J Stat Phys (2008) 131: 783–801
DOI 10.1007/s10955-008-9528-9

Group Testing with Random Pools: Phase Transitions and Optimal Strategy

M. Mézard · M. Tarzia · C. Toninelli

Received: 15 November 2007 / Accepted: 14 March 2008 / Published online: 1 April 2008

Abstract The problem of Group Testing is to identify defective items out of a set of objects by means of pool queries of the form "Does the pool contain at least a defective?". The aim is of course to perform detection with the fewest possible queries, a problem which has relevant practical applications in different fields including molecular biology and computer science. Here we study GT in the probabilistic setting focusing on the regime of small defective probability and large number of objects, $p \to 0$ and $N \to \infty$. We construct and analyze one-stage algorithms for which we establish the occurrence of a non-detection/detection phase transition resulting in a sharp threshold, $\overline{M}$, for the number of tests. By optimizing the pool design we construct algorithms whose detection threshold follows the optimal scaling $\overline{M} \propto Np|\log p|$. Then we consider two-stages algorithms and analyze their performance for different choices of the first stage pools. In particular, via a proper random choice of the pools, we construct algorithms which attain the optimal value (previously determined in (Mézard and Toninelli, arXiv:0706.3104)) for the mean number of tests required for complete detection. We finally discuss the optimal pool design in the case of finite p.

Keywords Group testing · Phase transitions · Information theory

M. Mézard (✉) · M. Tarzia
Université Paris-Sud, LPTMS, UMR8626, Bât. 100, 91405 Orsay cedex, France
e-mail: mezard@lptms.u-psud.fr

M. Mézard · M. Tarzia
CNRS, LPTMS, UMR8626, Bât. 100, 91405 Orsay cedex, France

C. Toninelli
Université Paris VI et VII, LPMA, UMR7599, 4 Pl. Jussieu, Paris, 75005 France

C. Toninelli
CNRS, LPMA, UMR7599, 4 Pl. Jussieu, Paris, 75005 France

1 Introduction

The general problem of *Group Testing* (GT) is to identify defective items in a set of objects. Each object can be either *defective* or OK and we are allowed only to test groups of items via the query "Does the pool contain at least one defective?". The aim is of course to perform detection in the most efficient way, namely with the fewest possible number of tests.

Apart from the original motivation of performing efficient mass blood testing [1], GT has been also applied in a variety of situations in molecular biology: blood screening for HIV tests [2], screening of clone libraries [3, 4], sequencing by hybridization [5, 6]. Furthermore it has proved relevant for fields other than biology including quality control in product testing [7], searching files in storage systems [8], data compression [9] and more recently in the context of data gathering in sensor networks [10]. We refer to [11, 12] for reviews on the different applications of GT.

The more abstract setting of GT is the following. We have N items and each one is associated with a binary random variable x which takes value 1 or 0. We want to detect the value of all variables by performing tests on pools of variables. Each test corresponds to an OR function among the variables of the group, i.e. it returns a binary variable which is equal to 1 if at least one variable of the pool equals 1 and it is equal to 0 otherwise. Here we will only deal with this (very much studied) choice for the tests, often referred to as the *gold-standard* case. It is however important to keep in mind for future work that in many biological applications one should include the possibility of faulty OR tests [2, 13].

The construction of any algorithm for GT involves two ingredients: the *pool design* (the choice of the groups over which tests are performed) and the *inference procedure* (how to detect the value of the items given the result of the tests). The pool design can be composed by one or more stages of parallel queries. For *one-stage* or *fully non-adaptive* algorithms all tests are specified in advance: the choice of the pools does not depend on the outcome of the tests. This would be in principle the easiest procedure for several biological applications. Indeed the test procedure can be destructive for the objects and repeated tests on the same sample require more sophisticated techniques. However the number of tests required by fully non-adaptive algorithms can be much larger than for adaptive ones. The best compromise for most screening procedures [14] is therefore to consider *two-stage* algorithms with a first stage containing a set of predetermined pools (which are tested in parallel) and a second stage which consists in an individual test of all the items whose value has not been determined with certainty in the first stage. Here we will construct one-stage approximate algorithms and two-stage exact algorithms.

In all our study we will focus on *probabilistic* GT in the *Bernoulli p-scheme*, i.e. the situation in which the status of the items are i.i.d. random variables which take value one with probability p and zero with probability $1-p$. In particular, we will be interested in constructing efficient detection algorithms for this GT problem in the limit of large number of objects and small defective probability, $N \to \infty$ and $p \to 0$, and we will focus on the case $N \to \infty$ and $p \to 0$ with $p = 1/N^{\beta}$ (the case $\beta = 0$ stands for $p \to 0$ after $N \to \infty$). This choice was first discussed by Berger and Levenshtein in the two-stage setting in [15] where they proved that for $\beta \in (0, 1)$ the minimal number of tests optimized over all exact two-stage procedure, $\overline{T}(N, p)$ is proportional to $Np|\log p|$.

In the one-stage case we will establish the occurrence of a *phase transition*: considering two simple inference algorithms, we identify a threshold $\overline{M}$ such that when $N \to \infty$ the probability of making at least one mistake in the detection goes to one if $M < \overline{M}$ and to zero if $M > \overline{M}$. By optimizing over the pool distribution, we will construct algorithms for which the detection threshold shows the optimal scaling $\overline{M} = (1-\beta)(\beta)^{-1}(\log 2)^{-2} Np|\log p|$.

Recently in [16] the value of the prefactor of $\overline{T}$ has been determined exactly when $\beta \in [0, 1/2)$ for two-stage procedures. More precisely, the authors have shown that: $\lim_{N\to\infty} \overline{T}/(Np|\log p|) = 1/(\log 2)^2$. Here we will discuss the performance of two-stage algorithms for different choices of the first stage pool design. In particular we will show that the optimal value is obtained on random pools with a properly chosen fixed number of tests per variable and of variables per test (regular-regular case) and also when the number of tests per variable is fixed but the number of variables per test is Poisson distributed (regular-Poisson case). On the other hand we will show that this optimal value can never be attained in the Poisson-regular or in the Poisson-Poisson case.

The paper is organized as follows: In Sect. 2 we introduce the factor graph representation of the problem in the most general case. In Sect. 3 we describe the first simple inference procedure which allows to identify the *sure variables*. In Sect. 4 we analyze one-stage approximate algorithms, while in Sect. 5 we turn to the two-stage exact setting. Finally, in Sect. 6 we give a perspective of our work in view of applications.

2 Pool Design: Random Factor Graphs

A convenient way to define the pool design of each stage is in term of a factor graph representation. We build a graph with two types of vertexes (also called nodes) corresponding to variables and tests. Variable nodes will be denoted by indexes $i, j, \ldots$ and depicted by circles. Function nodes will be denoted by indexes $a, b, \ldots$ and depicted by squares. Whenever a variable i belongs to test a we set an edge between vertex i and a. Thus if N is the overall number of items and M the number of parallel tests in the stage, we obtain a *bipartite graph* with N variable nodes and M test nodes with edges between variables and tests only (in Fig. 1 we depict a case with $N = 6, M = 4$). We denote by Λ_n the fraction of variable nodes of degree n and by P_n the fraction of function nodes of degree n. We will also use a practical representation of these *degree profiles*, standard in coding theory, in terms of their generating functions $\Lambda(x) = \sum_{n\geq 0} \Lambda_n x^n$ and $P(x) = \sum_{n\geq 0} P_n x^n$. In all our pool designs, the first stage is based on a randomly generated factor graph, from the uniform distribution over the set of graphs which have degree profiles $\Lambda(x)$ and $P(x)$. The average variable node and function node degrees are given respectively by $\sum_{n\geq 1} \Lambda_n n = \Lambda'(1)$ and $\sum_{n\geq 1} P_n n = P'(1)$. We also denote by λ_l and ρ_k the probability that a randomly chosen

