:

$$
\begin{aligned}
\max_{\theta\in\Theta} \mathbb{E}_\theta[Z(\tilde{\mathbf{c}})] = \max_{\boldsymbol{\gamma},\boldsymbol{\lambda}} \quad & \sum_{i=1}^{n}\sum_{k=0}^{K} k\gamma_{ik} p_{ik} \\
\text{s.t.} \quad & \sum_{\mathbf{x}\in\mathcal{X}} \lambda_{\mathbf{x}} = 1,\ \lambda_{\mathbf{x}} \ge 0\ \text{ for } \mathbf{x}\in\mathcal{X}, \\
& \sum_{k=0}^{K} p_{ik}\gamma_{ik} = \sum_{\mathbf{x}\in\mathcal{X}: x_i = 1} \lambda_{\mathbf{x}}, \\
& 0 \le \gamma_{ik} \le 1\ \text{ for } i\in[n].
\end{aligned}
$$

where the random variables $\tilde{c}_i \in [0,K]$ with $p_{ik} = \mathbb{P}(\tilde{c}_i = k)$ for $k\in[0,K]$ and $i\in[n]$.

(c) Independence: To compute $I(r) = \mathbb{P}_{\theta_{ind}}(Z(\tilde{\mathbf{c}} \ge r)$, we approximate the probability using a simulation of 10000 runs.

(d) Distribution maximizing $\mathbb{E}[Z(\tilde{\mathbf{c}}) - r)]^+$: A formulation that computes this maximum expectation can be derived using the techniques in [29, 11, 45] as,

$$
(4.1)\quad
\begin{aligned}
\max \mathbb{E}[Z(\tilde{\mathbf{c}}) - r]^+ = \max \quad & \sum_{(i,j)\in E}\sum_{k=0}^{K} k g_{ijk} p_{ijk} - r \sum_{\mathbf{x}\in\mathcal{X}} \lambda_{\mathbf{x}} \\
\text{s.t.} \quad & \sum_{\mathbf{x}\in\mathcal{X}} \lambda_{\mathbf{x}} \le 1,\ \lambda_{\mathbf{x}} \ge 0,\ \text{ for } \mathbf{x}\in\mathcal{X}. \\
& g_{ijk} \le 1,\ \text{ for } (i,j)\in E, k\in[0,K], \\
& \sum_{\mathbf{x}\in\mathcal{X}: x_{ij}=1} \lambda_{\mathbf{x}} = \sum_{k=0}^{K} g_{ijk} p_{ijk},\ \text{ for } (i,j)\in E, \\
& g_{ijk} \ge 0,\ \text{ for } (i,j)\in E, k\in[0,K].
\end{aligned}
$$

The term $\sum_{\mathbf{x}\in\mathcal{X}} \lambda_{\mathbf{x}}$ gives us $\mathbb{P}(Z(\tilde{\mathbf{c}}) \ge r)$ for the extremal distribution maximizing $\mathbb{E}[Z(\tilde{\mathbf{c}}) - r]^+$. We refer to this probability bound as 'Worst Exp' in all the plots.

4.1. Sums of Random Variables with Limited Dependence.

We first provide a numerical application of the weighted probability bounds to the sums of random variables by allowing for a limited degree of dependence. This is achieved by considering a split of the set of random variables into two sets - one allowing for extremal dependence among the variables and the other set containing mutually independent variables. The random variables across the two sets are assumed independent of each other. Specifically let $P(\tilde{\alpha}_i = 1) = 1 - P(\tilde{\alpha}_i = 0) = p_i$ for $i \in [n_1]$ and $P(\tilde{\beta}_j = 1) = 1 - P(\tilde{\beta}_j = 0) = q_j$ for $j \in [n_2]$. The dependence among random variables in $\tilde{\boldsymbol{\alpha}}$ is not specified while the random variables in $\tilde{\boldsymbol{\beta}}$ are mutually independent. The two sets of random variables are also independent of each other. Under this model, we will see that the bound on the tail probability of the sum of random variables can be reformulated using the weighted probability bound in Theorem 2.6 where the weights are appropriately computed.

Given $r \in [0, n_1 + n_2]$, let the tightest upper bound on the tail probability be:

$$
(4.2)\quad \overline{S}(r,1) = \max_{\theta\in\Theta_\ell} \mathbb{P}_\theta\left(\sum_{i=1}^{n_1} \tilde{\alpha}_i + \sum_{j=1}^{n_2} \tilde{\beta}_j \ge r\right),
$$

where Θ_ℓ is the set of distributions consistent with the given assumptions:

$$
\begin{aligned}
\Theta_\ell = \big\{\theta \in \mathbb{P}(\{0,1\}^{n_1+n_2}) : \mathbb{P}_\theta(\boldsymbol{\alpha},\boldsymbol{\beta}) = \mathbb{P}_\theta(\boldsymbol{\alpha})\,\mathbb{P}_{\theta_{ind}}(\boldsymbol{\beta}), \quad & \text{for } (\boldsymbol{\alpha},\boldsymbol{\beta}) \in \{0,1\}^{n_1+n_2}, \\
\mathbb{P}_\theta(\tilde{\alpha}_i = 1) = p_i, \quad & \text{for } i \in [n_1]\big\},
\end{aligned}
$$

where θ_{ind} is the product distribution for the independent variables in $\tilde{\boldsymbol{\beta}}$ supported on $\{0,1\}^{n_2}$. We refer to this as the "limited dependency" model. The probability can be rewritten as:

$$
\mathbb{P}_\theta\left(\sum_{i=1}^{n_1} \tilde{\alpha}_i + \sum_{j=1}^{n_2} \tilde{\beta}_j \ge r\right) = \sum_{\ell=0}^{n_2}\left[\mathbb{P}_{\theta_\alpha}\big(\sum_{i=1}^{n_1} \tilde{\alpha}_i \ge r-\ell\big)\mathbb{P}_{\theta_{ind}}\big(\sum_{j=1}^{n_2} \tilde{\beta}_j = \ell\big)\right],
$$

where θ_α is a feasible distribution of the random vector $\tilde{\boldsymbol{\alpha}}$ corresponding to θ and,

$$
\mathbb{P}_{\theta_\alpha}(\boldsymbol{\alpha}) = \sum_{\boldsymbol{\beta}\in\{0,1\}^{n_2}} \mathbb{P}_\theta((\boldsymbol{\alpha},\boldsymbol{\beta})), \quad \forall \boldsymbol{\alpha} \in \{0,1\}^{n_1}
$$

