Algorithmica (2018) 80:3225–3252
https://doi.org/10.1007/s00453-017-0383-4

Improved Bounds in Stochastic Matching and Optimization

Alok Baveja[1] · Amit Chavan[2] · Andrei Nikiforov[3] · Aravind Srinivasan[2] · Pan Xu[2]

Received: 15 July 2016 / Accepted: 12 October 2017 / Published online: 23 October 2017

Abstract Real-world problems often have parameters that are uncertain during the optimization phase; *stochastic optimization* or *stochastic programming* is a key approach introduced by Beale and by Dantzig in the 1950s to address such uncertainty. Matching is a classical problem in combinatorial optimization. Modern stochastic versions of this problem model problems in kidney exchange, for instance. We improve upon the current-best approximation bound of 3.709 for stochastic matching due to Adamczyk et al. (in: Algorithms-ESA 2015, Springer, Berlin, 2015) to 3.224; we also present improvements on Bansal et al. (Algorithmica 63(4):733–762, 2012) for hypergraph matching and for relaxed versions of the problem. These results are obtained by improved analyses and/or algorithms for rounding linear-programming relaxations of these problems.

✉ Pan Xu
panxu@cs.umd.edu

Alok Baveja
baveja@rutgers.edu

Amit Chavan
amitc@cs.umd.edu

Andrei Nikiforov
andnikif@camden.rutgers.edu

Aravind Srinivasan
srin@cs.umd.edu

1 Department of Supply Chain Management, Rutgers Business School, Rutgers, The State University of New Jersey, Piscataway, NJ 08854, USA

2 Department of Computer Science, University of Maryland, College Park, MD 20742, USA

3 School of Business, Rutgers, The State University of New Jersey, Camden, NJ 08102, USA

Keywords Stochastic optimization · Linear programming · Approximation algorithms · Randomized algorithms

1 Introduction

Stochastic optimization deals with problems where there is uncertainty in the input: we aim at optimizing or well-approximating the expected value of an objective function that involves random input parameters. This area dates back to the classical works of Beale [6] and Dantzig [9] from the 1950s; we refer the reader to works including Birge and Louveaux [7], Ruszczynski and Shapiro [20], Shapiro et al. [22] and the references therein for modern treatments of this topic. In stochastic optimization, we postulate a probability distribution over the uncertain input parameters, and compute a (two-stage or a multi-stage) solution that optimizes the expected value of the objective function: the uncertain data are revealed over the two or more stages, and later stages may adaptively use the values revealed in earlier stages. This approach has been very fruitful for a range of problems, in areas including network design, inventory control, facility location, e-commerce, and kidney exchange (see, e.g., [1,5,8,10,14–19,23,24]).

More generally, a key issue in stochastic optimization is how the probability distribution on the uncertain data is represented. There is a spectrum of possibilities for this distribution, with one tractable and concrete model being that the uncertain parameters are independent with known distributions, while an abstract approach assumes very little about the distribution, except that we can sample independently multiple times from a black-box representing the distribution. We make progress on fundamental problems at both of these settings, with *approximation bounds and algorithms* being a key theme, as they are in the applications cited above. Let us review our notions of approximation next.

Owing to the computational intractability (known, conjectured, or otherwise) of problems in combinatorial optimization, a powerful approach that has developed over more than four decades is that of *approximation algorithms*, where we aim at efficiently computing solutions that are within a guaranteed factor of optimal; see, e.g., the textbooks [25,26]. For maximization problems with a non-negative objective function, a ρ-approximation algorithm, for $\rho \geq 1$, is a polynomial-time algorithm that always delivers a solution of value at least $1/\rho$ times optimal; for randomized algorithms, the expected solution-value output should be at least $1/\rho$ times optimal, where this expectation is over the internal randomization of the algorithm. In the context of stochastic optimization (maximization), we need to be a little more careful, since the objective function value is random due to the randomness in the stochastic input; letting OPT denote the maximum-possible expected objective-function value over all possible terminating algorithms with no constraint on the running time, a ρ-approximation algorithm is one that outputs a solution of expected value at least OPT/ρ, where the expectation is over the uncertainty of the input, and over any internal randomization of the algorithm. This will be the notion of approximation employed in Sect. 2, where we discuss our approximation algorithms for stochastic matching in a model that posits the uncertain data as being independent with known distributions.

2 Related Work and Main Contributions

Matching is well-known to be a bedrock of combinatorial optimization—a problem that has also played a key role in the advancement of new algorithmic paradigms including parallel algorithms, randomized algorithms, and, more recently, online algorithms in sponsored-search advertising. However, we do not yet have a full algorithmic understanding even for various basic *stochastic* versions of the problem, which are motivated by applications, e.g., in kidney exchange and online dating [8]. We advance this goal by improving upon the bounds of Bansal et al. [5] and Adamczyk et al. [2] for stochastic-matching problems in graphs and in uniform hypergraphs.

Informally, the basic stochastic-matching problem is as follows [5,8]. We are given a graph $G = (V, E)$ with a weight $w_e \geq 0$ and a probability $p_e \in [0, 1]$ for each edge e; each vertex v also has a positive integral "patience" t_v. Our goal is to construct a matching of maximum weight; however, there are a few catches. First, the edges are only present probabilistically: each edge e is present *independently* with probability p_e, and the presence (or lack thereof) of any edge e can only be ascertained by probing for it—adaptively, in any order we choose. However, if we choose to probe $e = (u, v)$ and find that it is present, we are forced to add it to our matching: in particular, all edges incident on e are removed immediately if e is found to be present. Furthermore, the edges incident upon any vertex v can only be probed for up to t_v times; i.e., we cannot exceed the hard constraint of the patience of any vertex. Under these constraints, the goal is to find a matching of maximum expected weight, where the expectation is taken both over the stochastic existence of the edges, and over any internal randomization of our algorithm. (In online dating, for instance, a pair of people can be matched for a date only if they are available; the possible match can only be ascertained by setting up a date; and participants may have limits on the number of unsuccessful dates they are willing to participate in. Similarly for kidney exchange.) Intriguingly, it is not yet known if it is NP-hard to obtain the optimal expected solution efficiently, and therefore the focus has been on approximation algorithms. The state of the art in terms of approximation is from the work of Adamczyk et al. [2]: 2.845- and 3.709-approximations for bipartite and general graphs respectively, improving upon Bansal et al [5] (who had presented 3- and 4-approximations respectively). We present the following two improvements for the general graphs, with Theorem 2 being a bicriteria result that allows the patience constraints to be violated by at most 1:

Theorem 1 *There is a* 3.224*-approximation algorithm for the weighted stochastic matching problem on a general graph.*

Theorem 2 *There is a* 2.675*-approximation algorithm for the weighted stochastic matching on a general graph if the patience constraints are allowed to be violated by an additive error of* 1.

In essence, the LP-based approach of Bansal et al. [5] uses a dependent-rounding algorithm of Gandhi et al [13] to first guarantee that the patience constraints are satisfied with probability one within the context of their randomized algorithm; the probing is done on top of this setup. In contrast, we randomly permute the edges and then probe them in this order, with probing probabilities suggested by the LP—of course, not probing infeasible edges in the process. An edge is infeasible if a neighboring edge has already been placed in the matching, or if one of the two end-points has had its patience exhausted. While it is not too hard to incorporate the matching constraints here, the patience constraints are far more complex to handle well: e.g., direct use of Chernoff-type bounds will not help. We work to identify extremal input-instances for our algorithm and combine this with rigorous computer-aided calculations in order to conduct our analyses. Theorem 2 follows from a new attenuation idea. The algorithms themselves are quite simple to implement; the main feature of our work is a detailed analysis of the worst-case settings for our algorithms.

Theorems 3 and 4 of Sect. 6 improve upon the $(k+1)$-approximation of Bansal et al. [5] for weighted matching in k-uniform hypergraphs.

Notation As usual, we let “ln” denote the natural logarithm; we will in some places use $\exp(x)$ to denote e^x. Also, “w.l.o.g.” will be shorthand for “without loss of generality”.

3 Preliminaries

We will often consider a uniformly random permutation π on a set of items $I = \{e_1, e_2, \ldots, e_\ell\}$. We can assume that π is chosen as follows: for each item e, we pick independently and uniformly at random a real number $\pi(e) = a_e \in [0, 1]$, and then sort these in increasing order to obtain π. Note that we abuse notation by letting π denote both the permutation and the reals chosen; however, this choice will be clear from the context.

In the context of such a randomly-chosen permutation π of our set I, the FKG inequality [11] will be quite useful to us, as follows. A Boolean function $f : \{0,1\}^t \to \{0,1\}$ is termed *increasing* if for each input $x = (x_1, x_2, \ldots, x_t) \in \{0,1\}^t$, turning any x_i from 0 to 1 cannot change the value of $f(x)$ from 1 to 0; i.e., the value of f either remains unchanged by this bit-flip, or increases from 0 to 1. Similarly, $g : \{0,1\}^t \to \{0,1\}$ is *decreasing* if for each $x = (x_1, x_2, \ldots, x_t) \in \{0,1\}^t$, turning any x_i from 1 to 0 cannot change the value of $g(x)$ from 1 to 0. The FKG inequality states that if we have *independent* random bits $R_1, R_2, \ldots, R_t$, then for all k and for all increasing or all decreasing $f_1, f_2, \ldots, f_k$ that map $\{0,1\}^t$ to $\{0,1\}$,

$$\Pr\left[\bigwedge_{i=1}^{k} \Big(f_i(R_1, R_2, \ldots, R_t) = 1\Big)\right] \geq \prod_{i=1}^{k} \mathbb{E}\big[f_i(R_1, R_2, \ldots, R_t)\big];$$

In our analyses, we will often condition on an event A of the form “$\pi(e) = x$” (where π is our random permutation as above and $x \in [0,1]$), and will need to lower-bound certain probabilities of the form $\Pr\left[\bigwedge_{i=1}^{k} B_i \;\middle|\; A\right]$; the FKG inequality is quite useful if these events B_i have a certain structure [5,21]. For all $f \in I$ such that $f \neq e$, define a random bit R_f that is 1 if $\pi(f) \leq x$, and 0 otherwise; note that even conditional on the event A, these R_f are all independent. Now, if the B_i are Boolean functions of the tuple of bits R_f such that the B_i are all increasing or all decreasing, then the FKG inequality applied to the space where we condition on A, yields

$$\Pr\left[\bigwedge_{i=1}^{k} B_i \;\middle|\; A\right] \geq \prod_{i=1}^{k} \Pr[B_i \mid A]. \tag{1}$$

We will also make use of the following form of the Chernoff-Hoeffding bound [3]:

Definition 1 (*Chernoff-Hoeffding Bound*) Let $X_1, \ldots, X_n$ be n independent random variables with $0 \leq X_i \leq 1$. Let $X = X_1 + \ldots + X_n$ and $\mu = \mathbb{E}[X]$. Then for any $\epsilon > 0$,

$$\Pr[X \geq (1+\epsilon)\mu] \leq \exp\left(-\frac{\epsilon^2}{2+\epsilon}\mu\right), \text{ and}$$
$$\Pr[X \leq (1-\epsilon)\mu] \leq \exp\left(-\frac{\epsilon^2}{2}\mu\right)$$

Notation We will refer to a value $z \in [0, 1]$ as *floating* if $z \in (0, 1)$. We let Pois(λ) denote the Poisson distribution with mean λ. Also, "$R \sim D$" will denote that random variable R is sampled from distribution D.