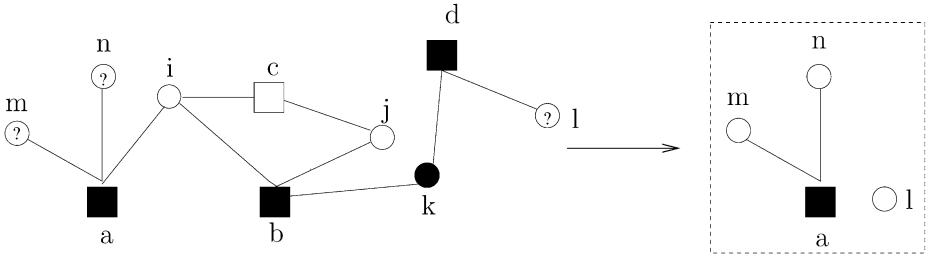

Fig. 1 *Left*: The factor graph corresponding to the first stage of testing. Circles represent variables. *Squares* represent tests, they are filled when the test output is one, empty when it is zero. Variables i and j are sure zeros, variable k is a sure one, variables m,n and l are undetermined. *Right*: The corresponding reduced graph where the sure variables (i, j, k) and the strippable tests (b, c, d) have been erased. Variable l is isolated

edge in the graph is adjacent to a variable node of degree l and to a test node of degree k, respectively. They are given by:

$$\lambda_l = \frac{l\Lambda_l}{\sum_{l'\geq 1} l'\Lambda_{l'}}, \qquad \rho_k = \frac{kP_k}{\sum_{k'\geq 1} k'P_{k'}}. \tag{1}$$

The *edge perspective degree profiles*, defined by

$$\lambda[x] = \sum_{l\geq 1} \lambda_l x^{l-1}, \qquad \rho[x] = \sum_{k\geq 1} \rho_k x^{k-1} \tag{2}$$

are thus given by $\lambda[x] \equiv \Lambda'[x]/\Lambda'(1)$ and $\rho[x] \equiv P'[x]/P'(1)$.

Note that the number of checks M can also be written in terms of these sequences, because the mean degree of variables, $\langle l\rangle = \Lambda'(1)$, and the mean degree of tests, $\langle k\rangle = P'(1)$, are related by $N\langle l\rangle = M\langle k\rangle$. As $\Lambda'(x) = \Lambda'(1)\Lambda(x)$ and $\Lambda(1) = 1$ we get $\langle l\rangle = [\int_0^1 \lambda(x)dx]^{-1}$. Therefore

$$M = N\frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}. \tag{3}$$

3 First Stage: Sure Variables and Isolated Variables

After the first stage of parallel tests we will either use an inference procedure to identify the result (in the one stage case) or choose a new set of pools based on the outcomes of the previous tests (in the two stage case). In our problem, the prior distribution of the N variables, $x = (x_1, \ldots, x_N)$, is Bernoulli: $B_p(x) = \prod_{i=1}^N p^{x_i}(1-p)^{1-x_i}$. Given the outputs of the tests, the inference problem consists in finding the configuration $\bar{x}$ which maximizes

$$P(x) = \frac{B_p(x)}{Z}\prod_{a=1}^{M} \mathbb{1}(T_a(x) = t_a). \tag{4}$$

Here t_a is the value of test a and $T_a(x) = 1$ if $\sum_{j\in\mathcal{N}_a} x_j > 0$, $T_a(x) = 0$ otherwise, where $\mathcal{N}_a$ is the pool of variables connected to a.

Since the minimization of the above function is in general a very difficult task, we start by checking whether some variables are identified with certainty by the first stage and then try to extract information on the remaining variables (see Fig. 1). The first observation is that in order for a variable to be a *sure zero* it should belong to at least one test with outcome zero. On the other hand in order to be a *sure one* it should belong to at least one positive test in which all the other variables are sure zeros. Variables that are neither sure zeros nor sure ones are the *undetermined variables*.

We start by noticing that if a test contains only zeros, or if it contains at least a sure one, then it does not carry any information on the undetermined variables. We call such a test *strippable*, as is the case for tests b, c, d in Fig. 1. We have no information on a variable if it is undetermined and all the tests to which it belongs are strippable. Such variables will be called *isolated*, as is the case for variable l in Fig. 1. The above terminology is motivated by the fact that all the information on the undetermined variables is encoded in a *reduced graph* (see right part of Fig. 1) constructed via the following stripping procedure: erase

all variable nodes which correspond to sure variables and all test nodes which are strippable (note that isolated variables are those that are not connected to any test in the reduced graph). Therefore the inference problem corresponding to the minimization of (4) can be rephrased as a Hitting Set problem on the corresponding reduced graph [17].

Given a variable i and a choice of the pools, the probability that x_i is a sure zero (p^i_{s0}) or a sure one (p^i_{s1}) can be derived as follows. Let us denote by $\mathcal{N}_a$ the set of variable nodes connected to test a and by $\mathcal{N}_i$ the set of test nodes connected to variable i. We introduce the indicator $G_i(x)$ that x_i is a sure 0 as well as the indicator $V_i(x)$ that x_i is a sure one:

$$G_i(x) = (1 - x_i)\left\{1 - \prod_{a \in \mathcal{N}_i} W_{i,a}(x)\right\}, \tag{5}$$

$$V_i(x) = x_i\left\{1 - \prod_{a \in \mathcal{N}_i}\left[1 - \prod_{\substack{k \in \mathcal{N}_a \\ k \neq i}} G_k(x)\right]\right\} \tag{6}$$

which are expressed in terms of

$$W_{i,a}(x) = 1 - \prod_{\substack{j \in \mathcal{N}_a \\ j \neq i}} (1 - x_j). \tag{7}$$

Then p^i_{s0} and p^i_{s1} are given by:

$$p^i_{s0} := \sum_x B_p(x) G_i(x), \tag{8}$$

$$p^i_{s1} := \sum_x B_p(x) V_i(x), \tag{9}$$

where the sum is over all $x \in \{0, 1\}^N$.

It is clear that (8) for p^i_{s0} involves only the variables at distance two from i. Thus, if i does not belong to a loop of length four in the factor graph, $W_{i,a}$ are independent variables and the mean over the variable values in (8) can be easy carried out yielding

$$p^i_{s0} = (1 - p)\left[1 - \prod_{a \in \mathcal{N}_i} (1 - (1 - p)^{k_a - 1})\right], \tag{10}$$

where $k_a = |\mathcal{N}_a|$ is the number of variables which belong to test a. We now take the expectation value of this quantity of the random factor graph ensemble with given degree profiles Λ, P. This leads to $p^i_{s0} = (1 - p)S_0$, with:

$$S_0 := \sum_l \Lambda_l \left(1 - \left(1 - \sum_k \rho_k (1 - p)^{k-1}\right)^l\right) = 1 - \Lambda[1 - \rho[1 - p]]. \tag{11}$$

Formula (9) for p^i_{s1} involves only variables at distance at most 4 from i. If the ball centered in i of radius 4 does not contain any loop, we can perform easily the mean over the variables in (9) and get $p^i_{s1} = pS_1$ with

$$S_1 := 1 - \Lambda[1 - \rho[(1 - p)(1 - \lambda[1 - \rho[1 - p]]]]. \tag{12}$$

The probabilities that t_a is strippable (R^a), x_i is an isolated zero (I_0^i) and x_i is an isolated one (I_1^i) are instead given by

$$R^a = \sum_x B_p(x) \left[\prod_{j \in \mathcal{N}_a}^{N} (1 - x_j) + 1 - \prod_{j \in \mathcal{N}_a}^{N} (1 - V_j(x)) \right], \tag{13}$$

$$I_0^i = \sum_x B_p(x)(1 - x_i) \prod_{a \in \mathcal{N}_i} \left(1 - \prod_{\substack{j \in \mathcal{N}_a \\ j \neq i}} (1 - V_j(x)) \right), \tag{14}$$

$$I_1^i = \sum_x B_p(x) x_i \prod_{a \in \mathcal{N}_i} \left(1 - \prod_{\substack{j \in \mathcal{N}_a \\ j \neq i}} (1 - V_j(x)) \right). \tag{15}$$