In this case, it is possible to compute the probabilities $\mathbb{P}_{\theta_{ind}}\big(\sum_{j=1}^{n_2}\tilde{\beta}_j = \ell\big),\ \ell \in [0, n_2]$ in polynomial time using dynamic programming recursion [9]. We can then reformulate (4.2) as follows:

(4.3)

$$\max_{\theta\in\Theta_\ell} \mathbb{P}_\theta\bigg(\sum_{i=1}^{n_1}\tilde{\alpha}_i + \sum_{j=1}^{n_2}\tilde{\beta}_j \ge r\bigg) = \max_{\theta_\alpha\in\Theta}\sum_{\ell=0}^{n_2}\Big[\mathbb{P}_{\theta_\alpha}\big(\sum_{i=1}^{n_1}\tilde{\alpha}_i \ge r-\ell\big)\mathbb{P}_{\theta_{ind}}\big(\sum_{j=1}^{n_2}\tilde{\beta}_j = \ell\big)\Big],$$

$$\Theta = \{\theta_\alpha \in \mathbb{P}(\{0,1\}^{n_1}) \ : \ \mathbb{P}_{\theta_\alpha}(\tilde{\alpha}_i = 1) = p_i, \ \text{ for } i \in [n_1]\}\,.$$

is the set of distributions defined on $\{0,1\}^{n_1}$ and consistent with the given marginal information for $\tilde{\boldsymbol{\alpha}}$. By rewriting the tail probabilities as:

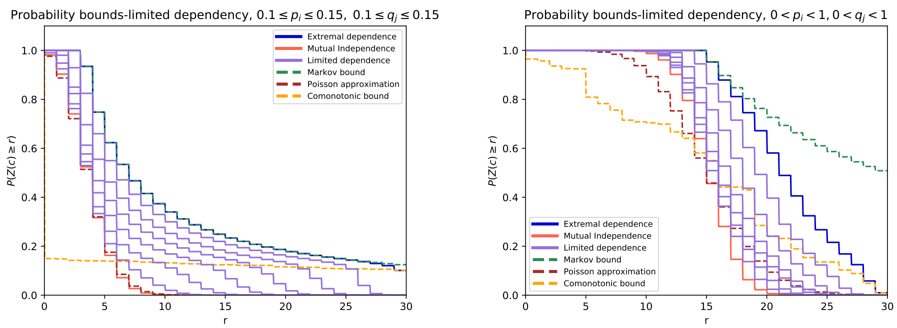
$$\mathbb{P}_{\theta_\alpha}\big(\sum_{i=1}^{n_1}\tilde{\alpha}_i \ge r-\ell\big) = \sum_{t=r-\ell}^{n_1}\mathbb{P}_{\theta_\alpha}\big(\sum_{i=1}^{n_1}\tilde{\alpha}_i = t\big),$$

we can cast (4.3) in the form of a weighted probability function similar to that in (2.6) with n_1 decision variables and weights $w_\ell = \mathbb{P}_{\theta_{ind}}\big(\sum_{j=1}^{n_2}\tilde{\beta}_j = \ell\big)$, for $\ell \in [0, n_2]$. The compact linear program (2.5) can now be used to compute the tight bound.

In this model, when $n = n_1$ and $n_2 = 0$, all the random variables are extremally dependent and the tight bound $S(r,1)$ is retrieved. Similarly, when $n = n_2$ and $n_1 = 0$, all the random variables are mutually independent and the tail probability bound $I(r,1)$ is retrieved. We also consider a Poisson approximation to the sum of Bernoulli random variables. [27] showed that the Poisson distribution can be used to approximate the probability distribution of sums of independent but not necessarily identical Bernoulli random variables, where the error of the approximation is small when the probabilities are small. The method from [40],[8] extends this idea and develops error bounds for the Poisson approximation of the distribution of sums of weakly dependent Bernoulli variables. We compare the limited dependency bounds computed from the compact linear program (2.5) with the two extremes of extremal dependence and complete independence and three other probabilities computed using a Poisson approximation, a comonotonic bound computed with perfectly dependent random variables and the Markov bound. Figure 3 shows the six bounds for $n = 30$ variables where the limited dependency bounds (in purple) have been selectively shown for $n_1 = 6, 10, 14, 18, 22, 26$ (left to right). In Figure 3a, we consider non-identical small marginal probabilities by uniformly and independently generating the them from $[0.1, 0.15]$ while in Figure 3b, we uniformly generate the probabilities from $[0, 1]$.

The Poisson approximation closely follows the independent tail probability $I(r,1)$ in Figure 3a as the theory suggests with the assumption of small probabilities while in Figure 3b, it initially underestimates the independent tail probability (for $r \le 15$) and then overestimates it. Due to the almost identical nature of the small probabilities in Figure 3a, the comonotonic bound plot remains almost flat for $r \ge 1$ and the Markov bound is very close to the extremally dependent bound $S(r,1)$ while this is not true in Figure 3b due to the non-identical probabilities. The results indicate that the LP approach can appropriately incorporate both independence and dependence considerations in computing the extremal tail probability bounds.