4 Stochastic Matching

We consider the following stochastic matching problem. The input is an undirected graph $G = (V, E)$ with a weight w_e and a probability value p_e on each edge $e \in E$. In addition, there is an integer value t_v—the *patience*—for each vertex $v \in V$. Initially, each vertex $v \in V$ has patience t_v. At any step in the algorithm, only an edge $e(u, v) \in E$ such that $t_u > 0$ and $t_v > 0$ can be probed. Upon probing such an edge e, one of the following happens: (1) with probability p_e, e exists; u and v get matched and are removed from G along with their incident edges, or (2) with probability $(1 - p_e)$, e does not exist; e is removed, and t_u and t_v are reduced by 1. (All these edge-existence events are independent.) We seek to find an adaptive strategy for probing edges; its performance is measured by the expected weight of the matched edges. We prove Theorem 1 now.

Consider the following natural LP relaxation [5]: for any vertex $v \in V$, $\partial(v)$ denotes the edges incident to v. The LP variable y_e denotes the probability that edge $e(u, v)$ gets probed in the adaptive strategy, and hence $y_e p_e$ is the probability that e gets matched in the strategy.

$$\text{Maximize} \sum_{e \in E} w_e y_e p_e \tag{2}$$

$$\text{Subject to} \sum_{e \in \partial(v)} y_e p_e \leq 1 \qquad \forall v \in V \tag{3}$$

$$\sum_{e \in \partial(v)} y_e \le t_v \qquad \forall v \in V \qquad (4)$$

$$0 \le y_e \le 1 \qquad \forall e \in E \qquad (5)$$

Lemma 1 [5] *The optimal value for the LP (2) is an upper bound on the performance of any adaptive algorithm for stochastic matching.*

For notational convenience, we use $\{y_e\}$ to denote the optimal solution to the LP in Eq. (2). For an edge $e(u, v)$, it is called *safe* at the time it is considered if: (1) neither u nor v is matched, and (2) $t_u > 0$ as well as $t_v > 0$. Our algorithm, denoted by SM_1, first fixes a uniformly random permutation π on the set of edges E. It then inspects the edges one by one in the order of π. If an edge e is safe, the algorithm probes it (independently) with probability y_e, otherwise it skips to the next one. Note that SM_1 is actually a special case of the algorithm presented in Bansal et al. [4] even though their analysis yields only a 5.75-approximation ratio. For ease of analysis, we state our algorithm SM_1 in a slightly different but equivalent way in Algorithm 1.

Algorithm 1: SM_1 : Stochastic Matching

1 Choose a random permutation π on E.
2 For each edge $e \in E$, generate a random bit $Y_e = 1$ independently with probability y_e. Let E' be the set of edges with $Y_e = 1$.
3 Follow the random order π to inspect edges in E'
4 If an edge e is safe, then probe it; otherwise, skip it.

To analyze the performance of our algorithm, we conduct an edge-by-edge analysis. Recall that $y_e p_e$ is the probability that e is matched in the LP (2), and the optimal value of the LP is exactly $\sum_{e \in E} w_e p_e y_e$. The expected weight of the matching found by our algorithm is $\mathbb{E}[\mathsf{SM}_a] = \sum_{e \in E} w_e p_e \cdot \Pr[e \in E'] \cdot \Pr[e \text{ gets probed} | e \in E']$, which is $\sum_{e \in E} w_e p_e y_e \cdot \Pr[e \text{ gets probed} | e \in E'] \ge \sum_{e \in E} w_e p_e y_e \lambda$, assuming $\Pr[e \text{ gets probed} | e \in E'] \ge \lambda$. This gives us a λ-approximation algorithm.

The subsequent discussion focuses on how to lower-bound the value of λ. Consider a specific edge $e = e(u, v)$, and let $E(u)$ be the set of edges incident to u *excluding* e itself, i.e. $E(u) = \partial(u) \backslash \{e\}$. Let $\pi(e) = x, 0 < x < 1$. Conditioned on $\pi(e) = x$, with $0 < x < 1$, and $Y_e = 1$, let $\mathcal{P}_u$ be the probability that e is *not blocked* by any of the edges in $E(u)$ in the algorithm SM_1. We say that e is *blocked* by some edge f in $E(u)$ if f gets matched or the patience constraint of u gets tight resulting from probing f (i.e. $t_u = 0$). We assume without loss of generality that $|E(u)| \ge t_u$, otherwise the patience constraint for node u is redundant.

A little thought gives us the following lower bound on $\mathcal{P}_u$:

$$\mathcal{P}_u \ge P_u = \sum_{S \subseteq E(u), |S| \le t_u - 1} x^{|S|} \prod_{f \in S} y_f (1 - p_f) \prod_{f \notin S} (1 - x y_f) \qquad (6)$$

To see why this is true, let Y'_f (for any $f \in E(u)$) be the indicator random variable that is 1 if and only if f gets matched when probed, i.e., $\Pr[Y'_f = 1] = p_f$. For each

$S \subseteq E(u)$ such that $|S| \leq t_u - 1$, we associate an event A_S that happens when both of the following conditions are met: (1) Each edge $f \in S$ falls before e in π with $Y_f = 1$ and $Y'_f = 0$; and (2) each edge $f \notin S$ either falls after e in π or $Y_f = 0$. We can see that this event guarantees that e will not be blocked by any edge of S. Thus, $\mathcal{P}_u$ should be at least the probability that one or more of A_S happen, which is exactly P_u.

Next, we focus on *adversarial configurations* of $E(u)$, i.e, how are the edges in $E(u)$ arranged so as to minimize the value of P_u subject to the constraints: (1) $\sum_{f \in E(u)} y_f p_f \leq 1$, (2) $\sum_{f \in E(u)} y_f \leq t_u$ and (3) $0 \leq y_f, p_f \leq 1$ for each $f \in E(u)$. Here we view x as a (given) parameter. We denote such adversarial configurations of $E(u)$ as the **worst-case structure (WS)** of $E(u)$. Notice that we give the (hypothetical) adversary extra power of manipulating the values of p_f and number of edges in $E(u)$, both of which are actually part of the input.

Lemma 2 *In WS, there will be at most one edge with* $p_f = 1$ *and at most one edge with* $0 < p_f < 1$. *All other edges must have* $p_f = 0$.

Proof We prove by contradiction. Assume there are two edges, say $p_1 = p_2 = 1$ in WS. Then, $y_1 + y_2 \leq 1$ since $\sum_i y_i p_i \leq 1$. We perturb the current configuration as follows: merge the two edges into a single edge e_3 where $y_3 = y_1 + y_2$ and $p_3 = 1$. After this perturbation, both values, $\sum_{f \in E(u)} y_f p_f$ and $\sum_{f \in E(u)} y_f$, remain unchanged. Thus, both the matching and patience constraints are maintained at u, and our perturbation gives a feasible configuration.

The change brought by this perturbation to the value P_u is as follows: for each non-zero term in P_u associated with some $S \subseteq E(u)$ where $e_1 \notin S, e_2 \notin S$, the term $(1 - xy_1)(1 - xy_2)$ will be replaced with $(1 - x(y_1 + y_2))$, which results in a strictly lower value of P_u. This is a contradiction.

Now assume there are two edges a, b with $0 < p_a, p_b < 1$ in WS. Consider the following perturbation: for some small $\varepsilon \neq 0$, set $p'_a = p_a + \varepsilon/y_a$ and $p'_b = p_b - \varepsilon/y_b$. After this perturbation, both of $\sum_{f \in E(u)} y_f p_f$ and $\sum_{f \in E(u)} y_f$ remain unchanged and the perturbed configuration is still feasible.

Let $f(\varepsilon)$ be the value of P_u after this update. In the expression of P_u, the terms contributing to ε^2 must be those associated with S where $a, b \in S$. Notice that

$$(1 - p'_a)(1 - p'_b) = (1 - p_a - \varepsilon/y_a)(1 - p_b + \varepsilon/y_b)$$

has a negative coefficient of ε^2, implying that the second derivative f'' is negative. Therefore we can always find a non-zero value of ε to make P_u strictly smaller. Again a contradiction. □

Let $E_1(u)$ and $E_0(u)$ be the set of edges in WS which have $p_f = 1$ and $p_f = 0$ respectively. Let a be the potential edge taking a floating value, $0 < p_a < 1$. Lemma 2 tells us $E_1(u)$ contains at most one such edge in the WS. Let $A = \sum_{f \in E_1(u)} y_f$.

Based on Lemma 2, we can update the expression of P_u as

$$P_u = (1 - xA)(1 - xy_a)\Pr[Z_u \leq t_u - 1] + (1 - xA)xy_a(1 - p_a)\Pr[Z_u \leq t_u - 2] \quad (7)$$

where $Z_u = \sum_{f \in E_0(u)} Z_f$ and the $(Z_f : f \in E_0(u))$ are independent Bernoulli random variables with $\Pr[Z_f = 1] = xy_f, \forall f \in E_0(u)$. (We are abusing notation in the equation $Z_u = \sum_{f \in E_0(u)} Z_f$ by reusing the symbol Z for the l.h.s. and the r.h.s.; this will not cause any confusion as the identity of Z will always be clear from the context.)

The following lemma is proved in the "Appendix".

Lemma 3 *In WS,* $p_a = 0$.

From Lemma 3, we can claim that there is no edge f which has $p_f \in (0, 1)$. Thus, we can further simplify the expression of P_u in Eq. (7) as

$$P_u = (1 - xA)\Pr[Z_u \le t_u - 1]. \tag{8}$$

Lemma 4 reveals additional structure of the WS.

Lemma 4 *In WS, we have* $A = 1$ *and* $Z_u \sim \text{Pois}(x(t_u - 1))$.