In this case, if there is no loop in the ball of radius 6 centered on i, we can again take the expectation over the random graph distribution which yield $I_0 = (1 - p)I$ and $I_1 = pI$, with

$$I = \Lambda\big[1 - \rho[1 - p\tilde{S}_1]\big], \tag{16}$$

$$\tilde{S}_1 := 1 - \lambda[1 - \rho[(1 - p)(1 - \lambda[1 - \rho[1 - p]]]]. \tag{17}$$

4 One-Stage Algorithms

In this section we analyze one-stage algorithms when the number of items, N, goes to infinity and the defect probability, p, goes to zero as $p = 1/N^\beta$ with $\beta > 0$. When constructing the pools we use random graph ensembles of two types: either regular-regular (R-R) graphs (fixed connectivity both for test and variable nodes) or regular-Poisson (R-P) graphs (fixed connectivity for variables, Poisson distribution for the test degree). As for the inference procedure we will consider two types of algorithms: Easy Algorithm (EA) and Belief Propagation (BP). We will show that both undergo a non-detection/detection phase transition when one varies the number of tests, M: we identify a threshold $\overline{M}$ such that for $M < \overline{M}$ the overall detection error goes (as $N \to \infty$) to one while for $M > \overline{M}$ it goes to zero. When $\beta < 1/3$ we can establish analytically the value of $\overline{M}$ which turns out to be equal for the two algorithms: EA and BP have the same performance in the large N limit. We will explain why this transition is robust and we will optimize the pool design (i.e. choice of the parameters of the regular-regular and regular-Poisson graphs) to obtain the smallest possible $\overline{M}$. The resulting algorithms have a threshold value which satisfies $\lim_{N\to\infty} \overline{M}/(Np|\log p|) = (1 - \beta)\beta^{-1}(\log 2)^{-2}$. This is the same scaling in N and p as for the optimal number of tests in an exact two-stage algorithm, albeit with a different prefactor.

4.1 Pool Design

Given a random graph ensemble, we denote by M the number of test nodes, by K the mean degree of tests (which also coincides with the degree of each test in the R-R case) and by L the degree of each variable and we work in a scaling regime characterized by two parameters c, α, defined by:

$$M = cN^{1-\beta} \log N, \qquad K = \alpha/p, \qquad L = MK/N = c\alpha \log N. \tag{18}$$

The degree profile polynomials are:

$$\Lambda^{\text{R-R}}[x] = x^L, \qquad \lambda^{\text{R-R}}[x] = x^{L-1}, \qquad P^{\text{R-R}}[x] = x^K, \qquad \rho^{\text{R-R}}[x] = x^{K-1},$$
$$\Lambda^{\text{R-P}}[x] = x^L, \qquad \lambda^{\text{R-P}}[x] = x^{L-1}, \qquad P^{\text{R-P}}[x] = \rho^{\text{R-P}}[x] = e^{K(x-1)}.$$

Then, if the hypotheses on the absence of short loops which lead to (11), (17) and (16) are valid, the probabilities S_0, S_1 and I are given in the R-R case by:

$$S_0 = 1 - (1 - (1-p)^{K-1})^L, \tag{19}$$
$$S_1 = 1 - \left\{1 - (1-p)^{K-1}\left[1 - \left(1 - (1-p)^{K-1}\right)^{L-1}\right]^{K-1}\right\}^L, \tag{20}$$
$$\tilde{S}_1 = 1 - \left\{1 - (1-p)^{K-1}\left[1 - \left(1 - (1-p)^{K-1}\right)^{L-1}\right]^{K-1}\right\}^{L-1}, \tag{21}$$
$$I = \left(1 - (1 - p\tilde{S}_1)^{K-1}\right)^L. \tag{22}$$

In the R-P case they are given by:

$$S_0 = 1 - (1 - \exp(-Kp))^L, \tag{23}$$
$$S_1 = 1 - \left(1 - \exp\left(-Kp - K(1-p)(1-e^{-Kp})^{L-1}\right)\right)^L, \tag{24}$$
$$\tilde{S}_1 = 1 - \left(1 - \exp\left(-Kp - K(1-p)(1-e^{-Kp})^{L-1}\right)\right)^{L-1}, \tag{25}$$
$$I = \left(1 - \exp(-Kp\tilde{S}_1)\right)^L. \tag{26}$$

It is easy to verify that in leading order when $N \to \infty$ and $p \to 0$ the above quantities for the regular regular and regular Poisson case coincide. They are given by

$$S_0 \simeq 1 - N^d, \tag{27}$$
$$S_1 \simeq \begin{cases} (c\alpha \log N) e^{-\alpha(1+N^{d+\beta}/b)} & \text{if } \beta + d > 0, \\ 1 - N^d & \text{if } \beta + d < 0, \\ 1 - N^{-c\alpha|\log(1-\exp(-2\alpha))|} & \text{if } \beta + d = 0 \end{cases} \tag{28}$$

and

$$I \simeq \begin{cases} (c\alpha \log N)^{c\alpha \log N} e^{-\alpha^2 c \log N(1+N^{d+\beta}/b)} & \text{if } \beta + d > 0, \\ N^d & \text{if } \beta + d < 0, \\ N^d & \text{if } \beta + d = 0, \end{cases} \tag{29}$$

where we set $b = b(\alpha) = (1 - \exp(-\alpha))$ and $d = d(\alpha, c) = -c\alpha|\log b|$, for $N \to \infty$.

Let us discuss in what range of β one expects the above asymptotic behaviors to be valid. As explained in Sect. 3, the only hypothesis in their derivation consists in neglecting the presence of some short loops in a proper neighborhood of the chosen variable. In particular the equation for S_0 is valid if we can neglect the presence of loops of length four through a given variable. Consider for definiteness the R-R case. The probability of having at least

one loop of length four through i, $P(L_4)$, verifies

$$P(L_4) \leq L^2 N \frac{\binom{M}{L-1}}{\binom{M}{L}} \simeq \frac{(\log p)^2}{Np^2}$$

which goes to zero for $\beta < 1/2$. Thus we are guaranteed that (19) is correct in this regime. By the same type of reasoning, we can show that the formulas for S_1 and I are valid respectively for $\beta < 1/4$ and $\beta < 1/6$. However through the following heuristic argument, one can expect that the formula for S_1 (resp. I) be correct in the larger regimes $\beta < 1/2$ (resp. $\beta < 1/3$). Indeed, when we evaluate S_1 we need to determine whether variables at distance 2 from a variable i are sure zeros. We expect the probability of this joint event to be well approximated by the product of the single event probabilities if the number of tests that a variable at distance 2 from i shares with the others is $\ll L$ and if the number of variables that a test at distance 3 from i shares with the others is $\ll K$. Both conditions are satisfied if $\beta < 1/2$ (the probability that a test at distance 3 belongs to more than one variable at distance 2 goes as $(1 - K/N)^{LK}$ and the probability that a variable at distance 4 belongs to more than one test at distance 3 goes as $(1 - K/N)^{L^2K}$). For I the argument is similar but, since we have a further shell in tests and variables to analyze in order to determine whether a variable is isolated or not, we get an extra factor KL in the exponents which lead to the validity of the approximations only for $\beta < 1/3$.