We next illustrate the usefulness of the limited dependency bound $\overline{S}(r,1)$ in (4.2) in applications where distributions of sums of dependent random variables are used by comparing it to the Poisson approximation. Denote the Poisson approximation on the tail probability of the sum of n Bernoulli random variables by,



(a) Small range of marginal probabilities (b) Larger range of marginal probabilities

Fig. 3: Step plots of upper bounds for $n_1 + n_2 = 30$

$S_P(r, 1) = \sum_{k=r}^{n} e^{-\lambda} \dfrac{\lambda^k}{k!}$. While several bounds on the error of the Poisson approximation have been proposed in the literature (see [31] for an overview), these bounds are not necessarily tight and typically become weaker with increase in the number of dependent variables (degree of dependence) and the marginal probabilities. However, since we can compute the exact bound $\overline{S}(r, 1)$, we can also compute the precise error of the Poisson approximation for given level of dependence n_1. A popular metric to measure the distance between two probability measures is the Kolmogorov distance (also called uniform distance) which is defined as :

$$d_K\left(F_X;\ F_Y\right) = \sup_x |F_X(x) - F_Y(y)|,$$

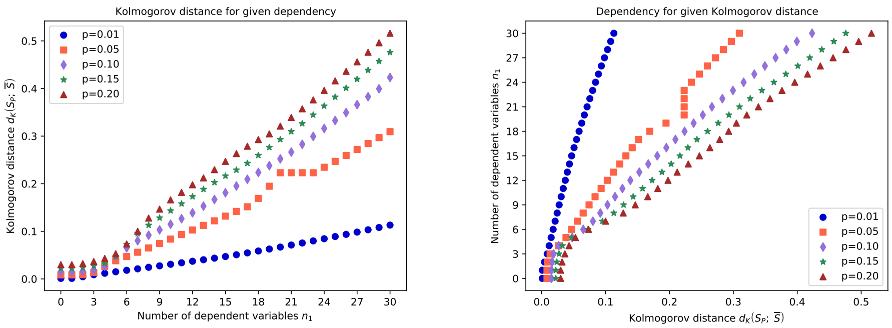
where $F_X(x)$, $F_Y(y)$ are the CDF's of the two distributions. Note that because of the absolute value used in the definition, we can use the right tail probability functions instead of the CDF's and thus compute the Kolmogorov distance between the Poisson and limited dependence distributions (for each level of dependence n_1) as follows :

$$d_K\left(S_P;\ \overline{S}\right) = \max_{r \in [0,n]} |S_P(r, 1) - \overline{S}(r, 1)|.$$

When the variables are identical and mutually independent (note that $n_1 = 0$ and $n_1 = 1$ are equivalent mutually independent cases), there are known bounds on the Kolmogorov distance between the Poisson and Binomial distributions. [41] proves an upper bound: $d_K\left(S_P(n, p);\ B(n, p)\right) \le p\pi^2 e^{2p(2-p)}/(16(1-p))$. However our observations show that this bound is weaker with increasing probabilities.

We next consider a numerical experiment with $n_1 + n_2 = 30$ Bernoulli random variables and identical probabilities p. Figure 4a plots the Kolmogorov distance from (4.1) for each $n_1 \in [0, 30]$ and $p = [0.01, 0.05, 0.1, 0.15, 0.2]$ while Figure 4b plots the numerical inverse of the first plot.

It is clear from the Figure 4a that the Kolmogorov distance increases monotonically with the number of dependent variables n_1. This concurs with the theoreticcal understanding that the Poisson approximation performs better with mutual independence and weak dependence. For the same level of dependence, as p increases, the Kolmogorov distance increases for the range of $p \in [0.01, 0.2]$ considered here. On the



(a) Kolmogorov distance for given dependency (b) Dependency for given Kolmogorov distance

Fig. 4: Scatter plots for the case of identical probabilities with $n_1 + n_2 = 30$

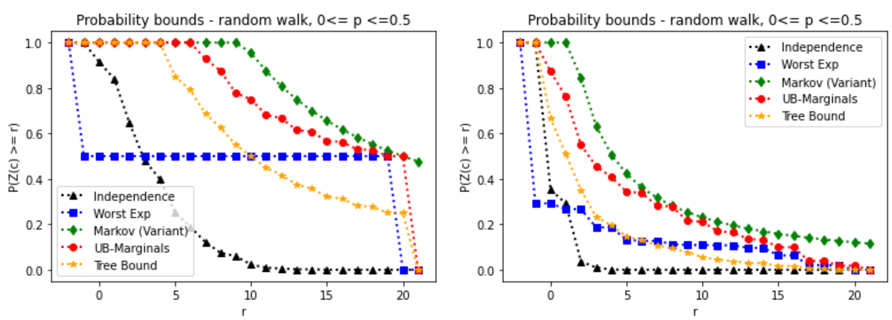
other hand, for a given error, the dependence level which can be accomodated within the specified error decreases with increasing p within the plotted range. However, our observation shows that this monotonicity need not hold for larger values of p. From Figure 4b, we can compute a desired level of dependence for a given error of the Poisson approximation (measured in terms of the Kolmogorov distance). This can provide useful information in applications where Poisson approximations for distributions of sums of dependent random variables are used (see [1] for a non-exhaustive list of examples) and the error tolerance is pre-specified. For example, this approach can be used by portfolio managers to recommend partially dependent stocks (where some stocks are dependent but the exact correlation is not specified and others are independent) based on the degree of risk aversion of the investors.