Proof We show $A = 1$ by contradiction. Assume $A < 1$ in WS. Notice that $E_0(u)$ is non-empty since $\mathbb{E}[Z_u] = \sum_{f \in E_0(u)} \mathbb{E}[Z_f] = x(t_u - A) > 0$. Next, consider an arbitrary edge $f \in E_0(u)$ with $y_f \in (0, 1]$. Let $Z'_u = Z_u - Z_f$. Then,

$$\begin{aligned} P_u &= (1 - xA)\Pr[Z_u \le t_u - 1] \\ &= (1 - xA)\left(\Pr[Z'_u \le t_u - 2] + (1 - y_f x)\Pr[Z'_u = t_u - 1]\right) \\ &= (1 - xA)\Pr[Z'_u \le t_u - 2] + (1 - (y_f + A)x + y_f A x^2)\Pr[Z'_u = t_u - 1]. \end{aligned}$$

We have two cases:

(i) $A < y_f$. In this case, P_u can be decreased by interchanging the values A and y_f.
(ii) $A \ge y_f$. In this case, P_u can be decreased by perturbing as $A' = A + \varepsilon$ and $y'_f = y_f - \varepsilon$ for some small $\epsilon > 0$.

Notice that in case **(i)**, after interchanging the values A and y_f, the value $\sum_{f \in E(u)} y_f p_f$ will change from A to y_f and thus is at most 1, since $y_f \le 1$ for each $f \in E$. As for case **(ii)**, the value $\sum_{f \in E(u)} y_f p_f$ will change from A to $A + \epsilon$. Since $A < 1$, we can always find a $\epsilon > 0$ such that $A + \epsilon \le 1$ such that the constraint $\sum_{f \in E(u)} y_f p_f \le 1$ is maintained. Thus, the value $(A + y_f)$ remains unchanged after perturbation in both cases and the constraint $\sum_{f \in E(u)} y_f \le t_u$ is maintained. In either case, we end up at a feasible configuration in which P_u is strictly lower than that in WS. This yields a contradiction.

The second part of the lemma, that $Z_u \sim \text{Pois}(x(t_u - 1))$, is proved in Lemma 11 in the "Appendix". □

At this point, we have all the ingredients to prove Theorem 1.

Proof We have $\Pr[e \text{ gets probed } | Y_e = 1] = \int_0^1 \mathcal{P}_u \mathcal{P}_v dx \ge \int_0^1 P_u P_v dx$, i.e., at least

$$H(t_u, t_v) \doteq \int_0^1 (1 - x)^2 \Pr[Z_u \le t_u - 1]\Pr[Z_v \le t_v - 1]dx,$$

where $Z_u \sim \text{Pois}(x(t_u - 1))$ and $Z_v \sim \text{Pois}(x(t_v - 1))$. We verified that the above expression has a minimum value of $0.31016 = 1/3.224$ at $t_u = t_v = 2$. All our numerical computations were done on Mathematica 10 with precision at least up to the fourth digit after the decimal point. We split the whole verifications into the following three cases: (1) $1 \leq t_u, t_v \leq 20$; (2) $t_u, t_v \geq 20$ and (3) $1 \leq t_u \leq 20$ while $t_v \geq 20$. Notice that $H(t_u, t_v)$ is symmetric in the two variables and thus our verifications are complete.

- For $1 \leq t_u, t_v \leq 20$, we can numerically verify that $H(t_u, t_v)$ achieves its minimum value of $0.31016 = 1/3.224$ at $t_u = t_v = 2$.
- For $t_u, t_v \geq 20$, the Chernoff bound from Definition 1 implies that $H(t_u, t_v)$ should be at least

$$\int_0^1 (1-x)^2 \left[1 - \exp\left(\frac{-\epsilon^2 x(t_u - 1)}{2+\epsilon}\right)\right]\left[1 - \exp\left(\frac{-\epsilon^2 x(t_v - 1)}{2+\epsilon}\right)\right] dx,$$

  where $\epsilon = \epsilon(x) = \frac{1}{x} - 1$; by plugging in $t_u = t_v = 20$, we can verify numerically that this integral is at least 0.316324.
- Similarly, for $1 \leq t_u \leq 20$ while $t_v \geq 20$, we can verify numerically (by checking all integers $1 \leq t_u \leq 20$) that with $\epsilon = \frac{1}{x} - 1$,

$$H(t_u, t_v) \geq \int_0^1 (1-x)^2 \Pr(Z_u \leq t_u - 1) \left[1 - \exp\left(\frac{-\epsilon^2 x(20 - 1)}{2+\epsilon}\right)\right] dx,$$

  which is at least 0.312253.

This establishes the key claim that $\Pr[e \text{ gets probed } | Y_e = 1] \geq 0.3101$ for each $e \in E$. □

5 Stochastic Matching with Relaxed Patience

In this section, we consider the variant of the stochastic matching problem in which the patience constraints are allowed to be violated by at most 1, and prove Theorem 2. From the analysis in Sect. 4, we observe that edges with a large $y_e p_e$ value are probed with a much higher probability than those with small ones. This indicates that small edges (those having a "small" $y_e p_e$ value) are the ones that are the bottleneck for the performance of our algorithm. Our high level idea here is to *attenuate* "large" edges in order to improve the performance of the small ones. The process of attenuation carefully calculates a value $h_e \in (0, 1]$, called the *attenuation factor*, for each $e \in E$. Thereafter, instead of probing an edge e with probability y_e as in algorithm SM_1, our algorithm probes it with probability $h_e y_e$. We will show that such a strategy balances the performance of large and small edges and improves the overall performance of SM_1.

The overall picture of our algorithm, denoted SM_2, is as follows. First we label each edge $e \in E$ as "large" if $y_e p_e > 1/2$ and "small" if $y_e p_e \leq 1/2$. Similar to SM_1, we follow a random permutation π on the set of edges E to inspect each edge. If an edge e is safe when considered, we probe it with probability $h_e y_e$; otherwise we skip it. Here $h_e = h$ if e is large and $h_e = 1$ otherwise, where $h \geq 1/2$ is a parameter that we optimize later. For ease of analysis, we state the algorithm SM_2 in an alternative but essentially equivalent way in Algorithm 2.

Algorithm 2: SM_2 : Stochastic matching with relaxed patience

1. Choose a random permutation π of E.
2. For each edge $e \in E$, set $h_e = h$ if $y_e p_e > 1/2$, set $h_e = 1$ otherwise.
3. For each edge $e \in E$, generate a random bit $Y_e = 1$ with probability $h_e y_e$. Let E' be the set of edges with $Y_e = 1$.
4. Follow the random order π to inspect edges in E'
5. If an edge e is safe, probe it; otherwise, skip it.

In the spirit of Sect. 4, we focus on analyzing the performance of an edge $e(u, v)$ in SM_2. However, this analysis is more involved and we present only the main results in this section. For detailed proofs, please refer to the "Appendix". All notation used in this section is consistent with those introduced in Sect. 4.

As before, we can write the expression for the lower bound P_u for $\mathcal{P}_u$ in (6) as follows:

$$\mathcal{P}_u \geq P_u = \sum_{S \subseteq E(u), |S| \leq t_u} x^{|S|} \prod_{f \in S} h_f y_f (1 - p_f) \prod_{f \notin S} (1 - x h_f y_f) \tag{9}$$

Notice that WS for a small edge e happens when $y_e p_e = 0$. In other words, we give the adversary power to set $\sum_{f \in E(u)} y_f p_f = 1$. For a large edge e with $y_e p_e > 1/2$, we re-define the WS by setting the constraints on the adversary as (1) $\sum_{f \in E(u)} y_f p_f \leq 1/2$, (2) $\sum_{f \in E(u)} y_f \leq t_u - 1/2$ and (3) $0 \leq y_f \leq 1$ for each $f \in E(u)$. The following lemma specifies the structure of the WS for a small edge and large edge respectively.

Lemma 5 *For $t_u \geq 2$, the WS at $E(u)$ is as follows:*

– *WS for a small edge can be characterized as either $Q_1 = (A = 1, y_a = 0, Z_u \sim \mathrm{Pois}(x(t_u - 1))$ or $Q_2 = (A = 1/2, y_a = B - 1/2, p_a = 1/(2y_a), Z_u \sim \mathrm{Pois}(x(t_u - B))$ for some $1 \leq B \leq 3/2$. The expression for P_u at Q_1 and Q_2 can be updated as below.*

$$P_u(\textbf{\textit{Small}}_1) = (1 - xh)\Pr[Z_u \leq t_u]$$
$$P_u(\textbf{\textit{Small}}_2) = \left(1 - \frac{1}{2}x\right)\left(\left(1 - \frac{1}{2}x\right)\Pr[Z_u \leq t_u - 1] + \left(1 - x\left(B - \frac{1}{2}\right)\right)\Pr[Z_u = t_u]\right)$$

– *WS for a large edge can be characterized as* $(A = 0, y_a = B, p_a = \frac{1}{2B}, Z_u \sim \text{Pois}(x(t_u - B - 1/2))$ *for some* $1/2 \leq B \leq 1$. *The updated expression for* P_u *is:*

$$P_u(\textbf{\textit{Large}}) = \left(1 - \frac{1}{2}x\right) \Pr[Z_u \leq t_u - 1] + (1 - xB) \Pr[Z_u = t_u]$$

Lemma 6 *The value* P_u *of the WS when* $t_u = 1$ *is at least as large as its value when* $t_u = 2$.

We prove Lemmas 5 and 6 in the "Appendix". Here we present the proof of Theorem 2 using these Lemmas.

Proof Lemma 6 implies that we can ignore the case $t_u = 1$. Depending on whether e is large or small, the probability that e gets probed in algorithm SM_2 in WS is:
(a) If e is a large edge:

$$\begin{aligned} \Pr[e \text{ gets probed}]/y_e &= h \Pr[e \text{ gets probed } | Y_e = 1] \\ &\geq \left(h \int_0^1 P_u(\textbf{Large}) P_v(\textbf{Large}) dx\right) \doteq \mathbf{I}_L \end{aligned}$$

(b) If e is a small edge, then we see
$\Pr[e \text{ gets probed }]/y_e = \Pr[e \text{ gets probed } | Y_e = 1]$, which is at least

$$\begin{aligned} \mathbf{I}_S \doteq \min\Bigg(& \int_0^1 P_u(\textbf{Small}_1) P_v(\textbf{Small}_1) dx, \\ & \int_0^1 P_u(\textbf{Small}_1) P_v(\textbf{Small}_2) dx, \int_0^1 P_u(\textbf{Small}_2) P_v(\textbf{Small}_2) dx\Bigg) \end{aligned}$$

The approximation ratio of algorithm SM_2 is determined by $\min(\mathbf{I}_L, \mathbf{I}_S)$. We can numerically verify that this minimum is maximized at $h = 0.7$, the value is 0.373799, and the configuration is $(y_e p_e = 0, t_u = t_v = 5, B_1 = B_2 = 1.4984)$. All the numerical details can be seen in the "Appendix". □

6 Stochastic Hypergraph Matching

We now consider stochastic matching in a k-uniform hypergraph, i.e., a hypergraph where all edges have size exactly k. However, unlike before, we do not consider patience constraints (the work of Bansal et al. [5] proceeds similarly). The following LP can be obtained by naturally extending the LP in (2), where $\partial(v)$ denotes the set of hyperedges incident to v:

$$\max \sum_{e \in E} w_e y_e p_e \text{ subject to } \sum_{e \in \partial(v)} y_e p_e \leq 1, \forall v \in V;\ 0 \leq y_e \leq 1, \forall e \in E \quad (10)$$

Theorems 3 and 4 improve upon the $(k+1)$-approximation of Bansal et al. [5] for weighted matching in k-uniform hypergraphs. Both of these algorithms classify the hyperedges as "small" or "large" based on the LP values, and treat each group separately. The difference is as follows. The algorithm of Theorem 3 attenuates the small edges to boost the performance of large edges; the algorithm of Theorem 4 uses a "weighted permutation" of the hyperedges such that each large edge has a higher chance to fall behind a small edge. Although Theorem 4 is asymptotically better, we present both theorems since their ideas can be useful elsewhere.