4.2 Easy Algorithm (EA)

A straightforward inference procedure is the one that fixes the sure variables to their value and does not analyze the remaining information carried by the tests, thus assigning to zero all other variables (since $p < 1/2$, the most probable value of a variable in absence of information is 0). We call this procedure *Easy Algorithm* (EA). By definition the probability that a variable is set to a wrong value, E_{bit}, is given by $E_{bit} = p - pS_1$. In the hypothesis of independent bit errors, i.e. if we suppose that the probability E_{tot} of making at least one mistake satisfies $E_{tot} = 1 - (1 - E_{bit})^N$ and if $\beta < 1/2$ (see the discussion at the end of previous section), we can apply (28) which yields

$$E_{tot} \simeq \begin{cases} 1 - \exp(-N^{1-\beta}) & \text{if } \beta + d > 0, \\ 1 - \exp(-N^{1-\beta+d}) & \text{if } \beta + d < 0, \\ 1 - \exp(-N^{1-\beta+c\alpha \log(1-\exp(-2\alpha))}) & \text{if } \beta + d = 0 \end{cases} \tag{30}$$

both for the R-R and R-P graphs. Therefore EA displays a phase transition in the large N limit, when one varies the rescaled number of tests, $c = M/(N^{1-\beta}\beta \log N)$, from a region at $c < \bar{c}(\alpha)$ in which the probability of at least one error, E_{tot}, goes to one, to a region $c > \bar{c}(\alpha)$ where it goes to zero. The threshold of this regime is given by

$$\bar{c}(\alpha) = \frac{1 - \beta}{\alpha |\log(1 - \exp(-\alpha))|}. \tag{31}$$

The most efficient pools, within the R-R and R-P families, are obtained by minimizing $\bar{c}(\alpha)$ with respect to $\alpha = Kp$. The value of the optimal threshold $\tilde{c} = \min_\alpha \bar{c}(\alpha)$ and the parameter $\tilde{\alpha}$ at which the optimal value is attained, namely $c(\tilde{\alpha}) = \tilde{c}$, are

$$\tilde{c} = \frac{1 - \beta}{(\log 2)^2}, \qquad \tilde{\alpha} = \log 2.$$

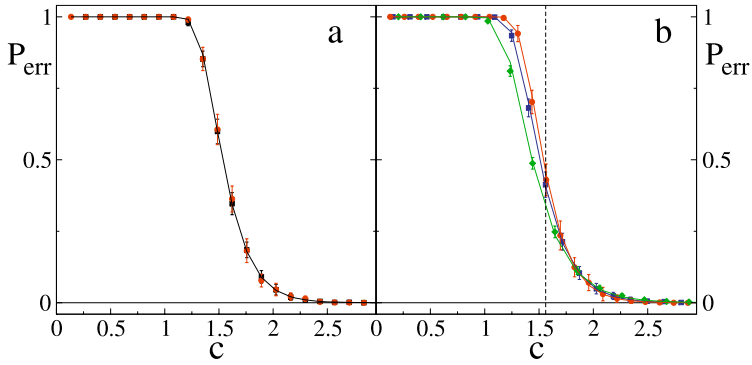

Fig. 2 (Color online) (**a**) Error probability as a function of c using EA (*red circles*) and BP (*black squares*) for regular-regular graphs. The graph parameters are chosen as in (18) with $p = N^{-\beta}$, $\beta = 1/4$, $\alpha = \tilde{\alpha} = \log 2$ and $N = 43321$. The continuous line corresponds to the theoretical prediction of (30). (**b**) Error probability as a function of c for EA. We set again $\beta = 1/4$ and $\alpha = \log 2$, while we choose $N = 1109$ (*green diamonds*), $N = 10401$ (*blue squares*), and $N = 63426$ (*red circles*). The vertical dashed line corresponds to the threshold $\overline{c}$, given by (31)

This, together with (18), gives a threshold

$$\overline{M} = N^{1-\beta} |\log N| (1-\beta) (\log 2)^{-2} \tag{32}$$

for the number of tests. Note that the threshold in the case $\beta = 0$, i.e. if we send $p \to 0$ after $N \to \infty$, is infinite. This corresponds to the fact that for any choice $M = CNp|\log p|$ and $K = \alpha/p$ the bit error $p(1 - S_1)$ stays finite when $N \to \infty$, since K and L depend only on p.

In order to verify the above results and the approximations on which they are based we have performed numerical simulations in the case of R-R graphs with $\beta = 1/4$, $\alpha = \tilde{\alpha}$ and different values of c. The results we obtain confirm that in this regime bit errors can be regarded as independent and formulas (19)–(22) are valid. The values of E_{tot} as a function of c are depicted in Fig. 2a for different values of N. The value of the threshold connectivity and the form of the finite size corrections for the total error (continuous curves) are in excellent agreement with the above predictions (30) and (31). These results support our hypotheses of independent bit errors and of neglecting small loops, and the agreement of the error curve with the theoretical prediction (30) supports our prediction of a phase transition taking place when $N \to \infty$ regime in this $\beta < 1/2$ case.

On the contrary, we have verified that when $\beta > 1/2$ both the independent bit error approximation and the approximation leading to (30) fail, as expected. This can be seen for example in Fig. 3a where we report the results for the case $\beta = 2/3$. In this case the numerical results (black dots) differ from the continuous line which corresponds to (30). With our method we cannot make any prediction on the case $\beta > 1/2$.

4.3 Belief Propagation (BP)

The algorithm considered in previous section is very simple but it does not exploit the information contained in the reduced graph (see Sect. 3). Much more information is contained in the results of the first stage tests. In principle, in order to exploit it optimally, one should find the most probable configuration according to (4). This would take exponential time.

We shall try instead to perform this task approximately with a fast algorithm based on Belief Propagation (BP). The BP algorithm gives an estimate, for each variable i, of the value of the marginal probability $P(x_i)$. Then we will set to one (to zero) variables for which $P(x_i) > 1/2$ (respectively $P(x_i) \leq 1/2$). Let us derive the BP equations for the marginal probabilities. We denote by $\mathcal{N}_i$ ($\mathcal{N}_a$) the set of function (variable) nodes connected to the variable node i (respectively to the function node a), by $P(x_i)^{i\to a}$ the probability of value x_i for the i-th variable in absence of test a and by $P(x_1, x_2, \ldots, x_n)^{(a)}$ the joint cavity distribution in the absence of a (so that $P(x_i)^{i\to a} = P(x_i)^{(a)}$)). We can then write

$$P(x_i)^{i\to a} \cong p^{x_i}(1-p)^{1-x_i} \prod_{b\in\mathcal{N}_i\setminus a} \left(\sum_{\vec{x}_{\partial_{a,i}}} P(\vec{x}_{\partial_{a,i}})^{(b)} \mathbb{1}(T_b(x) = t_b) \right),$$

where by $\vec{x}_{\partial_{a,i}}$ we denote the vector $\{x_j | j \in \mathcal{N}_a \setminus i\}$, and the symbol $\cong$ means equal up to a multiplicative constant, which is fixed by the normalization of the probability. Furthermore we make the usual assumption that the joint cavity distributions $P(\vec{x}_{\partial_{a,i}})^{(b)}$ factorize

$$P(\vec{x}_{\partial_{a,i}})^{(b)} = \prod_{j\in\mathcal{N}_a\setminus i} P(x_j)^{(b)} = \prod_{j\in\mathcal{N}_a\setminus i} P(x_j)^{j\to b}$$

which leads to closed equations for the set of single variable cavity probabilities. In order to simplify these equations we define a normalized message $P(x_i)^{a\to i}$ from function node a to variable node i as

$$P(x_i)^{a\to i} \cong \sum_{j\in\mathcal{N}_a\setminus i} P(x_j)^{(a)} \mathbb{1}(T_a(x) = t_a)$$

and therefore

$$P(x_i)^{i\to a} \cong p^{x_i}(1-p)^{1-x_i} \prod_{b\in\mathcal{N}_i\setminus a} P(x_i)^{b\to i}$$

and

$$P(x_i) \cong p^{x_i}(1-p)^{1-x_i} \prod_{b\in\mathcal{N}_i} P(x_i)^{b\to i}.$$

Using the fact that x_i takes values in $\{0, 1\}$ and that both $P^{a\to i}$ and $P^{i\to a}$ are normalized we introduce cavity fields $h_{i\to a}$ and cavity biases $u_{a\to i}$ defined as follows

$$\begin{aligned} P(x_i)^{a\to i} &= (1-u_{a\to i})\delta_{x_i,0} + u_{a\to i}\delta_{x_i,1}, \\ P(x_i)^{i\to a} &= (1-h_{i\to a})\delta_{x_i,0} + h_{i\to a}\delta_{x_i,1}. \end{aligned}$$