4.2. Random Walk: V-Polytope. We now consider the maximum of partial sums of random variables, a problem arising from applications in random walks. Consider a random vector $\tilde{\mathbf{c}}$ of size n and let $Z^{rw}(\tilde{\mathbf{c}}) = \max(\tilde{c}_1, \tilde{c}_1+\tilde{c}_2, \ldots, \sum_{i=1}^{n} \tilde{c}_i)$, where $\tilde{c}_i \in \{-1, 1\}$ for all $i \in [n]$. The tail behaviour of this quantity has been extensively studied (see [2]) and is of interest in settings such as risk and queueing theory. For example, when $n \to \infty$ and the random variables are mutually independent, the Lundberg inequality (see [3]) gives the tail probability bound, $\mathbb{P}_{\theta_{ind}}(Z^{rw}(\tilde{\mathbf{c}}) \geq r) \leq e^{-h_0 r}$, where h_0 is parameter dependent on the moment generating function of the distribution of $\tilde{\mathbf{c}}$. Several approximations for the distribution of $Z^{rw}(\tilde{\mathbf{c}})$ have been developed for the finite n case (see [10, 24]) using the marginal distributions. Here we consider the bounds on the tail probability with extremal dependence.

Let $U_{rw}(r)$ denote the maximum value of the tail probability over all joint distributions consistent with the given marginal distributions, $U^{rw}(r) = \max_{\theta\in\Theta} \mathbb{P}(Z^{rw}(\tilde{\mathbf{c}}) \geq r)$. Figure 5a illustrates the probability bounds for the case of identical probabilities with $p_i = 0.5$ for all $i \in [n]$. 'Tight UB' refers to the bound $U^{rw}(r)$. While the Markov bound applies to only non-negative random variables, in the random walk application considered, $Z^{rw}(\mathbf{c}) \in [-1, n]$. We therefore use the following variant,

$$\mathbb{P}(Z^{rw}(\tilde{\mathbf{c}}) \geq r) = \mathbb{P}(Z^{rw}(\tilde{\mathbf{c}}) + 1 \geq r + 1) \leq \min\left(\frac{\max_{\theta\in\Theta} \mathbb{E}_\theta[Z^{rw}(\tilde{\mathbf{c}})] + 1}{r+1}, 1\right).$$

We observe that the Markov bound is not a tight upper bound for this application. The probability bound 'Worst exp' refers to the bound attained by the comonotone distribution here (since $Z^{rw}(\mathbf{c})$ is a supermodular function and the comonotone distribution maximizes expectation of supermodular functions) so that $\mathbb{P}(\tilde{c}_1 = 1, \ldots, \tilde{c}_n = 1) = 0.5$ and $\mathbb{P}(\tilde{c}_1 = -1, \ldots, \tilde{c}_n = -1) = 0.5$. The tight upper bound labelled 'Tight UB' gives $U^{rw}(r)$ and is attained by a different distribution from the comonotone distribution. We also illustrate the tree bound using formulation in Theorem 2.10 for the case where the underlying bivariate distributions are specified as follows: $\mathbb{P}(\tilde{c}_i = 1, \tilde{c}_{i+1} = 1) = \mathbb{P}(\tilde{c}_i = 1)\mathbb{P}(\tilde{c}_{i+1} = 1)\, \forall i \in [1, n-1]$. Thus the bivariate distributions are known on a series graph in Theorem 2.10. We observe that the tree bound brings substantial reduction in the bound $U^{rw}(r)$ and is closer to independence.
Similar trends are observed for the case of non-identical probabilities in Figure 5b.



(a) The case of identical probabilities, $p_i = 0.5$.

(b) A case of non-identical probabilities, $p_i \leq 0.5$.

Fig. 5: Probability bounds for the random walk application.

4.3. PERT Networks. We now discuss our numerical results in PERT networks. We compute $U^{pert}(r)$ using the LP in Theorem 3.1. In the plots, $U^{pert}(r)$ is denoted by the label 'Tight UB'. The Markov bound evaluates to $\min(\max \mathbb{E}[Z(\tilde{\mathbf{c}})]/r, 1)$ where the maximum possible expectation bound is computed in polynomial time in the size of the graph using the below tight formulation from [29].

$$
\begin{aligned}
\max \mathbb{E}[Z(\tilde{\mathbf{c}})] = \min_{\mathbf{y},\mathbf{d},\mathbf{u}} \quad & u_s + \sum_{(i,j)\in E} \sum_{k=1}^{K} p_{ijk} y_{ijk} \\
\text{s.t} \quad & u_i - u_j \geq d_{ij}, \ \text{for } (i,j) \in E, \\
& u_t = 0, \ y_{ijk} \geq k - d_{ij}, \ \text{for } (i,j) \in E, \ \text{for } k \in [0, K], \\
& \mathbf{y} \geq 0, \mathbf{d}, \mathbf{u} \text{ unrestricted.}
\end{aligned}
$$

Formulation (4.1) is used to obtain the tail probability from a distribution that maximizes $\mathbb{E}[Z(\tilde{\mathbf{c}}) - r]^+$, where $\mathcal{X}$ denotes the set of s-t paths for PERT networks.

The network in Figure 6 with $n = 24$ nodes and a total of 29 edges or activities is considered. There are a total of 14 paths from s to t. The longest path from s to t contains 10 edges and hence the maximum possible completion time of the project is $10K$, where K is the maximum possible duration of each of the activities. This network was presented in [6, 5] where the worst case bounds for the expected time of completion was computed. We take $K = 10$ and for all edges (i,j), $p_{ijk} = 1/(K+1)$.