Note that the LP-based methods of Bansal et al. [5] and ours cannot in general do better than $k-1+1/k$ [12]; hence, we are close to optimal for LP-based approaches.

Theorem 3 *There is a $(k+\frac{1}{2}+o(1))$-approximation algorithm for the stochastic matching problem on a k-uniform hypergraph, where the "$o(1)$" term is a function of k that goes to zero as k becomes large.*

Theorem 4 *For any given $\epsilon > 0$, there is a $(k+\epsilon+o(1))$-approximation algorithm for the stochastic matching problem on a k-uniform hypergraph, where the "$o(1)$" term is a function of k that goes to zero as k becomes large.*

We next present the algorithms and proofs for these two theorems.

6.1 An Algorithm Achieving a $(k+1/2+o(1))$ Approximation Ratio

For notational convenience, let $\{y_e\}$ be an optimal solution to LP (10). At a high level, our algorithm proceeds according to the outline below. Let $c \geq 1/2$ be a parameter, which will be optimized at $1/2$ later.

1. Divide the edges into two sets, the "small" edge set $E_S = \{e | y_e p_e \leq c\}$, and the "large" edge set $E_L = E \setminus E_S$.
2. Choose a random permutation π of E_S.
3. Sample each edge $e \in E_S$ with probability y_e, independent of other edges. Let E'_S be the set of sampled edges.
4. Follow the order π to inspect if each (small) edge $e \in E'_S$ is safe or not. If e is safe, probe it with probability h_e; otherwise, skip it. Here $0 < h_e \leq 1$ is a parameter to be determined later.
5. After inspecting all small edges, remove all the unsafe large edges from E_L, and probe others with probability 1 (in arbitrary order).

Roughly speaking, an edge e being "safe" means that none of the edges in the neighborhood of e are matched. Later, we will give a definition that is both stronger and *exactly* computable. Based on the new definition, we compute an *attenuation factor* h_e for each $e \in E_S$, such that at the end of the algorithm, e is probed with probability *exactly equal* to y_e/λ. Here, $\lambda \geq 1$ is our target approximation ratio. All that remains is to analyze the performance of each large edge $e \in E_L$ and show that e is probed with probability at least y_e/λ. This, then, will give us a λ-approximation algorithm.

We redefine the notion of a small edge e being safe. Suppose π is the random order on E_S and $\pi(e) = x, 0 < x < 1$. Let $N_S[e]$ be the set of small edges in the neighborhood of e. For each $f \in N_S[e]$, let X_f, Y_f, Z_f be three random variables such that: $X_f = 1$ if f falls before e in π, $Y_f = 1$ if $f \in E'_S$ and $Z_f = 1$ if f exists in the hypergraph when probed. Note that the collection of random variables $\{X_f, Y_f, Z_f | f \in N_S[e]\}$ are mutually independent. For each $f \in N_S[e]$, let A_f be the event that $(X_f + Y_f + Z_f \leq 2)$ and $\mathsf{S}_e = \wedge_{f \in N_S[e]} A_f$. *We define e to be safe iff* S_e *happens*. Lemma 7 computes the probability that a small edge e is safe in our algorithm.

Lemma 7

$$\Pr[\mathsf{S}_e] = \int_0^1 \Pr[\mathsf{S}_e | \pi(e) = x] dx = \int_0^1 \prod_{f \in N_S[e]} (1 - x y_f p_f) dx. \tag{11}$$

Proof By definition, $\Pr[X_f = 1 | \pi(e) = x] = x$. Note that $\Pr[Y_f = 1] = y_f$, $\Pr[Z_f = 1] = p_f$, and that these two random variables are independent of $\pi(e)$. Thus, given $\pi(e) = x$, A_f will occur with probability $(1 - x y_f p_f)$. Since the A_f are independent for $f \in N_S[e]$, the proof is completed. □

Here are two interesting points for the event S_e: (1) When S_e happens, e must be safe according to our initial definition, i.e., none of the edges in its neighborhood get matched; the contrary is not true. Thus the new definition is more strict. (2) On checking e in the algorithm, we might not know if S_e occurs or not due to some missing Z_f for $f \in N_S[e]$. For instance, suppose some $f \in N_S[e]$ gets blocked by some small edge $f' \in N_S[f]$ while $X_f = Y_f = 1$. In this case, we do not know the value of Z_f since f will not be probed. In order to continue our algorithm, we simulate Z_f by generating a random bit $Z_f = 1$ with probability p_f and $Z_f = 0$ otherwise. Notice that if $Z_f = 1$, we will view e as not safe and will not probe it, even though it might be safe according to our initial definition.

The full picture of algorithm SM_3 can be seen in Algorithm 3.

6.2 Analysis of SM_3

We first analyze the performance of a small edge. For each edge $e \in E_S$,

$$\Pr[e \text{ gets probed}] = y_e h_e \Pr[e \text{ is safe} | Y_e = 1] = y_e h_e \Pr[\mathsf{S}_e].$$

To ensure that each small edge $e \in E_S$ is probed with probability *equal* to y_e/λ, we can set $h_e = 1/(\lambda \Pr[\mathsf{S}_e])$ if we can ensure that $\Pr[\mathsf{S}_e] \geq 1/\lambda$. The following lemma states that this goal is achievable. Recall that $c \geq 1/2$ is the threshold such that an edge e is small iff $y_e p_e \leq c$.

Lemma 8

$$\Pr[\mathsf{S}_e] \geq \frac{1 - (1 - c)^{k/c+1}}{k + c}$$

Algorithm 3: SM_3: Stochastic Matching on a k-uniform hypergraph

Initially all edges are safe.
Split the edges into two sets, the "small" edge set $E_S = \{e | y_e p_e \leq c\}$ and the "large" edge set $E_L = E \setminus E_S$ where $c \geq 1/2$.
Choose a random permutation π on E_S.
For each $e \in E_S$, generate a random bit $Y_e = 1$ with probability y_e. Let E'_S be the set of (small) edges with $Y_e = 1$.
Follow the random order π to check if S_e happens or not for each $e \in E'_S$.
  if S_e *happens* **then**
    Probe e with probability h_e.
    if *e is matched (exists)* **then**
      Set $Z_e = 1$ and mark all its neighboring large edges as unsafe.
    else
      Set $Z_e = 0$.
  else
    Generate a random bit $Z_e = 1$ with probability p_e.
  Probe each safe large edge with probability 1 in an arbitrary order.

Proof Consider a small edge e, say $e = (v_1, v_2, \cdots, v_k)$ and $\pi_e = x$. Let $E(v_i)$ be the set of edges incident to v_i excluding e itself. Notice that $N_S[e] = \cup_{i=1}^{k} E(v_i)$. Therefore by Lemma 7, we have

$$\Pr[\mathsf{S}_e | \pi(e) = x] = \prod_{f \in N_S[e]} (1 - x y_f p_f) \geq \prod_{i=1}^{k} \prod_{f \in E(v_i)} (1 - x y_f p_f)$$

From the proof of Lemma 10, we see that $\prod_{f \in E(v_i)} (1 - x y_f p_f) \geq (1 - xc)^{1/c}$ for each $1 \leq i \leq k$. Thus by an application of the FKG inequality as in (1), we get that $\Pr[\mathsf{S}_e | \pi(e) = x] \geq (1 - xc)^{k/c}$.

Integrating over [0, 1], we get

$$\Pr[\mathsf{S}_e] = \int_0^1 \Pr[\mathsf{S}_e | \pi(e) = x] \geq \frac{1 - (1-c)^{k/c+1}}{k + c} dx.$$

□

At this point, we have all the ingredients to prove Theorem 3.

Proof For small edges, Lemma 8 gives us a sufficient condition to guarantee that each small edge is probed with probability *exactly equal* to y_e/λ, i.e.,

$$\Pr[\mathsf{S}_e] \geq \frac{1 - (1-c)^{k/c+1}}{k + c} \geq \frac{1}{\lambda}. \tag{12}$$

We now analyze the performance of large edges in SM_3. For each $e \in E_L$, let S_e be the event that e is safe when considered in SM_3, i.e., none of small edges in the neighbor of e gets matched. Since each small edge f gets matched with probability equal to $\frac{y_f p_f}{\lambda}$, we have that for each large item $e \in E_L$, $\Pr[\mathsf{S}_e] \geq 1 - \frac{(1-c)k}{\lambda}$ by applying the union bound.

In order to ensure that each large edge gets probed with probability at least $\frac{y_e}{\lambda}$, it suffices to set

$$\Pr[\mathsf{S}_e] \geq 1 - \frac{(1-c)k}{\lambda} \geq \frac{1}{\lambda} \tag{13}$$

Observe that for a small edge e, the lower bound of $\Pr[\mathsf{S}_e]$ from (12) is a decreasing function of c, while for a large edge e, the lower bound in (13) is an increasing function of c. Thus to find the optimal value for λ, we choose c that maximizes the minimum of the two,

$$1 - \frac{(1-c)k}{\lambda} = \frac{1-(1-c)^{k/c+1}}{k+c} = \frac{1}{\lambda}$$

The solution above is $c = \frac{1}{k+1} + o(\frac{1}{k+1})$. However, this is not feasible because by assumption, $c \geq 1/2$. Thus the optimal c^* equals 1/2, in which case $\frac{1}{\lambda} = \frac{1}{k+1/2} - O(1/(k4^k))$, and each small edge is safe to probe with probability $\frac{1}{\lambda}$ while each large edge is safe with probability $\frac{1}{2} + o(1/k)$. □

6.3 An Algorithm Achieving a $(k + \epsilon + o(1))$ Approximation Ratio

In this section, we present a simple randomized algorithm that achieves an approximation ratio of $(k + \epsilon + o(1))$ for stochastic matching on a k-uniform hypergraph, where $\epsilon > 0$ is a given constant.