The BP equations for the cavity biases and fields are:

$$u_{a\to i} = \begin{cases} 0 & \text{if } t_a = 0, \\ (2 - \prod_{j\in\mathcal{N}_a\setminus i}(1-h_{j\to a}))^{-1} & \text{if } t_a = 1 \end{cases}$$

and

$$h_{i\to a} = \frac{p\prod_{b\in\mathcal{N}_i\setminus a} u_{b\to i}}{p\prod_{b\in\mathcal{N}_i\setminus a} u_{b\to i} + (1-p)\prod_{b\in\mathcal{N}_i\setminus a}(1-u_{b\to i})}.$$

Our BP-based detection procedure for GT is the following. First initialize the cavity and bias fields to some random values. Then iterate BP equations above until one reaches a fixed point, which is a solution of BP equations. Then, the marginal probability distribution $P(x_i)$ can be written as

$$P(x_i) = (1 - H_i)\delta_{x_i,0} + H_i\delta_{x_i,1}$$

with the full local field H_i satisfying

$$H_i = \frac{p\prod_{b\in\mathcal{N}_i} u_{b\to i}}{p\prod_{b\in\mathcal{N}_i} u_{b\to i} + (1-p)\prod_{b\in\mathcal{N}_i}(1-u_{b\to i})}.$$

The inference procedure is completed by setting x_i to one if $H_i > 1/2$, and to zero if $H_i \leq 1/2$. Note that this BP algorithm automatically gives the correct value for the sure variables. Furthermore one should expect that its performance is better than EA, since it also analyzes the information from tests which are non strippable.

In order to test the performance of BP algorithm we run the procedure on the regular-regular graph for $\beta = 1/4$ and $\alpha = \log 2$ as we did for EA. The total error probability as a function of c is reported in Fig. 2a (black squares). As for EA, a non-detection/detection phase transition occurs at $\tilde{c} = 1/(\log 2)^2$. Thus, even if BP is more sophisticated than EA, their performances coincide in the large N limit. Let us give a heuristic explanation of this result, which will be detailed in the last part of this section. BP sets to zero the sure zeros and to one the sures ones and the only additional information which it exploits with respect to EA is contained in the reduced graph. In particular the performance on the isolated variable (which are not connected to the tests of the reduced graph) is the same for the two algorithms (corresponding to the fact that we have no information on them). As is shown by the scaling in (28), in the $N \to \infty$ limit, almost all 'one' variables are sure as long as $c > \beta/(\alpha|\log(1 - \exp(-\alpha))|)$. This in turn implies that the probability for a test to be strippable (and thus not to carry additional information) goes to one in this regime (see (13)). Therefore EA and BP performances should coincide for $c > \beta/(\alpha|\log(1 - \exp(-\alpha))|)$. Since for $\beta > 1/2$ this regime includes the threshold point $\bar{c} = (1-\beta)/(\alpha|\log(1 - \exp(-\alpha))|)$, the equality of EA and BP thresholds should follow. A proof of this equality via the analysis of the probability that a variable is isolated will be detailed below .

In Fig. 3a we plot instead the total error of EA and BP when $\beta = 2/3$ for $N = 2^{15}$. The data indicates that the BP algorithm performs much better than EA in this case: the reduced graph carries information which is used by BP to optimize the procedure. We have also verified that the difference between BP and EA performance does not diminish as the size of the graph is increased. In Fig. 3b we plot the results for BP again in the case $\beta = 2/3$ but for different values of N. The data become sharper as N is increased. Similarly to the $\beta = 1/4$ case, this seems to indicate the presence of a sharp phase transition in the thermodynamic limit (although the evidence is not as strong, as we don't have any analytical prediction to compare the data with).

To summarize, one can expect that BP is better than EA only in the case $\beta > 1/3$. For smaller β, we shall now argue that BP and EA have the same performance.

Let us start by evaluating the non-detection/detection threshold from BP equations. We denote by $\mathcal{P}^0(H)$ and $\mathcal{P}^1(H)$ the expectation over the random graph distribution of the probability for the full local field on i conditioned to the fact that $x_i = 0$ and $x_i = 1$, respectively.

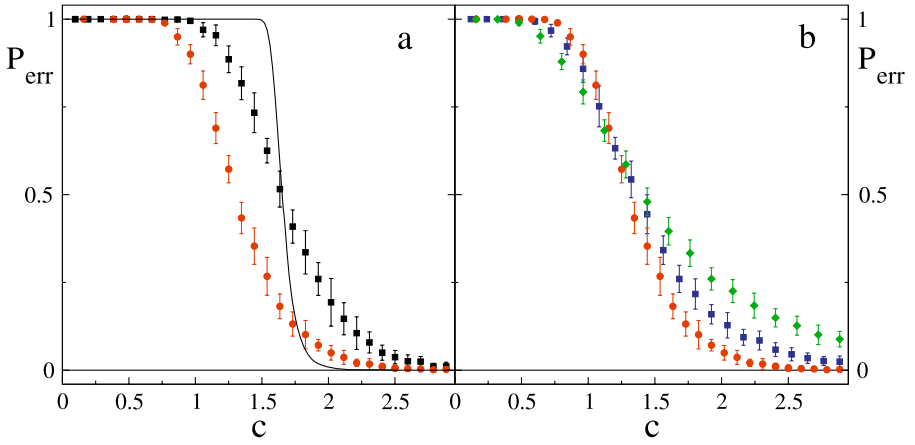

Fig. 3 (Color online) (**a**) Error probability as a function of c for a regular-regular graph using EA (*black squares*) and BP (*red circles*). The graph parameters are chosen as in (18), with $p = N^{-\beta}$, $\beta = 2/3$, $\alpha = 1$ and $N = 2^{15}$. The continuous line corresponds to formula (30). As explained in the text, the discrepancy between the latter and the numerical results confirms that in this regime the approximations leading to (30) are not verified. (**b**) Error probability as a function of c using BP. We set again $\beta = 2/3$, $\alpha = 1$ and we choose $N = 2^{15}$ (*red circles*), 2^{12} (*blue squares*), and 2^9 (*green diamonds*)

From the BP equations it is easy to obtain the following 'replica symmetric' cavity equations satisfied by $\mathcal{P}^0(H)$ and $\mathcal{P}^1(H)$ [18]:

$$\mathcal{P}^0(h) = \sum_{l\geq 0} \Lambda_l \int \prod_{b=1}^{l} dQ^0(u_b)\delta\left(h - \frac{p\prod_b u_b}{p\prod_b u_b + (1-p)\prod_b (1-u_b)}\right), \tag{33}$$

$$\mathcal{P}^1(h) = \sum_{l\geq 0} \Lambda_l \int \prod_{b=1}^{l} dQ^1(u_b)\delta\left(h - \frac{p\prod_b u_b}{p\prod_b u_b + (1-p)\prod_b (1-u_b)}\right) + \lambda_1\delta(h-1), \tag{34}$$

where

$$P^0(h) = \sum_{l\geq 1} \lambda_l \int \prod_{b=1}^{l-1} dQ^0(u_b)\delta\left(h - \frac{p\prod_b u_b}{p\prod_b u_b + (1-p)\prod_b (1-u_b)}\right), \tag{35}$$

$$P^1(h) = \sum_{l\geq 1} \lambda_l \int \prod_{b=1}^{l-1} dQ^1(u_b)\delta\left(h - \frac{p\prod_b u_b}{p\prod_b u_b + (1-p)\prod_b (1-u_b)}\right), \tag{36}$$

$$Q^0(u) = \sum_k \rho_k \int \prod_{j=1}^{k-1}\left[\sum_{y_j} p^{y_j}(1-p)^{(1-y_j)} dP^{y_j}(h_j)\right] \times \left[\delta(u)\prod_{j=1}^{k-1}\delta_{y_j,0} + \left(1 - \prod_j \delta_{y_j,0}\right)\delta\left(u - \frac{1}{2-\prod_j (1-h_j)}\right)\right], \tag{37}$$

$$Q^1(u) = \sum_k \rho_k \int \prod_{j=1}^{k-1} \Big[\sum_{y_j} p^{y_j} (1-p)^{(1-y_j)} dP^{y_j}(h_j) \Big] \times \Big[\delta\Big(u - \frac{1}{2 - \prod_j (1-h_j)} \Big) \Big]. \tag{38}$$

It is now easy to verify that $\mathcal{P}^0(0) = S_0$ and $\mathcal{P}^1(1) = S_1$, where S_0 and S_1 are the probability that a variable is sure zero and one respectively, and are given by (19) and (20). Furthermore the following relation holds

$$\mathcal{P}^0(p) = \mathcal{P}^1(p) \geq \Lambda[Q^0(1/2)] = \Lambda[Q^1(1/2)] = \Lambda[1 - \rho(1 - p\tilde{S}_1)] = I,$$

where I is the probability that a variable is isolated, given in (22).