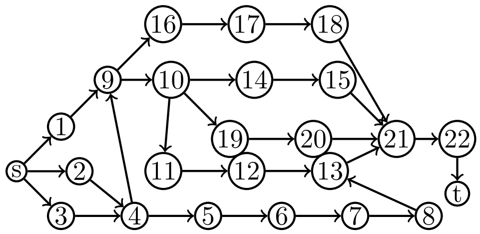
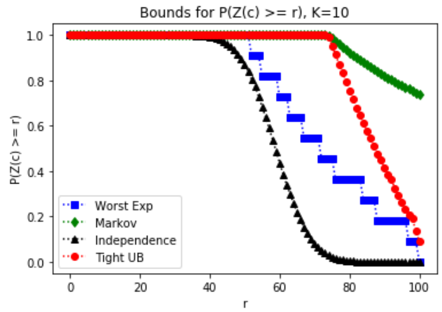
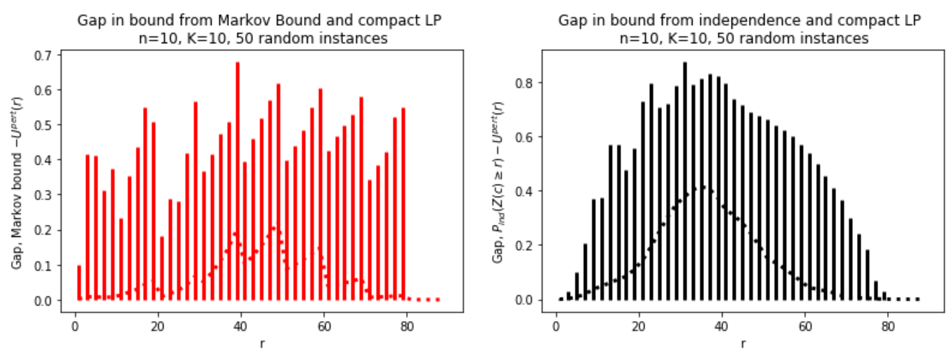


Fig. 6: Example 1



The Markov bound is not tight for this example while the gaps from independence and worst exp demonstrate significant gap with the tight bound. Here, the worst exp curve is closer to Tight UB than independence. However the distribution maximizing the worst case expectation does not maximize the tail probability.

4.3.1. Comparison of Bounds on Randomly Generated Instances. We now compare our bounds against the Markov bound and the bound from the independent distribution for a set of 50 randomly generated graphs and univariate marginals on $n = 10$ nodes with $K = 10$. In Figure 7a, we report the gap $M(r) - U^{pert}(r)$ for various values of r where $M(r)$ represents the Markov bound. The bars indicate the range between the minimum and maximum gaps while the dotted line provides the mean gap. Observe that the Markov bounds are not tight in general and always provide an upper bound for $U^{pert}(.)$. In Figure 7b, we report the gap $U^{pert}(r) - \mathbb{P}_{\theta_{ind}}(Z(\tilde{\mathbf{c}} \geq r)$ where θ_{ind} denotes the independent distribution. The independent distribution serves as lower bound for $U^{pert}(r)$ and is clearly not an extremal distribution.

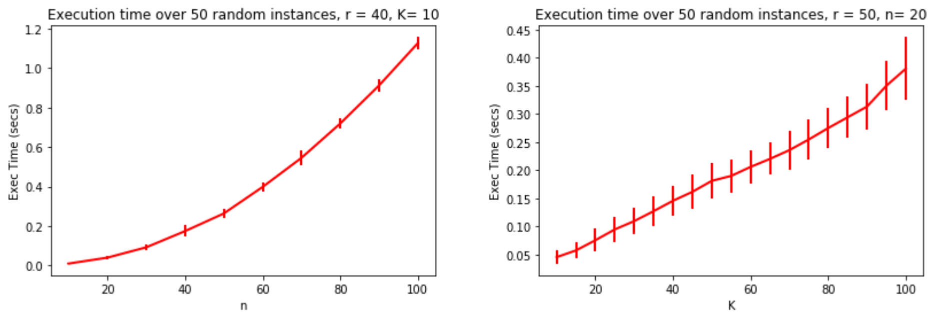


(a) Gap in Markov bound

(b) Gap in independence

Fig. 7: Comparison of gaps in various bounds over 50 randomly generated instances.

4.3.2. Computational Times. We now report the computational times of our compact linear program as a function of the number of nodes n as well as a function of K. Figure 8a shows the error bars of the execution time as a function of n, over 50 random instances with $r = 40$ and K fixed to 10. Even for $n = 100$ nodes, the execution time is about 1.2 seconds on an average. We performed the experiment for various values of $r \in \{10, \dots, 50\}$, however we did not observe significant difference in the results. In Figure 8b, we provide the error bars of the execution time as a function



(a) (b)

Fig. 8: Execution times of our compact linear program

of K, with $r = 50$ and $n = 20$. Over all instances, our compact LP takes a maximum of 0.45 seconds even when the support for the activity durations goes till $K = 100$.

Acknowledgements. The research of the fourth author was partly supported by MOE Academic Research Fund Tier 2 grant T2MOE1906, "Enhancing Robustness of Networks to Dependence via Optimization".

REFERENCES

[1] R. Arratia, L. Goldstein, and L. Gordon, *Poisson approximation and the Chen-Stein method*, Statistical Science, (1990), pp. 403–424.

[2] S. Asmussen, *Ruin probabilities*, in Advanced series on statistical science and applied probability, 2001.

[3] S. Asmussen and T. Rolski, *Risk theory in a periodic environment: The Cramér-Lundberg approximation and Lundberg's inequality*, Mathematics of Operations Research, 19 (1994), pp. 410–433.