Let (x, y) be an optimal solution to the LP (10). Assume w.l.o.g. $1/\epsilon = L$ where N is an integer. Let a be a constant such that $1-1/L < a < 1$. We say an edge e is "large" if $y_e p_e > 1/L$; otherwise we call e "small". For each small edge e, we draw a random real number x_e uniformly from [0, 1]. For each large edge e, we draw a random real number x_e from $[0, \delta]$ with density a and from $(\delta, 1]$ with density $(1 - a\delta)/(1 - \delta)$, where $\delta = \min(1, L(1-a^{1/(L-1)})$. Then we derive a random permutation π by sorting $\{x_e, e \in E\}$ in increasing order. Assuming L is sufficiently large, the value δ is at most $1/L + o(1/L)$. Notice that L, a and δ are all fixed constants. Based on π, we sketch our randomized algorithm as follows. Here we say an edge is *safe* iff none of its neighbors gets matched.

Algorithm 4: SM_4: Stochastic Matching on a k-uniform hypergraph

1 Initially all edges are safe.
2 Follow the random order π to check each edge $e \in E$ if it is safe or not.
3 If e is safe, then probe it with probability y_e; otherwise, skip it.

The lemmas below are useful for the proof of Theorem 4.

Lemma 9 *For any $c > 1/L$ and $0 < x < \delta$, we have*

$$1 - axc > (1 - x/L)^{cL}$$

Proof Define $F(x) = 1 - axc - (1 - x/L)^{cL}$. We can verify that: (1) $F(0) = 0$, and (2) $F'(x) > 0$ for any $0 \le x < \delta$. This gives the desired result. □

For each edge e, define

$$c_e = y_e p_e.$$

Consider an edge $e = (v_1, v_2, \cdots, v_k)$. Suppose $y_e p_e = c_e < 1 - 1/L$ and $x_e = x, 0 < x < \delta$. For each $1 \le i \le k$, let $E(v_i)$ denote the set of edges incident to v_i excluding e itself. Denote by $\mathcal{S}_i$ the event that none of the edges in $E(v_i)$ come before e and get matched.

Lemma 10

$$\Pr[\mathcal{S}_i] \ge (1 - x/L)^{(1-c_e)L}.$$

Proof From LP (10), we see $\sum_{f \in E(v_i)} y_f p_f \le 1 - c_e$. Let A and B be the set of small edges and large edges in $E(v_i)$ respectively. Observe that

$$\Pr[\mathcal{S}_i] \ge \prod_{f \in A} (1 - xc_f) \prod_{f \in B} (1 - axc_f). \tag{14}$$

Now we investigate how an adversary can minimize the RHS of (14) subject to the constraint $\sum_{f \in E(v_i)} y_f p_f \le 1 - c_e$. By Lemma 9, the adversary will not put any large edge f in B: otherwise it could further decrease the RHS by splitting f into $c_f L$ copies of small edges f' with each $c_{f'} = 1/L$ while maintaining the constraint. Thus the adversary aims to minimize $\prod_{f \in A} (1 - xc_f)$ subject to $\sum_{f \in E(v_i)} c_f \le 1 - c_e$ with $0 \le c_f \le 1/L$ for each f. By applying a local perturbation as in Lemma 2, the RHS will be minimized when there are $(1 - c_e)L$ small edges in A, with each such small edge f having $c_f = 1/L$. □

We next prove Theorem 4.

Proof We consider two cases.

1. Consider a small edge e, say $e = (v_1, v_2, \cdots, v_k)$ and $x_e = x$. From Lemma 10, we see $\Pr[\mathcal{S}_i] \ge (1 - x/L)^L$ for each $1 \le i \le k$. Thus by applying the FKG inequality (1), we get $\Pr\left[\bigwedge_i \mathcal{S}_i\right] \ge (1 - x/L)^{kL}$, which is followed by

$$\Pr[e \text{ is checked as safe }] \ge \int_0^\delta (1 - x/L)^{kL} dx = \frac{1}{k + 1/L} - O\left(k_0^k / k\right),$$

where $k_0 = (1 - \delta/L)^L < 1$ is bounded away from 1.

2. Consider a large edge e, say $e = (v_1, v_2, \cdots, v_k)$ and $x_e = x$. From Lemma 10, we see $\Pr[\mathcal{S}_i] \geq (1 - x/L)^{L-1}$ for each $1 \leq i \leq k$. Thus by applying FKG, we see when $x \leq \delta$, $\Pr\left[\bigwedge_i \mathcal{S}_i\right] \geq (1 - x/L)^{k(L-1)}$, which is followed by

$$\begin{aligned}\Pr[e \text{ is checked as safe }] &\geq \int_0^\delta a(1 - x/L)^{k(L-1)} dx \\ &\geq \frac{aL}{L-1} \frac{1}{k + 1/(L-1)} - O\left(k_1^k / k\right) > \frac{1}{k}\end{aligned}$$

where $k_1 = (1 - \delta/L)^{L-1} < 1$ is bounded away from 1; we use the fact that $a > 1 - 1/L$ to get the last inequality above.

□

7 Conclusion

We have considered randomized approximation algorithms for stochastic-matching problems. The algorithms themselves are quite simple to describe and implement. Several open questions remain, some of which are as follows.

In the context of stochastic matching, is the basic problem NP-hard? This would be interesting to ascertain even for the bipartite case. Assuming such hardness, it would be fruitful to determine the optimal approximation guarantee achievable in polynomial time: this could conceivably be 2. More generally, as compared to the mature body of work on optimal approximation thresholds in deterministic combinatorial optimization, such thresholds are ripe for understanding in the stochastic setting; stochastic matching would be an excellent start.

Acknowledgements A preliminary version of this paper appears as part of a paper in the *Proc. International Workshop on Approximation Algorithms for Combinatorial Optimization Problems* (APPROX), 2015. We thank R. Ravi and the referees for their valuable comments regarding the details as well as context of this work. The research of the first author is partially supported in part by grants from British Council's UKIERI program and the United States Department of Transportation's UTRC Region II consortium (RF # 49198-25-26). The research of the fourth author is partially supported in part by NSF Awards CNS-1010789, CCF-1422569 and CCF-1749864, and by research awards from Adobe, Inc. The research of the last author is supported in part by NSF Awards CNS 1010789 and CCF 1422569.

Appendix A: Proofs for Sect. 4

A.1 Proof of Lemma 3

Proof Let $B = A + y_a \leq 2$ be an arbitrary but a fixed feasible value; we now investigate how A and y_a are arranged in WS. A moment's reflection tells us that in WS we will have

$$\sum_{f \in E(u)} y_f p_f = A + y_a p_a = 1 \Rightarrow \ B = A + y_a \geq 1, \ y_a(1 - p_a) = B - 1$$

Recall that the update expression of P_u as shown in Equation (7) is as follows:

$$P_u = (1 - xA)(1 - xy_a)\Pr[Z_u \le t_u - 1] + (1 - xA)xy_a(1 - p_a)\Pr[Z_u \le t_u - 2]$$

Note that in WS, the values of $\Pr[Z_u \le t_u - 1]$ and $\Pr[Z_u \le t_u - 2]$ are functions of B and can be ignored (since in WS, $\mathbb{E}[Z_u] = x(t_u - B)$). For the rest of the expression, we have

$$(1 - xA)(1 - xy_a) \ge (1 - xB + x^2(B - 1)) \text{ and}$$
$$(1 - xA)xy_a(1 - p_a) \ge (1 - x)x(B - 1)$$

The two terms are together minimized when $A = 1$, $y_a = B - 1$ and $p_a = 0$. Note that in this configuration, $\sum_{f \in E(u)} y_f p_f = A + y_a p_a = 1$, and thus the matching constraint is maintained. Since $B = A + y_a$ is fixed, the patience constraint is maintained as well. Therefore, for any fixed value B, P_u will be minimized at the following *feasible* configuration: $A = 1$, $y_a = B - 1$ and $p_a = 0$. This completes our proof.

□

A.2 Statement of Lemma 11 and its Proof

Lemma 11 *Let Z be the sum of a finite collection of independent Bernoulli random variables with $\mathbb{E}[Z] = \mu$. For any $A > \mu$, $A \in \mathbb{Z}$, we have* $\Pr[Z \le A] \ge \Pr[Y \le A]$, *where $Y \sim$* $\mathrm{Pois}(\mu)$.

Lemma 11 follows directly from the following two propositions: Propositions 2 and 3. The proofs of the two propositions will both invoke Proposition 1 below, which we will show first.

Notation We let $\mathrm{B}(N, \mu/N)$ denote the Binomial distribution with parameters $(N, \mu/N)$.

Proposition 1 *Let $Z_x \sim \mathrm{B}(N, \mu/N)$ where $\ell \le N, \ell \in \mathbb{Z}$ and $\mu < \frac{N}{N+1}(\ell + 1)$. Then we have* $\Pr[Z_x = \ell] > \Pr[Z_x = \ell + 1]$.

Proof The result becomes trivial when $\ell = N$. We assume $\ell \le N - 1$.

$$\Pr[Z_x = \ell] = \binom{N}{\ell}\left(\frac{\mu}{N}\right)^{\ell}\left(1 - \frac{\mu}{N}\right)^{N-\ell}$$
$$\Pr[Z_x = \ell + 1] = \binom{N}{\ell + 1}\left(\frac{\mu}{N}\right)^{\ell+1}\left(1 - \frac{\mu}{N}\right)^{N-\ell-1}$$

We get that

$$\frac{\Pr[Z_x = \ell]}{\Pr[Z_x = \ell + 1]} > 1 \Leftrightarrow \frac{(\ell + 1)(N - \mu)}{(N - \ell)\mu} > 1$$
$$\Leftrightarrow \mu < \frac{N}{N + 1}(\ell + 1)$$

□

Proposition 2 considers the case when Z is a sum of at most N independent Bernoulli random variables, each having a mean value that lies in (0, 1]. Subject to this "at most N" restriction and the constraint that $\mathbb{E}[Z] = \mu$ for some given μ, it is easy to see that the problem of minimizing $\Pr[Z \leq A]$, where A is a positive integer that is at most $N - 1$, is that of minimizing a continuous function over a closed set (which in fact is a polytope); thus, this problem has a minimum (as opposed to an infimum). In the following paragraphs, we will repeatedly use the term "optimal configuration", which refers to any configuration of Z_is under which $\Pr[Z \leq A]$ achieves its minimum value; also recall that we refer to a value $z \in [0, 1]$ as "floating" if $z \in (0, 1)$.

Proposition 2 *For any given positive integers A and $N \geq A + 1$, let Z be the sum of at most N independent Bernoulli random variables Z_i with $\mathbb{E}[Z] = \mu$, where $\mu < A$. Then there exists an optimal configuration where each Bernoulli random variable Z_i has the same mean value, which, furthermore, is floating.*

Proof We first show that there exists an optimal configuration where for some (possibly empty) subset $S \subseteq \{1, 2, \ldots, N\}$, (i) all Z_i with $i \in S$ have mean value 1 each, and (ii) all Z_i with $i \notin S$ have the same floating mean value.