The probability of setting to a wrong value the i-th variable is $E_{bit} = E^0_{bit} + E^1_{bit}$ with

$$E^0_{bit} = (1-p) \int_{\frac{1}{2}}^{1} \mathcal{P}^0(H) dH, \tag{39}$$

$$E^1_{bit} = p \int_0^{\frac{1}{2}} \mathcal{P}^1(H) dH. \tag{40}$$

By using the above inequalities in these expressions for the bit error probabilities one obtains the following inequalities

$$E^0_{bit} \leq (1-p)(1 - \mathcal{P}^0(0) - \mathcal{P}^0(p)) = (1-p)(1 - S_0 - I), \tag{41}$$

$$pI = p\mathcal{P}^1(p) \leq E^1_{bit} \leq p(1 - \mathcal{P}^1(1)) = p(1 - S_1). \tag{42}$$

We will now show how it is possible to locate the non-detection/detection transition from these inequalities without the need to evaluate the bit error probabilities.

The leading order of the quantities S_0, S_1 and I have been evaluated in Sect. 4.1. Furthermore, for $\beta + d < 0$ the higher order corrections give $S_0 = 1 - N^d - fN^{-\beta+d} \log N$ and $I = N^d - fN^{d-\beta} \log N$ where $f = \exp(-\alpha)(\alpha/2 + 1)/(1 - \exp(-\alpha))$. Thus

$$N^{-\beta+d} \leq E_{bit} \leq 2fN^{-\beta+d} \log N.$$

Therefore, in the assumption of independent bit errors, we get

$$1 - \exp(-N^{1-\beta+d}) \leq E_{tot} = 1 - (1 - E^1_{bit} - E^0_{bit})^N \leq 1 - \exp(-N^{1-\beta+d} \log N)$$

for $\beta + d < 0$, namely $c\alpha |\log(1 - \exp(-\alpha))| > \beta$. Since $\beta < 1/2$ we have $1 - \beta > \beta$ and the above bounds on the total error imply the occurrence of a phase transition at the same value $\bar{c}(\alpha)$ found with the EA algorithm (see (31)). Thus the performance of EA and BP coincide if the approximations leading to (19), (20) and (22) are correct. By the discussion at the end of Sect. 4.1 we know that these approximations are under full control for $\beta < 1/6$ and we expect them to hold also up to $\beta < 1/3$. We conclude that in this regime the value of the threshold for BP transition equals the one for EA (31), as is indeed confirmed by the numerical results that we already discussed for the case $\beta = 1/4$ (see Fig. 2). We stress that there is no reason for that to be true in the regime where the approximations of neglecting proper loops which lead to (19), (20) and (22) do not hold.

For example, as is shown in Figs. 3a and b, in the case $\beta = 2/3$ even if a sharp non-detection/detection phase transition seems to occur when $N \to \infty$, the error probability is certainly not in agreement with (30) which for the chosen parameters would yield to a threshold at $c \simeq 1.453$.

Note that in the discussion above we have upper bounded the bit error with the error over all variables that are neither sure nor isolated and lower bounded it with the error over isolated variables. It is thus immediate to see that the position of the phase transition remains unchanged for all algorithms which set to zero all the isolated variables and set to the correct value the sure variables (EA is indeed the simplest algorithm which belongs to this class). This is due to the fact that the mean number of tests in the reduced graph goes to zero in the detection regime $-d > 1 - \beta > 2/3$, as can be checked using formula (13) and neglecting loops.

Finally, we would like to stress that even if we have shown that EA and BP inference procedures are optimal for R-R and P-R pool designs when $\beta < 1/3$, this does not imply that these algorithms are optimal for all the possible designs of the factor graph. One indication that they might be optimal comes from the results on two-stage exact algorithms presented in Sect. 5. As a further check we have evaluated the thresholds for the Poisson-Poisson (P-P) and Poisson-regular (P-R) cases. Using the same technique as above, we found in both cases a non-detection/detection phase transition which occurs at the same threshold for EA and BP. If we set $K = \alpha/p$, $M = c\alpha \log N$, $L = c\alpha \log p$ the threshold value is

$$\bar{c}(\alpha) = \frac{1-\beta}{\alpha \exp(-\alpha)}. \tag{43}$$

By optimizing (43) over the choice of α we get $\tilde{\alpha} = 1$ and $\overline{M} = eNp|\log p|$, which is larger than the optimal threshold for R-R and R-P.

5 Two-Stage Algorithms

In this section we analyze two-stage exact algorithms when the number of items, N, goes to infinity and the defect probability, p, goes to zero as $p = 1/N^{\beta}$. This setting was first discussed by Berger and Levenshtein in [15] where they proved that if $0 < \beta < 1$, the minimal (over all two-stage exact procedures) mean number of tests, $\overline{T}(N, p)$, satisfies the bounds

$$\frac{1}{\log 2} \le \lim_{N\to\infty} \frac{\overline{T}(N,p)}{Np|\log p|} \le \frac{4}{\beta}.$$

In [16] two of the authors have derived the prefactor for the above scaling when $0 \le \beta < 1/2$,

$$\lim_{N\to\infty} \frac{\overline{T}(N,p)}{Np|\log p|} = \frac{1}{(\log 2)^2} \tag{44}$$

and constructed a choice of algorithms over which this optimal value is attained. Note that our analysis includes the case $\beta = 0$, namely the situation in which the limit $p \to 0$ is taken after $N \to \infty$. Note that the asymptotic result (44) is $1/\log 2$ above the information theoretic bound $\overline{T}(N, p) \ge Np|\log p|/\log 2$. In Sect. 5.1 we give a short account of the derivation of (44) and we construct an optimal algorithm. In Sect. 5.2 we test the performance of algorithms corresponding to different choices of the random pools of the first stage.

5.1 Optimal Number of Tests for $p = 1/N^\beta$, $\beta \in (0, 1/2]$

An exact two-stage algorithm involves a first stage of tests after which all variables are identified and set to their value. Then a second stage is performed where all the remaining variables are individually tested. The mean number of tests, $T(N, p)$, is therefore given by

$$T(N, p) = M + N - \sum_{i=1}^{N} (p_{s0}^i + p_{s1}^i), \tag{45}$$

where M is the number of tests of the first stage and p_{s0}^i and p_{s1}^i are the probabilities for variable i to be sure zero and sure one. The latter in turn are given by (8) and (9) with $\mathcal{N}_a$'s and $\mathcal{N}_i$'s being the neighborhood of tests and variables of the first stage.