[4] M. O. Ball, C. J. Colbourn, and J. S. Provan, *Chapter 11 network reliability*, in Network Models, vol. 7 of Handbooks in Operations Research and Management Science, Elsevier, 1995, pp. 673 – 762.

[5] D. Bertsimas, K. Natarajan, and C.-P. Teo, *Probabilistic combinatorial optimization: Moments, semidefinite programming, and asymptotic bounds*, SIAM Journal on Optimization, 15 (2004), pp. 185–209.

[6] J. R. Birge and M. J. Maddox, *Bounds on expected project tardiness*, Operations Research, 43 (1995), pp. 838–850.

[7] J. Blanchet, H. Lam, Y. Liu, and R. Wang, *Convolution bounds on quantile aggregation*, 2021, https://arxiv.org/abs/2007.09320.

[8] L. H. Chen, *Poisson approximation for dependent trials*, The Annals of Probability, 3 (1975), pp. 534–545.

[9] S. X. Chen and J. S. Liu, *Statistical applications of the Poisson-binomial and conditional Bernoulli distributions*, Statistica Sinica, (1997), pp. 875–892.

[10] K. L. Chung, *On the maximum partial sums of sequences of independent random variables*, Transactions of the American Mathematical Society, 64 (1948), pp. 205–233.

[11] X. V. Doan and K. Natarajan, *On the complexity of nonoverlapping multivariate marginal bounds for probabilistic combinatorial optimization problems*, Operations Research, 60 (2012), pp. 138–149.

[12] B. Dodin, *Bounding the project completion time distribution in PERT networks*, Operations Research, 33 (1985), pp. 862–881.

[13] S. E. Elmaghraby, *Activity networks: Project planning and control by network models*, John Wiley & Sons, 1977.

[14] P. Embrechts and G. Puccetti, *Bounds for functions of dependent risks*, Finance and

Stochastics, 10 (2006), pp. 341–352.
[15] D. R. Fulkerson, *Expected critical path lengths in PERT networks*, Operations Research, 10 (1962), pp. 808–817.
[16] M. R. Garey and D. S. Johnson, *Computers and intractability*, vol. 174, freeman San Francisco, 1979.
[17] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, vol. 2, Springer Science & Business Media, 2012.
[18] L. Gurobi Optimization, *Gurobi optimizer reference manual*, 2020, http://www.gurobi.com.
[19] J. N. Hagstrom, *Computational complexity of PERT problems*, Networks, 18 (1988), pp. 139–147.
[20] T. Hailperin, *Best possible inequalities for the probability of a logical function of events*, *The American Mathematical Monthly*, 72 (1965), pp. 343–359.
[21] G. A. Hanasusanto, V. Roitch, D. Kuhn, and W. Wiesemann, *Ambiguous joint chance constraints under mean and dispersion information*, Operations Research, 65 (2017), pp. 751–767.
[22] J. Kleinberg, Y. Rabani, and E. Tardos, *Allocating bandwidth for bursty connections*, SIAM Journal on Computing, 30 (2000), pp. 191–217.
[23] G. B. Kleindorfer, *Bounding distributions for a stochastic acyclic network*, Operations Research, 19 (1971), pp. 1586–1601.
[24] D. Korshunov, *On distribution tail of the maximum of a random walk*, Stochastic Processes and their Applications, 72 (1997), pp. 97–103.
[25] V. Kreinovich and S. Ferson, *Computing best-possible bounds for the distribution of a sum of several variables is np-hard*, International Journal of Approximate Reasoning, 41 (2006), pp. 331 – 342.
[26] W. Kuo and M. J. Zuo, *Optimal Reliability Modeling: Principles and Applications*, John Wiley & Sons, 2003.
[27] L. Le Cam, *An approximation theorem for the poisson binomial distribution.*, Pacific Journal of Mathematics, 10 (1960), pp. 1181–1197.
[28] G. D. Makarov, *Estimates for the distribution function of a sum of two random variables when the marginal distributions are fixed*, Theory of Probability & Its Applications, 26 (1981), pp. 803–806.
[29] I. Meilijson and A. Nádas, *Convex majorization with an application to the length of critical paths*, Journal of Applied Probability, 16 (1979), pp. 671–677.
[30] R. H. Möhring, *Scheduling under Uncertainty: Bounding the Makespan Distribution*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 79–97.
[31] S. Novak, *Poisson approximation*, Probability Surveys, 16 (2019), pp. 228–276.
[32] D. Padmanabhan and K. Natarajan, *Tree bounds for sums of bernoulli random variables: A linear optimization approach*, INFORMS Journal of Optimization, (2021).
[33] I. Pitowsky, *Correlation polytopes: Their geometry and complexity*, Mathematical Programming, 50 (1991), pp. 395–414, https://doi.org/10.1007/BF01594946.
[34] G. Puccetti and L. Rüschendorf, *Computation of sharp bounds on the distribution of a function of dependent risks*, Journal of Computational and Applied Mathematics, 236 (2012), pp. 1833 – 1840.
[35] L. J. Ringer, *A statistical theory for PERT in which completion times of activities are interdependent*, Management Science, 17 (1971), pp. 717–723.
[36] E. Roos and D. den Hertog, *A distributionally robust analysis of the program evaluation and review technique*, European Journal of Operational Research, 291 (2021), pp. 918–928.
[37] B. Ruger, *Das maximale Signifikanzniveau des Tests Lehne Ho ab, wenn k unter n gegebenen Tests zur Ablehnungfuhren*, *Metrika*, 25 (1978), pp. 171–178.
[38] L. Rüschendorf, *Mathematical Risk Analysis: Dependence, Risk Bounds, Optimal Allocations and Portfolios*, Springer Series in Operations Research and Financial Engineering, Springer, Second ed., 2013.
[39] L. Rüschendorf, *Random variables with maximum sums*, Advances in Applied Probability, 14 (1982), pp. 623–632.
[40] C. Stein, *A bound for the error in the normal approximation to the distribution of a sum of dependent random variables*, in Proceedings of the sixth Berkeley symposium on mathematical statistics and probability, volume 2: Probability theory, University of California Press, 1972, pp. 583–602.
[41] I. Tsaregradskii, *On uniform approximation of the binomial distribution by infinitely divisible laws*, Theory of Probability & Its Applications, 3 (1958), pp. 434–438.
[42] B. Wang and R. Wang, *Joint mixability*, Mathematics of Operations Research, 41 (2016), pp. 808–826.