Consider an optimal configuration where there are two of our Bernoulli random variables, say Z_1 and Z_2, with different floating means. Let $\mathbb{E}[Z_1] = z_1$, $\mathbb{E}[Z_2] = z_2$ and Z_x be the sum of all the Z_i's excluding Z_1 and Z_2. Assume $0 < z_1 < z_2 < 1$. Notice that

$$\begin{aligned}\Pr[Z \leq A] = \Pr[Z_x \leq A - 2] + \Pr[Z_x = A - 1](1 - z_1 z_2)\\ + \Pr[Z_x = A](1 - z_1)(1 - z_2),\end{aligned}$$

and observe that the coefficient of $z_1 z_2$ is $\Pr[Z_x = A] - \Pr[Z_x = A - 1]$. We consider the following two cases:

- $\Pr[Z_x = A] - \Pr[Z_x = A - 1] > 0$. Then the value $\Pr[Z \leq A]$ can be strictly reduced by the perturbation: $z_1 \leftarrow z_1 - \epsilon, z_2 \leftarrow z_2 + \epsilon$.
- $\Pr[Z_x = A] - \Pr[Z_x = A - 1] < 0$. Then the value $\Pr[Z \leq A]$ can be strictly reduced by the perturbation: $z_1 \leftarrow z_1 + \epsilon, z_2 \leftarrow z_2 - \epsilon$.

Each of the above two cases will lead to a contradiction and thus we conclude $\Pr[Z_x = A] - \Pr[Z_x = A - 1] = 0$ in the original optimal configuration. Since the coefficient of the nonlinear term $z_1 z_2$ in the expression of $\Pr[Z \leq A]$ is zero, we see that our configuration remains optimal after resetting $z_1' = z_1 + z_2, z_2' = 0$ if $z_1 + z_2 \leq 1$ or $z_1' = 1, z_2' = z_1 + z_2 - 1$ if $z_1 + z_2 > 1$. After this change, we can successfully reduce the number of summands with a floating mean value; applying this strategy repeatedly, we reach a scenario where all floating means are the same.

Now we show that S must be empty in any optimal configuration obtained from the above routine. Assume w.l.o.g. that $|S| = 1$ (if $|S| > 1$, just iterate the argument for $|S| = 1$). Say $Z_1 = 1$ deterministically and all other Z_i have a floating mean value $0 < p < 1$. We arbitrarily select one random variable with floating mean, say Z_2, and let Z_x be the sum of all the other Z_i (i.e., all Z_i other than Z_1 and Z_2). Note that

$$\Pr[Z \leq A] = \Pr[Z_x + Z_2 \leq A - 1] = \Pr[Z_x \leq A - 2] + \Pr[Z_x = A - 1](1 - p) \quad (15)$$

where $\mu_x = \mathbb{E}[Z_x] = \mu - 1 - p = N'p$ with N' being the number of variables in Z_x.

Consider the following perturbation to Z_1 and Z_2: replace Z_1 and Z_2 by two i.i.d. Bernoulli random variables Z_0, Z_0' such that $\mathbb{E}[Z_0] = (1+p)/2 = q$. After this perturbation, we get a replacement Z' for Z such that

$$\Pr[Z' \le A] = \Pr[Z_0 + Z_0' + Z_x \le A] \tag{16}$$

$$= \Pr[Z_x \le A-2] + (1-q^2)\Pr[Z_x = A-1] + (1-q)^2 \Pr[Z_x = A] \tag{17}$$

To apply Proposition 1 for Z_x, we set $\ell = A-1$. Note that for $Z_x \sim \mathrm{B}(N', \mu_x/N')$, we have

$$\mu_x = N'p = \frac{N'}{N'+1}(\mu - 1) < \frac{N'}{N'+1}(\ell + 1)$$

Thus we get $\Pr[Z_x = A-1] > \Pr[Z_x = A]$; plugging this into (17) yields

$$\begin{aligned}\Pr[Z' \le A] &< \Pr[Z_x \le A-2] + ((1-q^2) + (1-q)^2)\Pr[Z_x = A-1] \\ &= \Pr[Z \le A],\end{aligned}$$

where the final equality follows from (15). This contradicts the assumption that the original configuration is optimal; thus, S must be empty. □

Let $\Pr(A, \mu, N)$ be the minimum value of $\Pr[Z \le A]$ under the restriction that the number of Bernoulli random variables with positive mean is at most N.

Proposition 3 *For any* $N \ge A+1$, *we have* $\Pr(A, \mu, N) > \Pr(A, \mu, N+1)$.

Proof From Proposition 2, we know $\Pr(A, \mu, N)$ can be achieved when Z follows a Binomial distribution with some parameters $N' \le N$ and μ/N'. Arbitrarily choose a random variable, Z_1, from Z. Let $\mathbb{E}[Z_1] = z = \mu/N'$ and $Z_x = \sum_{i=2}^{N'} Z_i$. Notice that $\mu_x = \mathbb{E}[Z_x] = \frac{N'-1}{N'}\mu$.

Consider perturbing the current configuration of Z as: replace Z_1 with Z_{1a} and Z_{1b} where $\mathbb{E}[Z_{1a}] = \mathbb{E}[Z_{1b}] = z/2$. Now consider $\Pr[Z' \le A]$ where $Z' = Z_x + Z_{1a} + Z_{1b}$. The new value is

$$\begin{aligned}\Pr[Z' \le A] = \Pr[Z_x \le A-2] + \left(1 - \frac{z^2}{4}\right)\Pr[Z_x = A-1] \\ + (1 - z/2)^2 Pr[Z_x = A]\end{aligned}$$

Notice that $\Pr[Z \le A] = \Pr[Z_x \le A-1] + \Pr[Z_x = A](1-z)$. Therefore we have

$$\Pr[Z \le A] - \Pr[Z' \le A] = \frac{1}{4}z^2(\Pr[Z_x = A-1] - \Pr[Z_x = A])$$

To apply Proposition 1 on Z_x, set $\ell = A - 1$. Note that we have

$$\mu_x = \frac{N'-1}{N'}\mu < \frac{N'-1}{N'}A = \frac{N'-1}{N'}(\ell+1)$$

Thus we conclude that $\Pr[Z_x = A - 1] > \Pr[Z_x = A]$, which implies $\Pr[Z \leq A] > \Pr[Z' \leq A]$. Notice that after the perturbation, the number of random variables with positive mean will be at most $N' + 1 \leq N + 1$. Thus $\Pr(A, \mu, N) = \Pr[Z \leq A] > \Pr[Z' \leq A] \geq \Pr(A, \mu, N + 1)$. □

Lemma 11 follows from the preceding two propositions.

Appendix B: Stochastic Matching with Relaxed Patience

B.1 Proof of Lemma 5

Lemma 5 mainly addresses the issue of the configuration of $E(u)$ in the WS, subject to the constraints: (1) $\sum_{f \in E(u)} y_f p_f \leq 1 - y_e p_e$, (2) $\sum_{f \in E(u)} y_f \leq t_u - y_e$ with $t_u \geq 2$ and (3) $0 \leq y_f \leq 1$ for each $f \in E(u)$. Notice that for any given pair (y_e, p_e), part of the result shown in Lemma 2 still applies here: i.e., at most one edge in $E(u)$ takes a floating p_f value. Recalling our previous notation from Sect. 4: (1) $E_1(u)$ and $E_0(u)$ are the set of edges in WS which have $p_f = 1$ and $p_f = 0$ respectively; (2) (y_a, p_a) is the unique potential edge that takes a floating $0 < p_a < 1$ value; and (3) $A = \sum_{f \in E_1(u)} y_f$, $Z_u = \sum_{f \in E_0(u)} Z_f$, where each Z_f is a Bernoulli random variable with mean $x \cdot y_f$ and all the Z_f's are independent.

Lemma 5 consists of the following three propositions; we assume $t_u \geq 2$, $1 > h \geq 1/2$.

Proposition 4 *Suppose e is a small edge and there is no large edge in $E(u)$. Then WS can be characterized as* $Q_2 = \Big(A = 1/2, y_a = B - 1/2, p_a = 1/(2y_a), Z_u \sim \text{Pois}(x(t_u - B))\Big)$ *for some* $1 \leq B \leq 3/2$.

Proposition 5 *Suppose e is a small edge and there is a large edge in $E(u)$. Then WS can be characterized as* $Q_1 = \Big(A = 1, y_a = 0, Z_u \sim \text{Pois}(x(t_u - 1))\Big)$.

Proposition 6 *Suppose e is a large edge. Then WS can be characterized as* $A = 0, y_a = B, p_a = 1/(2B), Z_u \sim \text{Pois}(x(t_u - B - 1/2))$ *for some* $1/2 \leq B \leq 1$.

To prove the three propositions above, we will repeatedly apply local perturbation techniques, similar to the one we used in Lemma 2.

Proof of Proposition 4

Proof A moment's reflection shows that in WS the matching constraint will be tight, i.e., $A + y_a p_a = 1$. Thus we have $A \geq 1/2$ since $y_a p_a \leq 1/2$. As a result, we know for $E_1(u)$ in WS, there will be one edge with $p = 1$, $y = 1/2$ and another edge with

$p = 1$, $y = A - 1/2$. Therefore the lower bound P_u of $\mathcal{P}_u$ in WS can be updated as follows:

$$P_u = \left(1 - \frac{1}{2}x\right)\left(1 - \left(A - \frac{1}{2}\right)x\right)(1 - xy_a)\Pr[Z_u \le t_u] + \left(1 - \frac{1}{2}x\right)\left(1 - \left(A - \frac{1}{2}\right)x\right)xy_a(1 - p_a)\Pr[Z_u \le t_u - 1]$$

Let $B = A + y_a$ be fixed. Substituting $A = 1 - y_a p_a$ into B, we have $y_a(1 - p_a) = B - 1$, implying $B \ge 1$ and $y_a \ge B - 1$. By applying the local perturbation argument, we get that for any given $B \ge 1$, in WS, (A, y_a) will take one of the following two (boundary) values: $Q_1 = (y_a = B - 1, p_a = 0, A = 1)$ where y_a reaches the lower bound and $Q_2 = (y_a = B - 1/2, p_a = \frac{1}{2y_a} = \frac{1}{2B-1}, A = 1/2)$ if $B \le 3/2$ and $Q_2 = (y_a = 1, p_a = 2 - B, A = B - 1)$ if $B \ge 3/2$ where y_a reaches the upper bound.

Note that Q_1 essentially states that in WS, there are two edges $y_1 = y_2 = 1/2$, $p_1 = p_2 = 1$ while no edge takes a floating p_f value. It can be viewed as a special case of Q_2 with $B = 1$ and thus can be ignored.