It is immediate to verify that in the limit $N \to \infty$ and $p \to 0$ the number of individual check over undetected ones is irrelevant, i.e.

$$\frac{T(N, p)}{Np|\log p|} = \frac{M + N - \sum_{i=1}^{N} p_{s0}^i}{Np|\log p|}. \tag{46}$$

Furthermore p_{s0}^i is always upper bounded by the expression (10) obtained by neglecting loops, as is proven in [16] by using Fortuin-Kasteleyn-Ginibre inequality [20]. We define $f(\vec{m})$ to be the fraction of sites such that among their neighbors there are m_1 tests of degree 1, m_2 tests of degree 2, etc. By using (10) and (46), the optimal number of tests over all two stage procedures can be lower bounded as

$$\frac{\overline{T}(N, p)}{Np|\log p|} \geq \inf_{f(\vec{m})} \left(\frac{\sum_{\vec{m}} f(\vec{m}) (\sum_{j=1}^{N} \frac{m_j}{j} + (1-p) P(\vec{m}))}{p|\log p|} \right), \tag{47}$$

where the infimum is over all possible probability distributions $f : (1, \ldots, N)^N \to \mathcal{R}^+$ with $\sum_{\vec{m}} f(\vec{m}) = 1$ and

$$P(\vec{m}) = \prod_{i=1}^{N} (1 - (1-p)^{j-1})^{m_j}. \tag{48}$$

Minimization over $f(\vec{m})$ can then be carried out and leads in the limit $p \to 0$ to

$$\frac{\overline{T}(N, p)}{Np|\log p|} \geq \frac{1}{(\log 2)^2}. \tag{49}$$

Furthermore the above minimization procedure shows that this infimum is attained for $f(\vec{m}) = \delta_{\vec{m}, \vec{m}}$ with $\vec{m}_i = \delta_{i, \log 2/p}[|\log p|/\log 2]$. This implies that the lower bound is saturated on the uniform distribution over regular-regular graphs with $L = [|\log p|/\log 2]$ and $K = [\log 2/p]$ provided that we can neglect loops in the evaluation of p_{s0}^i. This, as already explained in Sect. 4.1, is true as long as $\beta < 1/2$. Note that the optimal result is also attained if instead of a random construction of pools we fix a regular-regular graph which has no loops of length 4 and has the same choices of test and variable degrees as above. The existence of at least one of such a graph for these choices of K and L when $\beta < 1/2$ is guaranteed by the results in [19]. Thus we have established the result (44) for the optimal value of tests over all exact two-stage procedure, and we have shown a construction based on regular-regular graphs which attains this optimal value.

5.2 Testing Different Pool Designs for $p \to 0$

We will now check the performance of different pool designs corresponding to different random distributions for the pools in the first stage. In all cases we will fix the degree profiles Λ and P and consider a uniform distribution over graphs with these profiles. Using the notation of Sect. 3 and neglecting the presence of loops, the mean number of tests (45) can easily be rewritten

$$\frac{T(N,p)}{N} = \frac{\sum_k \rho_k/k}{\sum_l \lambda_l/l} + (1-p)\Lambda\big[1-\rho[1-p]\big] + p\Lambda\big[1-\rho\big[(1-p)(1-\lambda[1-\rho[1-p]])\big]\big] \tag{50}$$

(we suppose that the fraction of both test and variable nodes of degree zero is equal to zero). As for the one stage case, we consider four different choices of the connectivity distributions corresponding to regular-regular (R-R), regular-Poisson (R-P), Poisson-Poisson (P-P) and Poisson-regular (P-R) graphs and for each choice we have optimized over the parameters of the distribution. The corresponding degree profiles and edge perspectives are given in Sects. 4.2 and 4.3. The first term of the r.h.s. of (50) corresponds to the total number of tests of the first stage per variable, i.e. L/K, while the second and third terms correspond to $(1-p)(1-S_0)$ and pS_1 respectively, where S_0 and S_1 have already been evaluated in the previous section (see (19), (20), (23), (24)).

We now let $K = \alpha/p$ and $L = c\alpha|\log p| + v$ (in order to keep corrections in M to the leading term $Np|\log p|$) and we evaluate (50) for the different pool designs. Then we optimize over the parameters α and c.

5.2.1 Regular-Regular and Regular-Poisson Case

If we set $d = c\alpha|\log(1-\exp(-\alpha))|$, both in the R-R and R-P case we get

$$\frac{T(N,p)}{N} = cp|\log p| + vp/\alpha + p^d(1-\exp(-\alpha))^v + o(p^{1+d}). \tag{51}$$

Thus the optimal value for $p \to 0$ is given by $d = 1$, namely

$$c(\alpha) = \frac{1}{\alpha|\log(1-\exp(-\alpha))|}.$$

By optimizing over α we get $\bar{\alpha} = \log 2$ and $\bar{c} = 1/(\log 2)^2$. Then minimizing over v we get

$$\frac{T}{Np} = \left(\frac{1}{\log 2}\right)^2 (|\log p| + 1 + 2\log\log 2). \tag{52}$$

5.2.2 Poisson-Poisson and Poisson-Regular Case

If we set $f = c\alpha\exp(-\alpha)$, for both the P-P and P-R case we get

$$\frac{T(N,p)}{N} = cp|\log p| + vp/\alpha + p^f \exp(-v\exp(-\alpha)) + o(p^{1+f}). \tag{53}$$

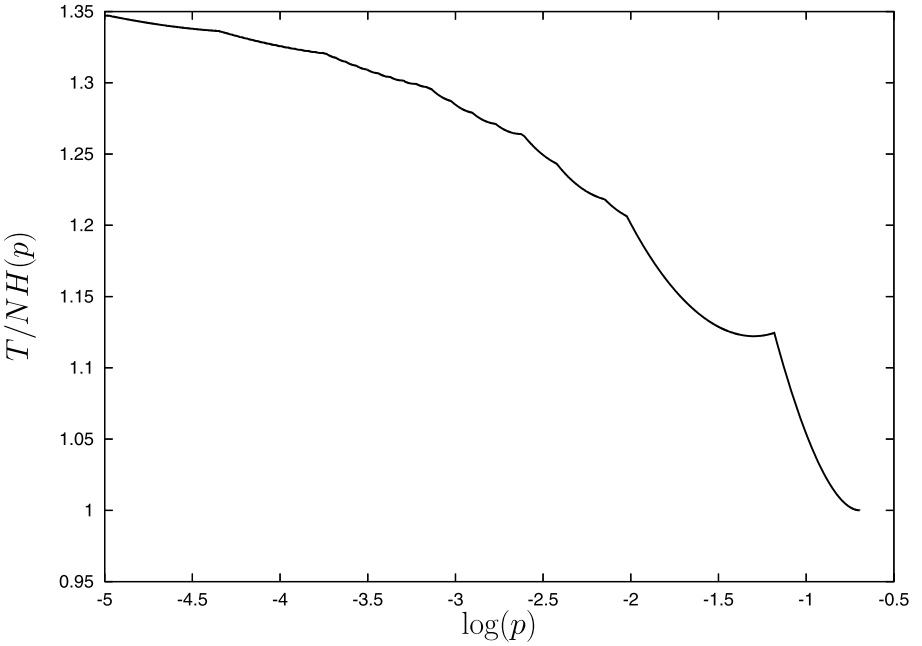

Fig. 4 Expected mean number of tests divided by the information theoretic lower bound $NH(p) = N(p\log_2 p + (1-p)\log_2(1-p))$ for the regular-regular graphs which optimize (50). The non-analyticity points correspond to the values of p where the optimal degree pair L, K changes, see Fig. 5. In the small p limit the curve goes asymptotically to $1/\log 2$ in agreement with (44)

Thus the optimal value for $p \to 0$ is given by $f = 1$, namely

$$c(\alpha) = \frac{1}{\alpha \exp(-\alpha)}.$$

By optimizing over α we get $\bar{\alpha} = 1$ and $\bar{c} = e$. Then minimizing over v we get $v = -e$, thus

$$\frac{T}{Np} = e|\log p| + o(p^f). \tag{54}$$

5.3 Optimal Algorithms at Finite p

The above results show that both for regular-regular and regular-Poisson graphs the optimal asymptotic value (44) can be reached in the case $p \to 0$, while this is true neither in the Poisson-Poisson nor in the Poisson-regular case. Note however that this does not exclude the existence of other distributions for which the optimal value is attained. We stress once more that even if when we performed optimization we did not make any assumption on how $p \to 0$, the results hold only if proper loops can be neglected in the resulting optimal graphs. This includes the following regimes: either $p \to 0$ after $N \to \infty$ or $p = 1/N^\beta$ with $\beta < 1/2$. The reason why we focused on the $p \to 0$ limit is twofold. On the one hand one often deals in practical applications with problems in which the defective probability is small. On the other hand the information theoretic lower bound $T(N, p) \geq Np|\log p|/\log 2$ already tells us that