[43] R. Wang, *Asymptotic bounds for the distribution of the sum of dependent random variables*, Journal of Applied Probability, 51 (2014), pp. 780–798.
[44] R. Wang, L. Peng, and J. Yang, *Bounds for the sum of dependent risks and worst value-at-risk with monotone marginal densities*, Finance and Stochastics, 17 (2013), pp. 395–417.
[45] G. Weiss, *Stochastic bounds on distributions of optimal value functions with applications to PERT, network flows and reliability*, Operations Research, 34 (1986), pp. 595–605.
[46] W. Wiesemann, *Optimization of temporal networks under uncertainty*, vol. 10, Springer Science & Business Media, 2012.
[47] W. Xie, S. Ahmed, and R. Jiang, *Optimized bonferroni approximations of distributionally robust joint chance constraints*, Mathematical Programming, (2019), pp. 1–34.
[48] E. Zemel, *Polynomial algorithms for estimating network reliability*, Networks, 12 (1982), pp. 439–452.

Appendix A. Proof of Corollary 2.2. The constraints on δ in the LP ensure that the above distribution is feasible and the constraints involving τ_{lx}, δ_{nklx} and $a_{\mathbf{x}}$ ensure that if $\tilde{z} = 1$, the generated realization $\mathbf{c}$ and $\mathbf{x}$ will have $\sum_{i=1}^{n} c_i x_i \geq r$.

Consistency. We will first prove that the distribution constructed is a feasible distribution and lies in Θ. We will prove this using induction.

$$\begin{aligned}
\mathbb{P}(\tilde{c}_1 = k) &= \mathbb{P}(\tilde{z} = 1) \sum_{\mathbf{x} \in \mathcal{X}} \mathbb{P}(\tilde{\mathbf{x}} = \mathbf{x} | \tilde{z} = 1) \mathbb{P}(\tilde{c}_1 = k | \mathbf{x}, \tilde{z} = 1) + \mathbb{P}(\tilde{z} = 0) \mathbb{P}(\tilde{c}_1 = k | \mathbf{x}, \tilde{z} = 0) \\
&= (\sum_{\mathbf{x} \in \mathcal{X}} a_{\mathbf{x}}) \sum_{\mathbf{x} \in \mathcal{X}} a_{\mathbf{x}} \frac{\delta_{1kkx}}{\sum_{k'} \delta_{1k'k'x}} x_1 + \frac{\delta_{1k0x}}{\sum_{k'} \delta_{1k'0x}} (1 - x_1) + b h_{ik}/b \\
&= h_{ik} + \delta_{1kkx} x_1 + \delta_{1k0x}(1 - x_1) = p_{1k}.
\end{aligned}$$

The last but one inequality holds since for any $\mathbf{x} \in \mathcal{X}$, if $x_1 = 1$, then the constraints in the LP ensure that, $\sum_k \delta_{1kkx} = \sum_{l=r}^{nK} \sum_{k'} \delta_{nk'lx} = \sum_{l=r}^{nK} \tau_{lx} = a_{\mathbf{x}}$ and, $\sum_k \delta_{1k0x} = \sum_{l=r}^{nK} \sum_{k'} \delta_{nk'lx} = \sum_{l=r}^{nK} \tau_{lx} = a_{\mathbf{x}}$, if $x_1 = 0$. Also, for all $l \in [0, nK]$, we have $\mathbb{P}(\tilde{c}_1 x_1 = l, \tilde{z} = 1, \tilde{\mathbf{x}} = \mathbf{x}) = \sum_{k' \in [0,K]} \delta_{1k'lx}$ if $x_1 = 1$ and, $\mathbb{P}(\tilde{c}_1 x_1 = 0, \tilde{z} = 1, \tilde{\mathbf{x}} = \mathbf{x}) = \sum_{k' \in [0,K]} \delta_{1k'0x}$ if $x_1 = 0$. We will now show that for any $l \in [0, nK]$,$\mathbb{P}(\sum_{j=1}^{i+1} \tilde{c}_j x_j = l, \tilde{z} = 1, \tilde{\mathbf{x}} = \mathbf{x}) = \sum_{k=0}^{K} \delta_{i+1,k,l,\mathbf{x}}$ via induction. The base case has already been shown for $i = 1$. Now assume, $\mathbb{P}(\sum_{j=1}^{i} \tilde{c}_j x_j = l, \tilde{z} = 1, \tilde{\mathbf{x}} = \mathbf{x}) = \sum_{k' \in [0,K]} \delta_{1k'lx}$ where $1 < i < n$. Then,