Now consider Q_2 with $3/2 \le B \le 2$. Assume the WS does not fall at some boundary value of B, i.e., $3/2 < B < 2$. Then we perturb $(A, y_b) \to (A + \epsilon, y_b - \epsilon)$, where y_b is an arbitrary edge in $E_0(u)$. We observe that the term involving ϵ^2 included in the expression of P_u after perturbation is

$$H(\epsilon^2) = (-x^2\epsilon^2)(1 - x)\Pr[Z'_u = t_u] + (-x^2\epsilon^2)\Pr[Z'_u \le t_u - 2] + \epsilon^2 x^3\left(\frac{1}{2} + y_b - 2A\right)$$

where $Z'_u = Z_u - Z_b$ and Z_b is a Bernoulli random variable associated with y_b. Notice that $y_b < 1/2$ and $A = B - 1 > 1/2$. Thus we get that $H(\epsilon^2) < 0$, implying that in WS, $B = 2$ or $3/2$. Again the case Q_2 with $B = 2$ can be ignored since it is a special case of Q_2 with $B = 1$. Therefore the WS can only fall in Q_2 with some $1 \le B \le 3/2$. □

Proof of Proposition 5

Proof We consider the following two cases.

- Consider the first case $A > 1/2$. Notice that in WS, the matching constraint will be tight, i.e., $A + y_a p_a = 1$. Thus $E_1(u)$ must include the large edge since $y_a p_a < 1/2$. For each A, the infimum value of $\prod_{f \in E_1(u)}(1 - xy_f)$ happens at a configuration where $E_1(u)$ consists of a large edge y_1 and at most one other light edge. Thus we can rewrite $\prod_{f \in E_1(u)}(1 - xy_f)$ as $(1 - xy_1 h)(1 - (A - y_1)x)$ where $1/2 < y_1 \le A$. Further, we observe that in WS, either $y_1 = A$ or $y_1 = 1/2 + \epsilon$. The latter is reduced to the case when all edges in $E(u)$ are small, since the adversary will set $y_1 = 1/2$ and y_1 will not be attenuated. Therefore we can update P_u as follows:

$$P_u = (1 - xAh)(1 - xy_a)\Pr[Z_u \le t_u] + (1 - xAh)xy_a(1 - p_a)\Pr[Z_u \le t_u - 1]$$

Let $B = A + y_a$ be fixed with some value $1 \leq B \leq 2$. Applying a similar analysis as in Proposition 4, we get that in WS, (A, y_a) take one of the two (boundary) values, either $Q_1 = (A = 1, y_a = B - 1, p_a = 0)$ or $Q_2 = (A = 1/2 + \epsilon, y_a = B - 1/2 - \epsilon, p_a = (1/2 - \epsilon)/y_a)$ if $B \leq 3/2$ or $Q_2 = (A = B - 1, y_a = 1, p_a = 2 - B)$ if $B > 3/2$.
For Q_1, the expression of P_u can be updated as

$$P_u = (1 - xh)\Pr[Z_u \leq t_u]$$

where $Z_u \sim \text{Pois}(x(t_u - 1))$.
For Q_2 with $B \leq 3/2$, it can be reduced to the case when no large item is in $E(u)$.
For Q_2 with $B \geq 3/2$, the expression of P_u can be updated as

$$P_u = (1 - xAh)(1 - x)\Pr[Z_u \leq t_u] + (1 - xAh)xA\Pr[Z_u \leq t_u - 1]$$

We know that for each $B \geq 3/2$, Z_u should follow a Poisson distribution with mean $x(t_u - B)$. For simplicity, we assume each edge in E_0 has a value of y_f which can be aribitrarily small. Select an edge, say y_b in $E_0(u)$, and perturb as $A \leftarrow A + \epsilon$, $y_b \leftarrow y_b - \epsilon$. We get that the terms involving ϵ^2 included in the final expession of P_u after perturbation, sum to

$$\begin{aligned} H(\epsilon^2) = &-x^2 h\epsilon^2(1 - x)\Pr[Z'_u = t_u] - x^2 h\epsilon^2 \Pr[Z'_u \leq t_u - 2] \\ &+ x^2\epsilon^2(1 - h - 2xAh + xhy_b)\Pr[Z'_u = t_u - 1] \end{aligned}$$

where $Z'_u = Z_u - Z_b$ and Z_b is a Bernoulli random variable associated with y_b. Notice that $\mathbb{E}[Z'_u] \leq x(t_u - B) < t_u - 1$, from which we get $\Pr[Z'_u = t_u - 1] < \Pr[Z'_u \leq t_u - 2]$. Thus we have that for any $h \geq 1/2$,

$$H(\epsilon^2) \leq x^2\epsilon^2(1 - 2h - 2xAh + xhy_b)\Pr[Z'_u = t_u - 1] < 0$$

Therefore we claim that in WS, A should arrive at a boundary value, i.e., either $A = 1$ or $A = 1/2 + \epsilon$. Both of these two cases have been analyzed before.

– Consider the second case $A \leq 1/2$. It implies that since $y_a p_a \geq 1/2$, a should be a large edge. We know that in WS, E_1 should consist of a single edge and P_u has the form:

$$P_u = (1 - xA)(1 - xhy_a)\Pr[Z_u \leq t_u] + (1 - xA)xhy_a(1 - p_a)\Pr[Z_u \leq t_u - 1]$$

When $B = A + y_a$ is fixed at some value $1 \leq B < 3/2$, we know (A, y_a) must take some (boundary) value in WS: either $Q_1 = (A = 1/2 - \epsilon, y_a = B - 1/2 + \epsilon, p_a = (1/2 + \epsilon)/y_a)$ or $Q_2 = (A = B - 1, y_a = 1, p_a = 2 - B)$. Similarly, we see that Q_1 can be ignored since it can be reduced to the case when $y_a p_a = 1/2$ such that it will not be attenuated.

Now we focus on the analysis of $Q_2 = (A = B - 1, y_a = 1, p_a = 2 - B)$ where $1 \leq B < 3/2$. The value of P_u can be updated as:

$$P_u = (1 - xA)(1 - xh)\Pr[Z_u \leq t_u] + (1 - xA)xAh\Pr[Z_u \leq t_u - 1]$$

Applying the same perturbation analysis as before, we get that in WS either $A = 0$ or $A = 1/2 - \epsilon$. The instance of $A = 0$ is just the case of Q_1 while the instance of $A = 1/2 - \epsilon$ can be reduced to the situation without attenuation.

□

Proof of Proposition 6

Proof In this case, we consider a large edge e with $y_e p_e > 1/2$. Recall that in WS, the adversary will try to minimize P_u subject to (1) $\sum_{f \in E(u)} y_f p_f \leq 1/2$, (2) $\sum_{f \in E(u)} y_f \leq t_u - 1/2$ and (3) $0 \leq y_f \leq 1$ for each $f \in E(u)$. In our context, we have in WS, $A + y_a p_a = 1/2$ and $A + y_a + \sum_{f \in E_0(u)} y_f = t_u - 1/2$.

Let $A + y_a = B$ be some fixed value at $1/2 \leq B \leq 3/2$. As before, we observe that in the WS, (A, y_a) should arrive at boundary points, either $Q_1 = (A = 1/2, y_a = B - 1/2, p_a = 0)$ or $Q_2 = (A = 0, y_a = B, p_a = 1/(2B))$ if $B \leq 1$ and $Q_2 = (A = B - 1, y_a = 1, p_a = 3/2 - B)$ if $B > 1$. Observe that Q_1 is a special case of Q_2 with $B = 1/2$ and thus can be ignored.

For the instance $Q_2 = (A = B - 1, y_a = 1, p_a = 3/2 - B)$ with $B \geq 1$. P_u can be updated as

$$P_u = (1 - Ax)(1 - x)\Pr[Z_u \leq t_u] + (1 - Ax)x(A + 1/2)\Pr[Z_u \leq t_u - 1]$$

Notice that $\mathbb{E}[Z_u] = x(t_u - B - 1/2) \leq t_u - 1$, implying that $Z_u \sim \text{Pois}(x(t_u - B - 1/2))$. Perturb in the same way as before: $A \leftarrow A + \epsilon$ and $y_b \leftarrow y_b - \epsilon$ where y_b is an arbitrary edge in $E_2(u)$. We get that the coefficient of ϵ^2 is

$$H(\epsilon^2) \leq -x^3 \left(y_b - \frac{1}{2} - 2A \right) Pr[Z'_u = t_u - 1] < 0$$

Thus we claim that if WS arrives at $Q_2 = (A = B - 1, y_a = 1, p_a = 3/2 - B)$ with some $1 \leq B \leq 3/2$, then B must be at boundary points either $B = 1$ or $B = 3/2$. Both of these two can be viewed as special instances of Q_2 with $1/2 \leq B \leq 1$, and thus can be ignored. □

B.2 Proof of Lemma 6

Proof We split our discussion into the following two cases.

- Consider the first case when e is small with $y_e p_e = 0$. Note that in WS, we have $A + y_a + \sum_{f \in E_0(u)} y_f = 1$. Thus we can set $p_a = 1$, since this does not violate the matching constraint and potentially decreases the value of P_u. This means we can assume in WS there is no floating edge.

  After a similar analysis in Lemma 3, we find that in WS, either $A = 1$ or $A = 1/2$.

When $A = 1$, $P_u = (1 - xh)$ which is larger or equal to that at Q_1 when $t_u \geq 2$, just as shown in Lemma 5.

When $A = 1/2$,

$$P_u = \left(1 - \frac{1}{2}x\right) \Pr[Z_u \leq 1] = \left(1 - \frac{1}{2}x\right)\left(1 + \frac{1}{2}x\right), Z_u \sim \text{Pois}\left(\frac{1}{2}x\right)$$

Consider the P_u in WS at Q_2 with $B = 1$: just as shown in Lemma 5, we have

$$P_u \leq \left(1 - \frac{1}{2}x\right)^2 \Pr[Z_u \leq t_u] \leq \left(1 - \frac{1}{2}x\right)\left(1 + \frac{1}{2}x\right)$$

Thus we claim that WS can not satisfy $t_u = 1$ when $y_e p_e = 0$.

- Consider the second case when e is large with $y_e p_e > 1/2$. Similarly we assume no floating edge in WS and $A = 1/2$. Therefore we have $P_u = (1 - 1/2x)$. Notice that when $t_u \geq 2$, in WS the bound on P_u in case Q_2 shown in Lemma 5 is

$$P_u \leq (1 - 1/2x) \Pr[Z_u \leq t_u] \leq (1 - 1/2x)$$

since $B \geq 1/2$. Thus we claim that WS could not satisfy $t_u = 1$ when $y_e p_e > 1/2$.

□

B.3 Numerical Verification Details in the Proof of Theorem 2:

The following numerical verifications are similar to those shown in the proof of Theorem 1. All our numerical computations were done on Mathematica 10 with precision at least up to the fourth digit after the decimal point.