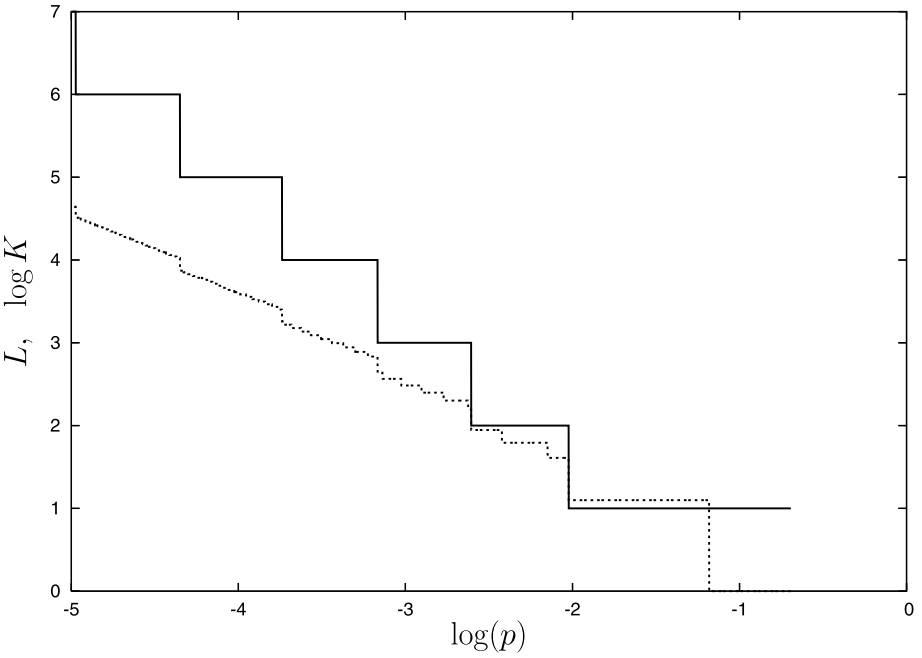

Fig. 5 Values of L (*continuous line*) and of $\log K$ (*dotted line*) corresponding to the couples L, K which give the optimal mean number of tests of Fig. 4

if $p \not\to 0$ the number of tests is proportional to N as in the trivial procedure which tests all variables individually. However one could be interested in the optimal random pool design for the first stage if instead p is held fixed. A natural conjecture in view of the results of the previous sections is that, at least for sufficiently small p, this corresponds again to a regular-regular graph. In order to solve this problem one should find the best degree sequences Λ, P which minimize the expression (50). This is a hard minimization problem which we simplified by first proving for a general choice of N and p that at most 3 coefficients Λ_l and at most 5 coefficient P_r are non zero in the optimal sequence. Plugging this information in some numerical minimization procedure of (50), we have observed that for most values of p the optimal degree sequence is the regular-regular one. There are also some values where the optimal graph is slightly more complicated. For instance for $p = .03$, the best sequences we found are $\Lambda[x] = x^4$ and $P[x] = .45164\, x^{21} + .54836\, x^{22}$, giving $T = .25450$, slightly better than the one obtained with the optimal regular-regular one, $\Lambda[x] = x^4$ and $P[x] = x^{22}$, giving $T = .25454$. But for all values of p we have explored, we have always found that either the regular-regular graph is optimal, or the optimal graph has superposition of two neighboring degrees of the variables, as in this $p = .03$ case. In any case regular-regular is always very close to the optimal structure. In Fig. 4 we depict the expected mean number of tests (divided by the information theoretic lower bound $NH(p) = N(p \log_2 p + (1-p)\log_2(1-p))$) obtained by the numerical minimization of (50) on the ensemble of regular-regular graphs. In the small p limit the curve goes asymptotically to $1/\log 2$ as predicted by (44). In Fig. 5 we depict instead the corresponding optimal degree couples K, L. Note that the non-analyticity points for the expected mean number of tests correspond to the values of p where the optimal degree pair L, K changes.

6 Perspectives

As recalled in the introduction, Group Testing strategies are used in a variety of situations ranging from molecular biology to computer science [1–12]. In most of the applications it is important to take into account the possibility of errors in the tests answers [2, 13, 21–23], i.e. to consider the *faulty-case* instead of the *gold-standard case* analyzed in this work. BP equations for cavity biases and fields analogous to those of Sect. 4.3 can be derived also in the faulty setting and a natural development of the present work is to analyze the performance of the corresponding BP algorithm. A similar task has been performed in [13] for a setting relevant for fault diagnosis in computer networks.

It is important to notice that the relevant form of the test errors depends on the specific application at hand. In the majority of the situations in which GT is a useful tool, one can assume that the errors occur independently in different pools. Thus the error model is completely defined by the probability of *false positive* and *false negative* answers. These can be pool independent or in some cases they depend only on the size of the pool. An example of the latter situation is given by blood screening experiments for which the false negative probability increases with the size of the pools due to the inevitable dilution effect [2, 21].

Finally, it is important to bear in mind that, at variance with our analysis, in practical situations one should take into account finite size corrections as well as the fact that the maximal size of the pool may be limited by experimental constraints.

Acknowledgements This work has been supported in part by the EC grant "Evergrow", IP 1935 of FET-IST.

References

1. Dorfman, D.: Ann. Math. Stat. **14**, 436 (1943)
2. Zenios, S.A., Wein, L.M.: Stat. Med. **17**, 1447 (1998)
3. Barillot, E., Lacroix, B., Cohen, D.: Nucleic. Acids Res. **19**, 6241 (1991)
4. Bruno, W.J., Baldings, D.J., Knill, E., Bruce, D., Whittaker, C., Dogget, N., Stalling, R., Torney, D.C.: Genomics **26**, 21 (1995)
5. Margaritis, D., Skiena, S.: In: Proc. Found. Comput. Sci., p. 613 (1995)
6. Pevzner, P.A., Lipshutz, R.: Toward DNA sequencing chips. In: Proc. 19th Int. Conf. Math. Found. Comput. Sci. Lecture Notes on Computer Sciences. Springer, Berlin (1994)
7. Sobel, M., Groll, P.A.: Bell Syst. Tech. J. **28**, 1179 (1959)
8. Kautz, W.H., Singleton, R.C.: IEEE Trans. Inf. Theory **10**, 363 (1964)
9. Hong, E.H., Ladner, R.E.: IEEE Trans. Image Process. **11**, 901 (2002)
10. Hong, Y.W., Scaglione, A.: In: Proc. IEEE Inf. Theory Workshop, p. 298 (2004)
11. Du, D.Z., Hwang, F.K.: Combinatorial Group Testing and Its Applications. World Scientific, Singapore (2000)
12. Balding, D.J., Bruno, W.J., Knill, E., Torney, D.C.: A Comparative Survey of Nonadaptive Pooling Designs. Springer, New York (1996). Speed, T.S., Waterman, M. (eds.)
13. Zheng, A.X., Rish, I., Beygelzimer, A.: IBM Research Report RC23441 (W0411-168) (2004)
14. Knill, E.: In: Proc. 6th Ann. ACM-SIAM Symp. Discr. Algorithms, p. 369 (1995)
15. Berger, T., Levenshtein, V.I.: IEEE Trans. Inf. Theory **48**, 1741 (2002)
16. Mézard, M., Toninelli, C.: arXiv:0706.3104
17. Mézard, M.M., Tarzia, M.: Phys. Rev. E **76**, 041124 (2007)
18. Richardson, T.J., Urbanke, R.: IEEE Trans. Inf. Theory **47**, 599 (2001)
19. Lu, J., Moura, J.M.F.: IEEE Trans. Magn. **42**, 208 (2006)
20. Fortuin, C.M., Kasteleyn, P.W., Ginibre, J.: Commun. Math. Phys. **22**, 89 (1971)
21. Gupta, D., Malina, R.: Stat. Med. **18**, 1049 (1999)
22. Macula, A.: Ann. Comb. **3**, 61 (1999)
23. Knill, E., Bruno, W.J., Torney, D.C.: Discrete Appl. Math. **88**, 261 (1988)