$$\begin{aligned}
\mathbb{P}(\sum_{j=1}^{i+1} \tilde{c}_j x_j = l, \tilde{z} = 1, \tilde{\mathbf{x}} = \mathbf{x}) &= x_{i+1} \sum_k \mathbb{P}(\sum_{j=1}^{i} \tilde{c}_j x_j = l - k, \tilde{c}_{i+1} = k, \tilde{z} = 1, \tilde{\mathbf{x}}) \\
&\quad + (1 - x_{i+1}) \sum_k \mathbb{P}(\sum_{j=1}^{i} \tilde{c}_j x_j = l, \tilde{c}_{i+1} = k, \tilde{z} = 1, \tilde{\mathbf{x}}) \\
&= x_{i+1} \sum_k \mathbb{P}(\sum_{j=1}^{i} \tilde{c}_j x_j = l - k, \tilde{z} = 1, \tilde{\mathbf{x}}) \mathbb{P}(\tilde{c}_{i+1} = k | \sum_{j=1}^{i} \tilde{c}_j x_j = l - k, \tilde{z} = 1, \tilde{\mathbf{x}}) \\
&\quad + (1 - x_{i+1}) \sum_k \mathbb{P}(\sum_{j=1}^{i} \tilde{c}_j x_j = l, \tilde{z} = 1, \tilde{\mathbf{x}}) \mathbb{P}(\tilde{c}_{i+1} = k | \sum_{j=1}^{i} \tilde{c}_j x_j = l, \tilde{z} = 1, \tilde{\mathbf{x}}) \\
&= x_{i+1} \sum_k \sum_{k'} \delta_{i-1,k',l-k,\mathbf{x}} \frac{\delta_{ikl\mathbf{x}}}{\sum_{k''} \delta_{ik''l\mathbf{x}}} + (1 - x_{i+1}) \sum_k \sum_{k'} \delta_{i-1,k',l,\mathbf{x}} \frac{\delta_{ikl\mathbf{x}}}{\sum_{k''} \delta_{ik''l\mathbf{x}}} \\
&= \sum_{k=0}^{K} \delta_{ikl\mathbf{x}}
\end{aligned}$$

$$\mathbb{P}(\tilde{c}_i = k) = \sum_{\mathbf{x}\in\mathcal{X}} \sum_{l=0}^{nK} x_i \mathbb{P}(\tilde{z} = 1, \tilde{\mathbf{x}} = \mathbf{x}, \sum_{j=1}^{i} \tilde{c}_i\tilde{x}_i = l, \tilde{c}_i = k)$$
$$+ \sum_{\mathbf{x}\in\mathcal{X}} \sum_{l=0}^{nK} (1-x_i) \mathbb{P}(\tilde{z} = 1, \tilde{\mathbf{x}} = \mathbf{x}, \sum_{j=1}^{i} \tilde{c}_i\tilde{x}_i = l, \tilde{c}_i = k) + \mathbb{P}(\tilde{z} = 1, \tilde{c}_i = k)$$
$$= \sum_{\mathbf{x}\in\mathcal{X}} \sum_{l=k}^{nK} x_i \delta_{ikl\mathbf{x}} + \sum_{\mathbf{x}\in\mathcal{X}} \sum_{l=0}^{nK} (1-x_i)\delta_{ikl\mathbf{x}} + h_{ik} = p_{ik}$$

The last equality arises due to the constraints on δ in the LP in Theorem 2.1.

Optimality. For the distribution constructed, we have that

$$\mathbb{P}(\max_{\mathbf{x}} \tilde{\mathbf{c}}'\mathbf{x} \geq r) \geq \mathbb{P}(\tilde{z} = 1, \max_{\mathbf{x}} \tilde{\mathbf{c}}'\mathbf{x} \geq r) \geq \sum_{\mathbf{x}\in\mathcal{X}} \mathbb{P}(\tilde{z} = 1, \tilde{\mathbf{x}} = \mathbf{x}, \tilde{\mathbf{c}}'\tilde{\mathbf{x}} \geq r)$$
$$= \sum_{\mathbf{x}\in\mathcal{X}} \sum_{l=r}^{nK} \mathbb{P}(\tilde{z} = 1, \tilde{\mathbf{x}} = \mathbf{x}, \tilde{\mathbf{c}}'\tilde{\mathbf{x}} = l) = \sum_{\mathbf{x}\in\mathcal{X}} \sum_{l=r}^{nK} \sum_{k=0}^{K} \delta_{nkl\mathbf{x}} = \sum_{\mathbf{x}\in\mathcal{X}} a_{\mathbf{x}}$$

Thus the distribution constructed indeed attains the optimal value of the LP.

Appendix B. Proof of Theorem 2.10 . Upon writing down the dual of the corresponding exponential sized formulation, the analogous separation problem here boils down to enforcing $\min\{\lambda + \sum_{i=1}^n \alpha_i c_i + \sum_{(i,j)\in E} \beta_{ij} c_i c_j \text{ s.t } Z(\mathbf{c}) \geq r, \mathbf{c} \in \{0,1\}^n, \} \geq 1$, where λ, α, β are not restricted in sign. One can again use similar techniques as earlier to reformulate the above using $|\mathcal{X}|$ optimization problems,

$$\min \left\{ \lambda + \sum_{i=1}^n \alpha_i c_i + \sum_{(i,j)\in E} \beta_{ij} c_i c_j \text{ s.t } \sum_{i=1}^n c_i x_i \geq r, \mathbf{c} \in \{0,1\}^n, \right\} \geq 1 \,\forall\, \mathbf{x} \in \mathcal{X}.$$

To solve the problem above for each $\mathbf{x}$, one can adapt the dynamic programming solution in [32](originally designed for constraints $\sum_{i=1}^n c_i \geq r$) to handle the constraint $\sum_{i=1}^n c_i x_i \geq r$. Such a reformulation provides all but the first three constraints in the formulation in Theorem 2.10. From [32] the first three constraints reformulate,

$$\min \left\{ \lambda + \sum_{i=1}^n \alpha_i c_i + \sum_{(i,j)\in E} \beta_{ij} c_i c_j \text{ s.t } \mathbf{c} \in \{0,1\}^n \right\} \geq 0.$$