1. Consider a small edge e with $y_e p_e = 0$ where both $E(u)$ and $E(v)$ have WS at Q_2 just as shown in Lemma 5 with $B = B_u$ and $B = B_v$ respectively. In this case, the Chernoff-Hoeffding bound is

$$P_u(\textbf{Small}_2) \geq \mathbf{L}_a(t_u) = \left(1 - \frac{1}{2}x\right)^2 \left[1 - \exp\left(\frac{-\epsilon^2}{2 + \epsilon}x\left(t_u - \frac{3}{2}\right)\right)\right]$$

where $\epsilon = \frac{1}{x} - 1$.
We verify that:
   - When $t_u, t_v \geq 150$, we see

$$\int_0^1 P_u(\textbf{Small}_2) P_v(\textbf{Small}_2) dx \geq \int_0^1 \mathbf{L}_a^2(150) dx = 0.374$$

   - When $2 \leq t_u, t_v \leq 150$, the integral $\int_0^1 P_u(\textbf{Small}_2) P_v(\textbf{Small}_2) dx$ gets its minimum value of 0.373799 at $B_u = B_v = 1.4984$, $t_u = t_v = 5$.

– When $2 \le t_u \le 150$ while $t_v \ge 150$, we see that

$$\int_0^1 P_u(\mathbf{Small}_2)P_v(\mathbf{Small}_2)dx \ge \int_0^1 P_u(\mathbf{Small}_2)\mathbf{L}_a(150)dx,$$

and that the latter integral gets its minimum value of 0.373899 at $B_u = 1.48529$, $t_u = 5$.

2. Consider a large item $y_e p_e = 1/2 + \epsilon$. In this case, the Chernoff-Hoeffding bound is

$$P_u(\mathbf{Large}) \ge \mathbf{L}_b(t_u) = \left(1 - \frac{1}{2}x\right)\left[1 - \exp\left(\frac{-\epsilon^2}{2+\epsilon}x\left(t_u - \frac{3}{2}\right)\right)\right]$$

where $\epsilon = \frac{1}{x} - 1$.
We verify that:
– When $t_u, t_v \ge 110$, we see

$$\int_0^1 P_u(\mathbf{Large})P_v(\mathbf{Large})dx \ge \int_0^1 \mathbf{L}_b^2(110)dx = 0.539476$$

– When $1 \le t_u, t_v \le 110$, the integral $\int_0^1 P_u(\mathbf{Large})P_v(\mathbf{Large})dx$ gets its minimum value of 0.54563 at $t_u = t_v = 6$, $B_u = B_v = 1$.
– When $2 \le t_u \le 110$, $t_v \ge 110$, we see that

$$\int_0^1 P_u(\mathbf{Large})P_v(\mathbf{Large})dx \ge \int_0^1 P_u(\mathbf{Large})\mathbf{L}_b(110)dx$$

and the latter integral gets its minimum value of 0.536973 at $t_u = 5$, $B_u = 1$.
Thus to reach an approximation ratio of 0.373799, it suffices to set $h \ge 0.6961$.

3. Consider a small item $y_e p_e = 0$ where both $E(u)$ and $E(v)$ have WS at Q_1 just as shown in Lemma 5 with $h = 0.7$.

The Chernoff-Hoeffding bound is,

$$P_u(\mathbf{Small}_1) \ge \mathbf{L}_c(t_u) = (1 - hx)\left[1 - \exp\left(\frac{-\epsilon^2}{2+\epsilon}x(t_u - 1)\right)\right]$$

with $h = 0.7$, $\epsilon = \frac{1}{x} - 1$.
We verify that:
– When $t_u, t_v \ge 100$,

$$\int_0^1 P_u(\mathbf{Small}_1)P_v(\mathbf{Small}_1)dx \ge \int_0^1 \mathbf{L}_c^2(100)dx = 0.442734$$

– When $2 \le t_u, t_v \le 100$, the integral $\int_0^1 P_u(\mathbf{Small}_1)P_v(\mathbf{Small}_1)dx$ gets its minimum value of 0.445811 at $t_u = t_v = 6$.

– When $2 \leq t_u \leq 100$ and $t_v \geq 100$, we see that

$$\int_0^1 P_u(\mathbf{Small}_1)P_v(\mathbf{Small}_1)dx \geq \int_0^1 P_u(\mathbf{Small}_1)\mathbf{L}_c(100)dx$$

and the latter integral gets its minimum value of 0.441362 at $t_u = 5$.

4. Now consider a small item $y_e p_e = 0$ where $E(u)$ has WS at Q_1 with $h = 0.7$ while $E(v)$ has WS at Q_2 with some B_v.

We verify that:

– When $t_u, t_v \geq 30$,

$$\int_0^1 P_u(\mathbf{Small}_1)P_v(\mathbf{Small}_2)dx \geq \int_0^1 \mathbf{L}_c(30)\mathbf{L}_a(30)dx = 0.383453$$

– When $1 \leq t_u, t_v \leq 30$, the integral $\int_0^1 P_u(\mathbf{Small}_1)P_v(\mathbf{Small}_2)dx$ gets its minimum value of 0.40739 at $t_u = 6$, $t_v = 6$, $B_v = 1.49814$.

– When $2 \leq t_u \leq 30$ while $t_v \geq 30$,

$$\int_0^1 P_u(\mathbf{Small}_1)P_v(\mathbf{Small}_2)dx \geq \int_0^1 P_u(\mathbf{Small}_1)\mathbf{L}_a(30)dx$$

and the latter integral gets its minimum value of 0.389957 at $t_u = 4$.

– When $t_u \geq 30$ while $2 \leq t_v \leq 30$,

$$\int_0^1 P_u(\mathbf{Small}_1)P_v(\mathbf{Small}_2)dx \geq \int_0^1 \mathbf{L}_c(30)P_v(\mathbf{Small}_2)dx$$

and the latter integral gets its minimum value of 0.404117 at $t_v = 5$, $B = 1.47987$.

Thus we conclude that the bottleneck configuration is $y_e p_e = 0$, with both of $E(u)$ and $E(v)$ having WS at Q_2 with $t_u = t_v = 5$, $B_u = B_v = 1.4984$. The resultant approximation ratio is 0.373799.

References

1. Abolhassani, M., Esfandiari, H., Hajiaghayi, M., Mahini, H., Malec, D., Srinivasan, A.: Selling tomorrow's bargains today. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, pp. 337–345 (2015)
2. Adamczyk, M., Grandoni, F., Mukherjee, J.: Improved approximation algorithms for stochastic matching. In: Bansal, N., Finocchi, I. (eds.) Algorithms—ESA 2015. Lecture Notes in Computer Science, vol. 9294. Springer, Berlin, Heidelberg (2015)
3. Alon, N., Spencer, J.H.: Wiley interscience series in discrete mathematics and optimization. In: The Probabilistic Method, vol. 6, pp. 85–96 (2008)
4. Bansal, N., Gupta, A., Nagarajan, V., Rudra, A.: When lp is the cure for your matching woes: approximating stochastic matchings. arXiv preprint arXiv:1003.0167v1 [cs DS] (2010)

5. Bansal, N., Gupta, A., Li, J., Mestre, J., Nagarajan, V., Rudra, A.: When LP is the cure for your matching woes: improved bounds for stochastic matchings. Algorithmica **63**(4), 733–762 (2012)
6. Beale, E.M.: On minimizing a convex function subject to linear inequalities. J. Roy. Stat. Soc. B Met. **17**, 173–184 (1955)
7. Birge, J.R., Louveaux, F.: Introduction to Stochastic Programming. Springer, New York (1997)
8. Chen, N., Immorlica, N., Karlin, A.R., Mahdian, M., Rudra, A.: Approximating matches made in heaven. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) Automata, Languages and Programming. ICALP 2009. Lecture Notes in Computer Science, vol. 5555. Springer, Berlin, Heidelberg (2009)
9. Dantzig, G.B.: Linear programming under uncertainty. Manag. Sci. **1**(3–4), 197–206 (1955)
10. Dean, B.C., Goemans, M.X., Vondrák, J.: Approximating the stochastic knapsack problem: the benefit of adaptivity. Math. Oper. Res. **33**(4), 945–964 (2008)
11. Fortuin, C.M., Ginibre, J., Kasteleyn, P.N.: Correlational inequalities for partially ordered sets. Commun. Math. Phys. **22**, 89–103 (1971)
12. Füredi, Z., Kahn, J., Seymour, P.D.: On the fractional matching polytope of a hypergraph. Combinatorica **13**(2), 167–180 (1993)
13. Gandhi, R., Khuller, S., Parthasarathy, S., Srinivasan, A.: Dependent rounding and its applications to approximation algorithms. J. ACM (JACM) **53**(3), 324–360 (2006)
14. Garg, N., Gupta, A., Leonardi, S., Sankowski, P.: Stochastic analyses for online combinatorial optimization problems. In: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, pp. 942–951 (2008)
15. Gupta, A., Kumar, A.: A constant-factor approximation for stochastic steiner forest. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, pp 659–668. ACM (2009)
16. Gupta, A., Ravi, R., Sinha, A.: LP rounding approximation algorithms for stochastic network design. Math. Oper. Res. **32**(2), 345–364 (2007)
17. Immorlica, N., Karger, D., Minkoff, M., Mirrokni, V.S.: On the costs and benefits of procrastination: approximation algorithms for stochastic combinatorial optimization problems. In: Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, pp 691–700 (2004)
18. Levi, R., Pál, M., Roundy, R.O., Shmoys, D.B.: Approximation algorithms for stochastic inventory control models. Math. Oper. Res. **32**(2), 284–302 (2007)
19. Ravi, R., Sinha, A.: Hedging uncertainty: approximation algorithms for stochastic optimization problems. Math. Program. **108**(1), 97–114 (2006)
20. Ruszczynski, A.P., Shapiro, A.: Stochastic Programming, vol. 10. Elsevier, Amsterdam (2003)
21. Shachnai, H., Srinivasan, A.: Finding large independent sets in graphs and hypergraphs. SIAM J. Discret. Math. **18**, 488–500 (2004)
22. Shapiro, A., Dentcheva, D., Ruszczynski, A.: Lectures on Stochastic Programming—Modeling and Theory, vol. 16, 2nd edn. SIAM, Philadelphia (2014)
23. Shmoys, D.B., Swamy, C.: An approximation scheme for stochastic linear programming and its application to stochastic integer programs. J. ACM (JACM) **53**(6), 978–1012 (2006)
24. Srinivasan, A.: Approximation algorithms for stochastic and risk-averse optimization. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp 1305–1313. Society for Industrial and Applied Mathematics (2007)
25. Vazirani, V.V.: Approximation Algorithms. Springer, Berlin (2013)
26. Williamson, D.P., Shmoys, D.B.: The Design of Approximation Algorithms. Cambridge University Press, Cambridge (2011)